

A HETEROGENEOUS DATA ENSEMBLE APPROACH FOR PROTEIN FUNCTION PREDICTION UNDER MITOCHONDRION ORGANIZATION

Dinithi Navodhya Sumanaweera

158013D



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Thesis/Dissertation submitted in partial fulfillment of the requirements for the
degree of Master of Science (Research) in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

October 2016

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books)

Signature:

Date:



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The above candidate has carried out research for the Masters thesis/Dissertation under my supervision.

Signature of the Supervisor:

Date:

Name of the Supervisor: Dr. Amal Shehan Perera

ABSTRACT

A heterogeneous data ensemble approach for the classification of *Saccharomyces cerevisiae* proteins under ‘mitochondrion organization’

Proteins are the real role players in keeping a cell healthy and well functioning. An important group of proteins is the subset of mitochondrial proteins that engage in the assembly, arrangement and disassembly of the mitochondrion. Several of them have been identified to cause human diseases. Hence, annotating proteins under the ‘mitochondrion organization’ Biology process is vital for identifying disease causative factors and for designing therapeutics. As manual annotation requires costly and laborious in vitro methods, in silico function prediction is preferred nowadays. Recent studies identify the importance of incorporating data from various biological aspects, to formulate a strong functional context for classification. In addition, many approaches from literature employ ensemble classifiers to attain a higher prediction accuracy. However, an insightful approach for accurate classification; biological data utilization; and biological data type significance determination; is still in need. This study presents an assessment of a heterogeneous data ensemble to classify *Saccharomyces cerevisiae* proteins under ‘mitochondrion organization’. The ensemble consists of nine euclidean-distance based nearest neighbour models and three affinity-based neighbourhood models; it utilizes sequences, protein domains, peptide chain properties, gene expression, secondary structure and interactions. The base models were trained upon annotations from the Gene Ontology, as well as from a publicly available benchmark gold dataset. They show a substantial level of disagreement, implying their effectiveness in collective decision making. Six combination schemes were evaluated for fusing the base model outputs. A Genetic Algorithmically weighted ensemble gives the highest improvement to the best performing base classifier, by displaying an average area under the Receiver Operating Characteristic curve of 92.52%. Moreover, it is capable of determining the biological importance of each data type. Overall, the proposed heterogeneous data ensemble is capable of identifying eight disease related proteins and one disease related protein in a strong and moderate sense, respectively.

Keywords: yeast; proteins; mitochondrion; weighted ensemble; data heterogeneity; genetic algorithm; supervised learning

To my beloved parents, grandmother and brother



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

ACKNOWLEDGEMENT

I would like to express my heartiest and sincere gratitude,

To my parents, for all their support, guidance, motivation and inspiration

To my advisor and supervisor Dr. Amal Shehan Perera, for his immense support, invaluable advice, continuous guidance and encouragement, through productive discussions and progress reviews, in making this research a success

To my Research Review Committee: Prof. Nalin Wickramarachchi and Dr. Dulani Meedeniya for their constructive feedback and encouragement

To Prof. T. L. Shamala Tirimanne from the University of Colombo, for offering me with her expertise in Biology through informative discussions, despite her busy schedule



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

To Dr. Surangika Ranathunga and Dr. Charith Chitraranjan, for those illuminating and motivating discussions despite their busy schedules

To Prof. Gihan Dias, for his constant advice and guidance

To Prof. Vajira H. W. Dissanayake and Mr. Nilaksha Neththikumara from the Human Genetics Unit, University of Colombo, for providing me with Training in Bioinformatics

To the Department of Computer Science and Engineering, the Senate Research Grant Committee, the Faculty of Graduate Studies and the staff in general at the University of Moratuwa, for supporting and facilitating my research with necessary resources throughout the course of study

TABLE OF CONTENTS

Declaration of the Candidate & Supervisor	i
Abstract	ii
Acknowledgement	iv
Table of Contents	v
List of Figures	1
List of Tables	3
List of Abbreviations	4
1 Introduction	6
1.1 <i>Saccharomyces cerevisiae</i>	6
1.2 Importance of ‘mitochondrion organization’	7
1.3 Need for Protein Function Prediction	9
1.4 Problem Definition	9
1.5 Research Objectives	12
1.6 Contributions	12
1.7 Organization	13
2 Background Study	14
2.1 Bioinformatics and Biological Data Mining	14
2.2 Overview to Functional Genomics and Proteins	16
2.2.1 Basics of Proteins	17
2.2.2 Structure of Proteins	18
2.2.3 Protein Folding	21
2.2.4 Protein Motifs	21
2.2.5 Importance of Protein Structure Determination	22
2.2.6 Protein Domains	22
2.2.7 Protein Families	23
2.2.8 Origination of Proteins	23
2.3 Functions of Proteins	25
2.4 Importance of Protein Function Annotation	26

2.5	Biological Data Sources for Functional Genomics	28
2.6	Microarray Gene Expression	29
2.7	Gene Ontology (GO) Functional Classification Scheme	32
2.7.1	Standard Format	33
2.7.2	Hierarchical Structure	33
2.7.3	Annotations	34
2.8	The ‘mitochondrion organization’ GO Term	35
3	Literature Review	37
3.1	Homology based Protein Function Prediction	37
3.2	Multi-class Classification and Data Heterogeneity	38
3.2.1	A True Path Rule Hierarchical Ensemble Approach	38
3.2.2	Hierarchical Classification of G Protein-Coupled Receptors	40
3.2.3	Predictive Clustering Trees and their Ensembles	42
3.2.4	Bayesian Hierarchical Correction	43
3.2.5	HML Boosting	45
3.2.6	Label Similarity Incorporated kNN Algorithm	46
3.2.7	SVM based Ensemble Framework	47
3.2.8	Hierarchical Bayesian Integration Algorithm	49
3.2.9	Semi Supervised Multi-label Collective Classification	50
3.2.10	Ensemble based GPCR Class Prediction	51
3.2.11	Transductive Multi-label Ensemble Classification	51
3.2.12	MS-kNN for Multiple Data Integration	53
3.2.13	Functional Association Network based Approaches	54
3.2.14	BLAST based Local Prediction	55
3.2.15	An Ensemble for ‘mitochondrion organization’ Prediction	57
3.3	Selection of Positive and Negative Examples	57
3.3.1	Negative Example Selection Methods	58
3.4	Class Imbalance	64
3.4.1	Class Imbalance and Feature Selection	66

4	Methodology	67
4.1	Data Retrieval and Preprocessing	67
4.1.1	Protein Annotation Data	68
4.1.2	Sequence Data	68
4.1.3	Domain Data	69
4.1.4	Properties Data	70
4.1.5	Gene Expression data	70
4.1.6	Secondary Structure Data	77
4.1.7	Interaction Data	77
4.2	Protein Instance Representation Methods	78
4.2.1	Pseudo Amino Acid Composition (PAAC)	78
4.2.2	Quasi-Sequence-Order Descriptor (QSOD)	80
4.2.3	Conjoint Triad Descriptors	81
4.2.4	Secondary Structure based Representation	83
4.2.5	Latent Dirichlet Allocation (LDA) Topic Representation	83
4.2.6	Gene Expression Profile Representation	86
4.3	Ensemble Based Classification	87
4.3.1	Heterogeneous Data Ensemble	89
4.3.2	Affinity-based Neighbourhood models	89
4.3.3	Nearest Neighbour Models	89
4.3.4	Base Model Combination Scheme	92
4.3.5	Performance Measures	95
5	Experimentation, Results Analysis and Discussion	99
5.1	Experimental Setup	99
5.2	LDA Topic Modeling based Approach	102
5.3	Optimal Number of Neighbours	104
5.4	Base Model Evaluation	106
5.5	Evaluation of the Inter-rater Agreement	108
5.6	Genetic Algorithm based Weight Optimization	109
5.7	Ensemble Classification Performance	115
5.8	Identification of Disease Related Proteins	119

6	Conclusions and Recommendations	122
	References	126
A	Exploratory Data Analysis	139
A.1	Initial Analysis of GO Annotations	139
A.2	Data Visualizations	139



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF FIGURES

Figure 1.1.1	<i>Saccharomyces cerevisiae</i>	7
Figure 2.2.1	Amino acid residue and Peptide bond formation	17
Figure 2.2.2	List of 20 Amino acid types	18
Figure 2.2.3	Example alpha helix protein and beta sheet protein	19
Figure 2.2.4	Parallel and anti-parallel beta sheets	20
Figure 2.2.5	Gene expression	23
Figure 2.6.6	Microarray technology	30
Figure 2.8.7	GO ancestor chart for ‘mitochondrion organization’	36
Figure 3.2.1	Fuzzy kNN ensemble model by Gu et al.(2015)	52
Figure 3.2.2	Directed bi-relation graph	52
Figure 4.1.1	Example FASTA sequence of a protein	69
Figure 4.2.2	Amino acid residue classification	82
Figure 4.2.3	Conjoint triads	82
Figure 4.3.4	Parallel and anti-parallel β sheet formation	91
Figure 4.3.5	Example bayesian network	96
Figure 4.3.6	Kappa scale	98
Figure 5.2.1	ROC plots for the LDA model based approach	104
Figure 5.3.2	k vs mean AUC	105
Figure 5.6.3	(a) GA optimized weights (b) mean ROC AUC of base models	110
Figure 5.6.4	Best fitness value over each sample	110
Figure 5.6.5	Order of data types with respect to both average and maximum fitness giving weight vectors	112
Figure 5.7.6	ROC plots of base models and GA-weighted Ensemble	117
Figure 5.7.7	ROC plots of Ensemble models	119
Figure 5.8.8	Disease protein identification matrix	121
Figure 5.8.9	Disease related protein identification over the 10 samples	121
Figure A.2.1	Expressions 1 - Before normalization/preprocessing	140
Figure A.2.2	Expressions 1 - After normalization/preprocessing	141

Figure A.2.3	Expressions 2 - MA plots before background correction	141
Figure A.2.4	Expressions 2 - MA plots after background correction	142
Figure A.2.5	Expressions 2 - MA plots after within/between array normalization	142
Figure A.2.6	Expressions 2 - After normalization/preprocessing	143
Figure A.2.7	Expressions 2 - After further normalization	144
Figure A.2.8	Expressions 3 - before & after normalization/preprocessing	145
Figure A.2.9	Expressions 4 - Before & after normalization/preprocessing	146
Figure A.2.10	Expression profiles of housekeeping genes	147
Figure A.2.11	Properties Data	148



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF TABLES

Table 1.1	Disease related ‘mitochondrion organization’ proteins as listed in [8]	8
Table 5.1	LDA model based approach evaluation results	102
Table 5.2	Individual base model performance results I	106
Table 5.3	Individual base model performance results II	107
Table 5.4	Kappa measure for individual samples	109
Table 5.5	GA optimized weights for all 10 samples	114
Table 5.6	Ensemble performance results I	115
Table 5.7	Ensemble performance results II	115
Table 5.8	PR curve AUC values of ensemble models	119



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF ABBREVIATIONS

AGPS	Annotating Genes with Positive Samples
ANOVA	Analysis of Variance
AUC	Area Under the Curve
BioGRID	Biological General Repository for Interaction Datasets
BLAST	Basic Local Alignment Search Tool
CAFA	Critical Assessment of protein Function Annotation
CD	Czekanowski-Dice
CTD	Conjoint Triad Descriptor
Da	Dalton (atomic mass unit)
DF	Degrees of Freedom
DNA	Deoxyribonucleic Acid
FunCat	Functional Catalogue
GA	Genetic Algorithm
GO	Gene Ontology
GPCR	G Protein-Coupled Receptor
HER2	Human Epidermal Growth Factor Receptor 2
IEA	Inferred from Electronic Annotation
LDA	Latent Dirichlet Allocation
MIPS	Munich Information Center for Protein Sequences
NGS	Next Generation Sequencing
NLP	Natural Language Processing
NMR	Nucleic Magnetic Resonance
NN	Nearest Neighbour
mRNA	Messenger Ribonucleic Acid



PAAC	Pseudo Amino Acid Composition
PCT	Predictive Clustering Tree
PDB	Protein Data Bank
PPI	Protein Protein Interactions
PR	Precision-Recall
QSOD	Quasi Sequence Order Descriptor
RNA	Ribonucleic Acid
ROC	Receiver Operating Characteristic
SGD	Saccharomyces Genome Database
SS	Secondary Structure
SVM	Support Vector Machine
TMC	Transductive Multi-label Classifier
TPR	True Path Rule
3D	Three dimensional



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Chapter 1

INTRODUCTION

This dissertation presents a research study carried out on mining heterogeneous biological data, for predicting protein functions of a well-known and a well-studied model organism: *Saccharomyces cerevisiae*. The primary functional class of interest is the ‘mitochondrion organization’.

Proteins are the real role players in keeping a cell healthy and well functioning. They often engage in various molecular, cellular and physiological activities that are essential for the well-being of an organism [1]. It could be either to maintain metabolism and cellular homeostasis under varying environmental conditions, or to regulate and organize cellular reproduction, growth and development [2]. As a standard, these functions are well-defined through species independent functional classification schemes such as Gene Ontology (GO) [3] and MIPS FunCat [4]. Any abnormality in protein folding, expression or regulation might cause a disruption in their functions, impeding essential biological pathways and resulting in diseases or other adverse phenotypes (e.g. breast cancer progression [5], sickle cell anaemia [6]). Hence, revealing protein functions is vital for understanding complex biological processes, for identifying disease causative factors and for designing therapeutics.

1.1 *Saccharomyces cerevisiae*

Saccharomyces cerevisiae is one of the widely and commonly used single cellular microorganisms for studying protein functions of higher order eukaryotes such as humans. It is a species of yeast, categorized under the eukaryota domain and belonging to the kingdom Fungi [7]. Figure 1.1.1 presents the microscopic and cellular view of *S. cerevisiae*. Yeast is generally believed to be having the minimal set of genes required to sustain the eukaryotic free living organisms [8]. Most

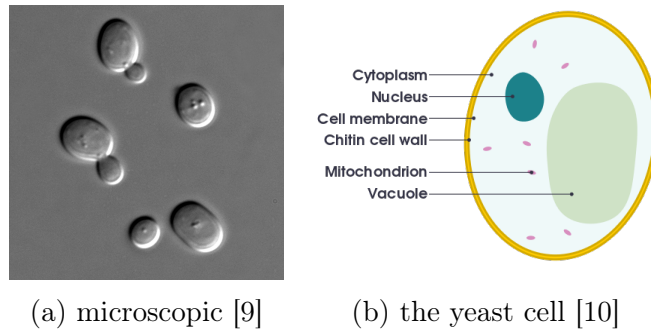


Figure 1.1.1: *Saccharomyces cerevisiae*

importantly, it is known to be substantially contributing towards understanding eukaryotic biology. The main reasons are the evolutionary conservation of many yeast genes in eukaryotes and the much higher feasibility of carrying out yeast genome-scale experiments (i.e. the ability for quick reproduction and growth under variety of conditions inside a laboratory [8]). For instance, a yeast protein mutant which is responsible for a certain phenotype, could suggest the existence of a human ortholog that causes a similar phenotype in humans [2]. Thus, ever since the yeast genome got published in 1996 as the very first eukaryotic genome to be sequenced, most of the GO annotations, eukaryotic gene/protein functions and interactions have been derived through yeast genome wide studies [2]. Due to this reason, quite a large number of data are available for *S. cerevisiae*.

1.2 Importance of ‘mitochondrion organization’

When it comes to human health, mitochondria plays an essential role by acting as the cells’ power house. This cellular organelle maintains the cellular energy balance and calcium signalling modulation, while giving house for many other significant biosynthetic pathways [11]. According to the Gene Ontology, ‘mitochondrion organization’ (GO:0007005) is the cellular level process which is responsible for the assembly, arrangement and disassembly of a mitochondrion (including the replication of the mitochondrial genome; mitochondrial morphogenesis and distribution; and the synthesis of new mitochondrial components). If any protein that is engaged in this process becomes impaired somehow, it would lead to a

disordered cell functionality, disabling the mitochondrial function [11]. Consequently, the condition may get manifested as a disease. One in five mitochondrial proteins are known to be human disease related [12].

Saccharomyces cerevisiae facilitates the understanding of many mitochondrial human diseases. For instance, human orthologs of some *S. cerevisiae* proteins affect mitochondrial respiration due to mutations in them [8]. Exploring the corresponding yeast proteins would be beneficial to decipher the disease causality and develop treatment procedures. Nine proteins can be identified as involved in GO:0007005, among the list of human disease related proteins given by Barrientos [8]. They are presented along with their related clinical manifestations in Table 1.1. *S. cerevisiae* is an ideal model organism to examine such human diseases, as mitochondrial biogenesis is one of the conserved cellular functions from yeast to human. Hence, *S. cerevisiae* protein classification under ‘mitochondrion organization’ functional class is much important for identifying and classifying disease related human orthologs.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Protein/ortholog	Clinical manifestations
SDH1/SDHA	Leigh syndrome
BCS1/BCS1L	Tubulopathy, encephalopathy and liver failure
COX10/COX10	Ataxia, tubulopathy
SCO1/SCO1	Hepatic failure, encephalopathy
CYC3/HCCS	Microphthalmia with linear skin defects syndrome ¹
AAC1/ANT1	Progressive external ophthalmoplegia ²
AFG3/SPG7	Spastic paraplegia
FUM1/FH	Encephalomyopathy
MGM1/OPA1	Optic atrophy type I

Table 1.1: Disease related ‘mitochondrion organization’ proteins as listed in [8]

¹X-linked dominant disorder

²Autosomal dominant disorder

1.3 Need for Protein Function Prediction

High-throughput sequencing technologies have enabled the discovery of new proteins at a brisk pace, eliciting the need for their function annotation at an equivalent rate. Day by day, more and more human diseases are getting prevalent and thus, efficient protein function annotation has become a necessity. However, manual annotation practice requires costly and laborious in vitro methods such as the purification of a protein of interest, gene knockout, fusion protein preparation and conduction of various biological experiments (e.g. two hybrid screening) [13]. They need a huge experimental and human effort, often resulting in a low throughput at a higher cost (in terms of time, effort and equipment); and an infeasibility to reach the current discovery rate of unknown proteins. Moreover, only a single protein can be focused at a time during this manual practice. Thus, the requirement for developing effective computational protein function prediction methods has arisen. Such in silico protein function prediction not only reduces the overall cost, but also acts as a guide to further experimental validation and biocuration of inferred function annotations. An enormous amount of continuously proliferating biological data and a considerable amount of manually annotated data, pave the way to build supervised learning and prediction models. Biological data types include nucleotide sequences; amino acid sequences; gene expression data; molecular interaction data; protein structural data; biomedical literature; and various other experimental data types. Thus numerous protein function prediction models are being introduced, while global initiatives such as CAFA [14] are in effect for collaboration, assessment and further effort encouragement.

1.4 Problem Definition

Protein function prediction is a widely addressed problem in Functional Genomics research. The ultimate goal is to annotate proteins with their corresponding functions as to understand the various kinds of biological processes and pathways that these macromolecules are engaged in. The task is primarily categorized under su-

pervised learning, and the protein functions are taken to be the target concepts (i.e. the classes of interest). Due to a single proteins' involvement in multiple functions, the fundamental problem entails the multi-class, multi-label classification need. In addition, the hierarchically structured functional classes; highly skewed class distributions; elusive nature of negative examples; large number of functional classes; different degrees of reliability for existing annotations [15]; and the extremely large protein instance space, make the problem domain much more intricate. Hence computational researchers tend to focus more on a single aspect of the problem at a time. Further, Biologists often study a single protein function at a time. Moreover, a generic learning model may not be effective for protein function prediction, since different functional contexts require different learning strategies under different concerns.

In the focus of a single protein function, the basic initial step is to gather the set of proteins which have already been annotated (i.e. positive examples) under the function of interest. It is also required to gather a set of proteins which do not engage in the function of interest (i.e. negative examples). At this point, the issue of class imbalance and the ambiguity at negative example selection should be taken care of. The class imbalance is caused by the fact that only few proteins engage in a particular function. In addition, the negative example selection is difficult due to the incompleteness of experimentally validated protein annotations. This is because, not all non-annotated proteins are true negative proteins. Some of them might not have been identified as engaged in the function of interest yet. Ideally, an experimentally verified positive and negative protein set should be obtained for training a classification model. Also a variety of biological data types have to be incorporated as to formulate the functional context during the model learning process. Next, a model is built to learn how to distinguish between a positive protein and a negative protein with respect to the particular function class. It can then be used to obtain a posterior probability value which indicates the class membership of a functional-contextually unknown protein. Ultimately, this model output can support annotation decision making and further experi-

mental validation.

The plethora of literature leverages different data types for protein function prediction, as researchers identify their importance to the supervised learning setting. The latent network of how varying biological aspects interconnect to form a functional context, is convoluted and still not completely understood. Nevertheless, it can devise the role of a protein at different levels of abstraction. For instance, proteins do not usually operate in isolation, but interact or bind with other proteins and molecules to perform the intended functions. Thus, the affinity between two proteins can suggest that they are involved in the same biological process. Moreover, homologous protein sequences have a chance of sharing a conserved genomic region which is corresponding to the same function. Furthermore, the presence of a certain structural motif can be evidential of a certain protein function as well (e.g. zinc finger structural motif for DNA binding). At the higher level, the stable conformation of a protein in three dimensional space affects how it carries-out functions by interacting with other molecules and the surrounding environment. In addition, their subcellular localization, targeted molecules, the level of expression in tissues and their role in the growth or the development of an organism altogether form their functional context [13]. Integration of such varying functional aspects can give a more confident clue about protein functionality. Hence, when determining the functions of a protein, those factors have to be taken into consideration.

Overall, this supervised learning problem requires a more insightful approach for attaining a higher classification accuracy, while effectively utilizing previously mentioned biological data types, and determining each of their significance for the functional context of interest.

1.5 Research Objectives

The ultimate objective of this research was to assess the use of a properly engineered heterogeneous data ensemble classification model for recognizing a *Saccharomyces cerevisiae* proteins' engagement in the 'mitochondrion organization' biology process. The challenge introduces the need for reliable dataset selection, data integration, data type specific quality control, preprocessing and effective utilization. An important consideration was given to the methodology of harnessing a diverse range of heterogeneous biological data sets (often complex and noisy), which together explain the 'mitochondrion organization' context..

The approach incorporates six types of biological data: amino acid sequences; protein domains; gene expression; peptide chain properties; secondary structure; and interactions. Each type undergoes specific preprocessing prior to data mining, for an accurate protein instance representation. This is a crucial step for dealing with complications, un-reliabilities, outliers, inconsistencies, varying data ranges and varying data formats, introduced by these data. Especially the biological datasets produced by high throughput experiments may suffer from high error rates and random noise [16, 17, 18]. Moreover, unreliable sources or data instances should be avoided, as usage of such error prone data could lead to error propagation, resulting in even more erroneous results.

1.6 Contributions

During the course of study, the following contributions were made.

- Evaluation of an LDA topic modeling approach for representing a protein domain specific amino acid sequence
- Evaluation of a genetic algorithmically (GA) weighted heterogeneous data ensemble approach
- Comparison with four other base-line combination schemes for fusing base model outputs

- Evaluation of a second level ensemble of different combination schemes

1.7 Organization

The rest of the chapters are organized as follow. Chapter 2 presents the basic concepts and background knowledge related to Biological data mining, Functional Genomics and proteins. Chapter 3 gives a comprehensive literature review in terms of the existing protein function prediction methods. Chapter 4 elaborates on the data material, computational methods and tools used for developing the protein function prediction approach. Chapter 5 explains the experimental setup and presents a result analysis with discussion, followed by conclusions and future recommendations in Chapter 6.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Chapter 2

BACKGROUND STUDY

In nature, biological systems are very complex in terms of their formation, existence, function and maintenance. Hence a background study was conducted prior to the literature review, in order to understand the exact biological problem domain. Many online resources such as [19, 20] were referred and Biology expert advice was received during the following domain knowledge acquisition process.

2.1 Bioinformatics and Biological Data Mining

High-throughput technologies enable the Biology research community to acquire massive volumes of data, generated from various in vitro and wet-lab experiments. Decoding what is concealed within, is the key to solve mysteries behind biological systems and unravel the causative factors of diseases. These data contain all the answers to the continually arising biological questions. The proliferation and the growing complexity of such data introduce various challenges in terms of their retrieval, storage, management and analysis. Bioinformatics is an interdisciplinary field emerged in early 1950s to address those key concerns and facilitate effective life science research. In brief, it engages in applying Computer Science and Informatics to build pipelines, tools, techniques, algorithms and computational models for the purpose. The name of the field itself is an umbrella term, coined in 1970.

Ever since Watson and Crick suggested the double-helix structure of DNA in 1953, many important molecular biological discoveries were made in parallel to the development of computer systems, programming languages and algorithms. The first protein to be sequenced was Insulin. As more protein sequences were getting sequenced, early researchers realized the potential of applying computational methods to go beyond the existing understanding. For instance in 1951, a com-

puter program was developed to determine the structure of a protein for the first time (i.e. Myoglobin). In early 70s, the famous sequence alignment algorithms: Needleman-Wunsch local alignment and Smith-Waterman global alignment were developed. The year 1977 marked a turning point when a highly accurate technique known as Sanger sequencing was introduced for DNA sequencing, outpacing protein sequencing. Later in 1988, genome-level sequencing was initiated with the embark of the Human Genome Project, which was completed in 2004 by publishing the complete human genome. Meanwhile, *Haemophilus influenzae* Rd bacteria genome was the first to be completely sequenced using the shotgun approach, followed by the complete genome sequencing of *E. coli* and yeast. A new turning point was defined when next generation sequencing (NGS) technologies came into the existence in the year 2004. NGS allows fast and efficient sequencing of multiple individual genomes at once, ameliorating the earlier whole-genome sequencing rate. Moreover, microarray technologies were introduced to overcome obstacles present in conventional expression measuring methods, such as single gene expression measurement at a time. However, these novel technologies introduced many more challenges to the field, as the data generated by them tend to contain many errors and noise. [19]



University of Moratuwa, Sri Lanka.
Challenges to The field & Dissertations
www.lib.mrt.ac.lk

Every event across the historical Bioinformatics timeline so far, has resulted in gathering an abundance of data in the form of sequences, structures etc. Many repositories are maintained for accessing existing data and depositing new data. Numerous collaborative projects are in effort to biocurate and to increase the quality and reliability of these data. Moreover, a layered data generation can be observed, as the researchers utilize available biological data to create new data. The Bioinformatics and Computational Biology field is flourishing day by day, introducing more and more novel approaches to analyze the extensive amount of biological data and catch up with their unprecedented rate of generation. Due to the data heterogeneity caused by varying biological aspects or the differences in experimental platforms and methods, a need has arisen for specific quality control and preprocessing, prior to data analysis and mining.

Life scientists encounter diverse biological research problems related to organisms, ranging from single cellular to higher order eukaryotes. It might be either to accurately identify genes; to determine proteins encoded by genes; to annotate proteins with their functions; to solve protein structures; to pinpoint disease causing mutations; or to figure out evolutionary relationships shared by different species. Before utilizing the wealth of heterogeneous biological data at hand to address such problems, a researcher firstly needs to recognize the types of data that are advantageous for the application. Once decided, next thing is to select and retrieve appropriate datasets from reliable data sources. The data will then undergo basic exploratory analysis for decision making regarding further quality control and preprocessing steps. This is essential to remove inconsistencies, noise and to make them adhere to certain standards and rules, as required by the particular research context. Moreover, some studies require comparability, enforcing data normalization in certain ways. The diversity in data sources and biological studies in literature makes the selections much challenging. Finally, the prepared datasets can then be leveraged using different data mining techniques to answer the research question at hand.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

2.2 Overview to Functional Genomics and Proteins

Merriam Webster Dictionary defines Functional Genomics as the branch of genomics that uses various techniques to analyze the function of genes and proteins. Today the field is majorly driven by the use of computational tools and techniques alongside wet-lab experiments, to extract knowledge from vast amounts of genomic and transcriptomic data. This area of study investigates functional roles of genes/proteins, protein-protein/gene-protein interactions, gene expression and their differential expression patterns in the presence of certain conditions [21].

2.2.1 Basics of Proteins

A protein is a macromolecule, consisting of one or more peptide chains made out of 20 types of amino acid residuals. The amino acids are covalently linked as a chain by peptide bonds. The common chemical structure of an amino acid has an amino group (NH₂), alpha carbon and a carboxyl group (COOH), as shown in Figure 2.2.1. R denotes the side chain, which is the only portion that is different by the type. Figure 2.2.2 lists out all 20 amino acid types, along with some of their chemical characteristics. Amino acid side chains have their own physicochemical characteristics such as the electric charge (i.e. uncharged-polar, positively-charged, negatively-charged), hydrophobicity and hydrophilicity. Moreover, a side chain may be acidic (e.g. Asp and Glu) or basic (e.g. Lys, Arg and His). Asn, Gln, Ser, Thr and Tyr are uncharged polar side chains, while the rest are non-polar side chains. The start of an amino acid chain is called the N terminus (i.e. terminated by a free -NH₂ amine group), whereas the end is denoted by C terminus (i.e. terminated by a free -COOH carboxyl group). The amino acid sequence is conventionally written from N terminus residue to the C terminus residue.

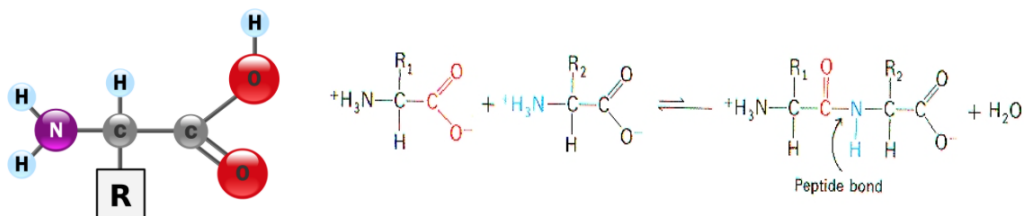


Figure 2.2.1: Amino acid residue [23] and peptide bond formation [24]

Amino acid	Symbol		Formula	Molecular weight (Da)
Alanine	Ala	A	$C_3H_7N_1O_2$	89.09
Cysteine	Cys	C	$C_3H_7N_1O_2S_1$	121.16
Aspartic acid	Asp	D	$C_4H_7N_1O_4$	133.1
Glutamic acid	Glu	E	$C_5H_9N_1O_4$	147.13
Phenylalanine	Phe	F	$C_9H_9N_1O_2$	165.19
Glycine	Gly	G	$C_2H_5N_1O_2$	75.07
Histidine	His	H	$C_6H_9N_3O_2$	155.16
Isoleucine	Ile	I	$C_6H_{13}N_1O_2$	131.17
Lysine	Lys	K	$C_6H_{14}N_2O_2$	146.19
Leucine	Leu	L	$C_6H_{12}N_2O_2$	131.17
Methionine	Met	M	$C_5H_{11}N_1O_2S_1$	149.21
Asparagine	Asn	N	$C_4H_8N_2O_3$	132.12
Proline	Pro	P	$C_5H_9N_1O_2$	115.13
Glutamine	Gln	Q	$C_5H_{10}N_2O_3$	146.15
Arginine	Arg	R	$C_6H_{14}N_4O_2$	174.2
Serine	Ser	S	$C_3H_7N_1O_3$	105.09
Threonine	Thr	T	$C_4H_9N_1O_3$	119.12
Valine	Val	V	$C_6H_{11}N_1O_2$	117.15
Tryptophan	Trp	W	$C_{11}H_{12}N_2O_2$	204.23
Tyrosine	Tyr	Y	$C_9H_9N_1O_3$	181.19

Figure 2.2.2: List of 20 Amino acid types [25]

organismic characteristics. Thus, it is vital to understand their functions.

2.2.2 Structure of Proteins

In general, the structure of a protein refers to the conformation of all of its atoms in three dimensional space to support its existence and function. This is defined in four levels: the primary structure; secondary structure; tertiary structure; and quaternary structure. [26]

1. Primary Structure

This is the amino acid sequence, referring to the linear polypeptide backbone with no shape.

2. Secondary Structure

This is the local folding of polypeptide regions due to the nature of chemical bonds within the chain. The specific local structural shape elements are caused by the intermolecular and intramolecular H bonding of N-H and C=O groups. A local region may get folded into one of the two commonly folding patterns: alpha helix (α -helix) and beta (β) sheet. Alpha helix structure is a spiral conformation in which, the backbone coils around an imaginary helix axis in clockwise direction. Beta sheet refers to a conformation consisted of beta strands which are connected laterally by at least 2 or 3 backbone H bonds, when backbone folds back on itself to make pleats. Random coils with turn and interconnecting loops act as connectors of such folding patterns within the structure. For some proteins, secondary structure is merely a set of alpha helices and thus, they are known as α -helix proteins (e.g. Myoglobin). Similarly there exists β sheet proteins as well (e.g. Antibodies, T cell receptors). Figure 2.2.3 shows example structures for the two protein types. However, many proteins have both α -helices and β sheets. There are 2 types of beta sheets, parallel beta sheets and anti-parallel beta sheets. A parallel beta sheet is formed by two beta strands, running in the same direction and are held together by hydrogen bonds between them. If two beta strands that run in opposite directions are held together by hydrogen bonds, it will result in forming an anti-parallel beta sheet. [27] Figure 2.2.4 from [27] illustrates the difference between the two types.

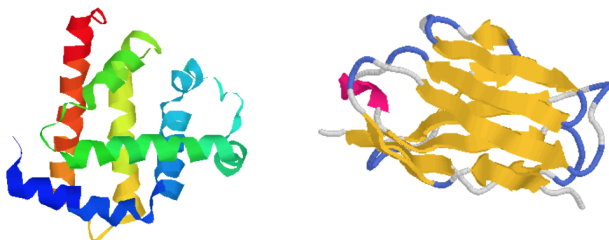


Figure 2.2.3: Example alpha helix protein (right) and beta sheet protein (left)

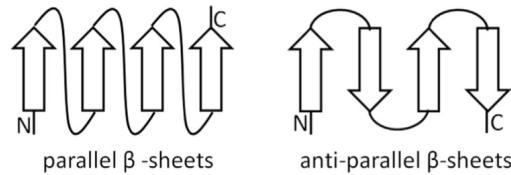


Figure 2.2.4: Parallel and anti-parallel beta sheets [27]

3. Tertiary Structure

This refers to the three dimensionally folded structure of the peptide chain into a specific conformation. Many forces (e.g. polar/non-polar interactions, hydrogen bonds, van-der-waals forces, ionic interactions) act to hold the peptide chain in this final configuration. This should be stable (i.e. having the lowest energy state) under the corresponding physiological conditions, for a peptide chain to function as a protein. A folded single peptide chain may or may not be independently functional. For becoming functional, it may require to get into the next level of structure.

4. Quaternary Structure

Quaternary structure describes the structural system of two or more peptide chains which chemically bond to form a protein complex. The individually folded chains (i.e. protein subunits) fold again into a structure in 3D space, allowing the required inter-chain physical interactions. The quaternary structure explains how different subunits pack together to form the overall protein structure. For instance, Haemoglobin protein has 4 subunits (i.e. 2 alpha chains and 2 beta chains).

A protein structure can be visualized in terms of its space filling view (i.e. with actual size and location of each atom), as well as of its secondary structure view (i.e. polypeptide chain presented as a ribbon to show the locations of alpha helices and beta sheets). The 3D protein folding is not solely based on the sequence, as there are other external factors in the cellular environment which contribute to the final conformation of a protein. For instance, the shape might depend on the proteins' localization (e.g. cytoplasm localized, membrane localized etc.). [26]

Experimental protein structure determination through methods such as X-ray crystallography and Nucleic Magnetic Resonance (NMR) are highly expensive in terms of time and cost. Commonly used X-ray crystallography requires the protein to be crystallized prior to experimentation. Moreover, this technique cannot capture the structural variances caused by the dynamic nature of proteins, when they constantly undergo conformational changes. Furthermore, protein crystallization is somewhat difficult, especially for membrane proteins. Membrane proteins exist in lipid environments and their shape changes when they are not in the original environment, leading to incorrect results for structure determination. Thus computational structure prediction approaches are preferred. [20]

2.2.3 Protein Folding

Folding structure of a protein is very important, since it has a key impact over the gaining of the proteins' ability to perform the intended functions. The physicochemical characteristics of its amino acid side chains affect the way they participate in gaining a stable fold. For example, the hydrophilic residues fold in such a way that the hydrophobic amino acids do not get exposed to H_2O . Those characteristics are also important for the type of functions they perform. Due to the large number of possible protein primary structures and the variation of bond angles between amino acids, theoretically there should be a large number of different folds, making up a huge structural space. However, due to the physicochemical constraints, nature has limited the possible protein folding space. Thus, even two unrelated proteins may fold into similar 3D structures. The same folding type might be reused again and again to perform completely new functions. [28]

2.2.4 Protein Motifs

Protein motifs are of two types: sequence motifs and structural motifs. Sequence motif is an amino acid sequence pattern that is widespread and conjectured to have a biological significance. Structural motif is a super secondary structure, defined by the connectivity between secondary structure elements (alpha helices

and beta strands). It is formed by the folding of a consecutive sequence region in the primary structure. Some common examples for protein structural motifs include beta hairpin, helix-turn-helix and helix-loop-helix. A protein domain can have a combination of protein motifs. [28]

2.2.5 Importance of Protein Structure Determination

Protein structure has quite a strong relationship with the functions it is intended to perform. Therefore it is a key to understand the detailed functional mechanism of a protein. Besides, it is said that the structure is more conserved than the sequence. Certain substructures and motifs may give clues about the function. For instance, the helix-loop-helix structural motif plays an important role in DNA binding and thus, it can be used to characterize transcription factors. However in overall, there are very common structural motifs that cannot be used for distinguishing proteins with respect to their functions. In other words, the same motif may appear in proteins with dissimilar functions. Hence, the structure should be carefully exploited, for instance, by a powerful evidence in functional inference.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Another important aspect of protein structure usage is in the context of drug design. Drugs are usually small molecules that interact with proteins. The purpose of a drug might be to get binded to a protein and disrupt or shut down its function. Since this is a physical interaction between the targeted proteins and the drug in three dimension, it is required to know the exact protein structure in order to design the proper drug.

2.2.6 Protein Domains

A domain is a stable, independent folding unit of a protein, formed by a segment of the corresponding polypeptide chain. It is usually responsible for a single, distinguishable function of the protein. DNA binding domain site, catalytic sites in enzymes and ligand or other protein binding domains are some examples for protein domains. A domain is independent because they are often cloned, expressed

or purified independently of the rest of the protein. In some cases, each domain in a protein is encoded by a separate exon in the gene. A protein can contain several domains (e.g. IE0T with 12 domains) or only a single domain (e.g. Myoglobin and cytochrome complex). Usually the proteins from the same family have the same set of domains. It is also possible to fuse several known domains artificially into a protein molecule, creating a chimeric protein. [29, 30, 31]

2.2.7 Protein Families

Protein family is a group of proteins conjectured to be sharing a common evolutionary origin, reflected by their related functions and similarities in sequence or structure. Protein families can be organized in a hierarchy. When the related sequences of a family are aligned, a consensus sequence (i.e. a sequence signature) may be identified, reflecting a domain or a motif. The existence of a particular motif or domain could give a signal of common functional families. [32, 33]

2.2.8 Origination of Proteins

Gene Expression is the process by which the proteins are originated within a cell. This is often known as the central dogma in Molecular Biology. Figure 2.2.5 gives an overview to the process.

Genes are the coding regions located on the genome (i.e. DNA) and they en-

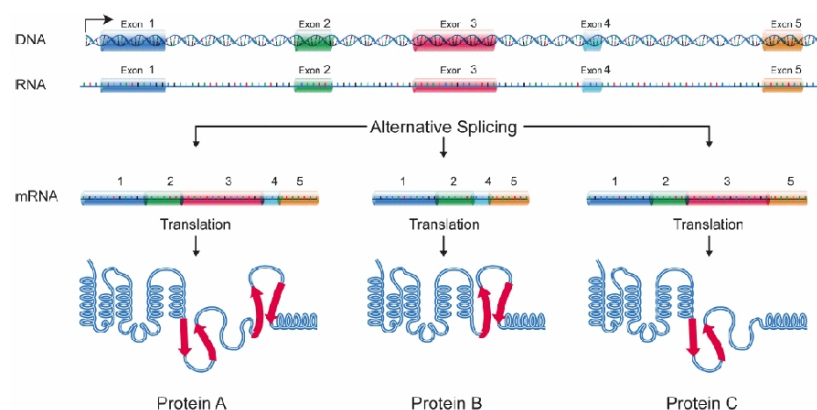



Figure 2.2.5: Gene expression [34]

code information necessary for the construction of a protein. During the process of gene expression, a gene is transcribed into a messenger RNA (mRNA), which is then translated into an amino acid chain. This peptide chain might be an independent protein or a subunit protein of a protein complex. In DNA translation, the template strand is read from 3' to 5', while mRNA is constructed in 5' to 3' direction. An intermediate step called splicing is carried out when split genes are present (mostly in higher order organisms). A split gene is a gene where non coding segments called introns appear in between the biological information coding segments called exons. Splicing is done in order to discard introns and assemble only the exons into a messenger RNA. Sometimes, a phenomenon known as alternative splicing can happen, where a different combination of exons gets assembled to synthesize a different protein at each expression.

Even though every cell of an organism contains the same genome, not every gene is expressed. This is the very reason for different cell types to act differently. For example, a neural cell does not act in the same way as a stem cell does. The gene expression could also differ from stage to stage within the cell life cycle, according to the necessity of protein types. It can even happen in response to a certain environmental condition or in the presence of a certain substance. For instance, the plant cells switch-on genes whose products are related to photosynthesis, in response to light [35]. This is gene expression regulation and there are specific types of proteins that engage in this control mechanism, by binding themselves to the genome. This ensures that only the required proteins are synthesized everytime. However, some genes are expressed all the time, since their products are continuously required for certain cellular processes such as metabolism pathways. They are commonly known as housekeeping genes. Irrespective of the gene expression regulatory mechanism, the entirety of proteins in existence throughout an organisms' life cycle is referred to as its proteome [35].

2.3 Functions of Proteins

The definition of protein function is somewhat vague because, a broad range of biological features at different levels of abstraction is required to describe a protein function. The protein functional space represents how proteins act to form complex cellular functional components, for realizing the genotype into the phenotype. Thus, Bork et al. [1] describes a functional aspect through a hierarchical representation of molecular, cellular and phenotypic functions. At the bottom level of the hierarchy, a protein carries out elementary molecular functions such as ligand binding, catalytic activity and conformational changes. In the next higher level, the collective action of a set of proteins drives cellular functions such as metabolic pathways and signal transduction cascades. The entirety of such physiological sub-systems and their interaction with various environmental factors determine the phenotype (morphology, physiology and behavior) of an organism.

 A protein obtains the appropriate structure and chemical characteristics for its intended functions as per the instructions given by its corresponding gene. In overall, different types of proteins are responsible for different functional contexts. While some proteins regulate the gene expression, some involve in different stages of the gene expression process. For instance, DNA transcription is done by RNA Polymerase: a protein in the form of an enzyme. The mRNA translation is done by ribosome: a protein complex known to be the protein synthesizer in a cell. Proteins can have work inside the cell, as well as outside the cell. For instance, some cells secrete proteins into their surrounding extracellular fluid.

Following are some distinct functions, that various protein types are responsible for.

- Constitution of tissues and organs (structural proteins)
- Facilitation and catalysis of biochemical reactions (enzymes)

- Maintenance of the cellular environment (transmembrane proteins)
- Signal transduction (signal receptors)
- Carrying out substances throughout the body (e.g. Haemoglobin)
- Muscle contraction (e.g. Myosin, Actin)
- Immune system (antibodies)
- Physiological process regulation and growth control
- Transcription, translation and expression regulation (transcription factors)

2.4 Importance of Protein Function Annotation

The entire well existence of an organism depends on its proteins, as they are responsible for performing cellular, molecular and biological functions that are required to maintain a flawless and healthy biological system. Even though it is the genome which encodes the instruction manual for building-up and maintaining an organism, proteins are the actual workers to implement it. Hence, the causative factor for a certain disease might be the changes in gene expression regulation (overexpression/underexpression/inhibition of expression), protein misfolding or due to a mutant protein.

For instance, Sickle cell anaemia is caused by a nonsynonymous mutation (i.e. point mutation that alters the amino acid sequence of the protein) in Haemoglobin. Haemoglobin is the Oxygen carrier protein in red blood cells, by binding O_2 molecules to itself. The steady supply of O_2 is maintained by red blood cells which circulate around the body, delivering O_2 from lungs to the tissue cells. The disk shape of red blood cells containing the Haemoglobin makes them to be flexible in moving through large and small blood vessels. However, the cells containing sickle Haemoglobin are of the shape of a crescent and thus, they become inflexible. They can easily be stucked into blood vessels, causing a blockage in the blood cell flow. The resultant poor O_2 delivery could lead to organ damage, chronic ongoing pains and severe pain attacks. Also sickle blood cell life span

is lower than normal cell life, resulting in an anaemia condition in which, the number of red blood cells in blood is lower than the normal. [6]

The reason behind a certain disease condition can be due to the malfunctioning of an already known protein, an already discovered but functionally unknown protein or an undiscovered protein. In order to find drug targets for preventing, curing, controlling or managing such a condition, it is important to know the exact proteins which are in effect and what their functions are. Hence it is necessary to conduct experiments as to discover the protein functional factors behind regularities/irregularities in phenotypes, and to understand how they relate to disease origination, progression and development. This could help to design and develop the right kind of drug to be targeted at the responsible protein, or the treatment procedure to control and manage the disease condition.

Today, there are many proteins which have been identified as biomarkers (i.e. measurable indicators of the severity of the presence of some disease state). For instance, HER2 receptors are receptor proteins in breast cells. Their task is to control the healthy growth of a breast cell. However, this is a protein encoded by an oncogene (i.e. a gene with a potential to cause cancer) called ERBB2 gene. In some breast cancer patients, this gene is amplified, resulting in HER2 protein overexpression. Such case is known as HER2 positive breast cancer and they tend to grow faster and be more likely to spread and come back compared to HER2 negative breast cancers. Hence this protein has become an important biomarker and a therapeutic target. HER2 positive patients are given specific kinds of medication. A common one is Herceptin which attaches itself to HER2 receptor proteins and blocks them from receiving growth signals. This could help to slow down or even stop the growth of the breast cancer. [5]

Further identification of such biomarker proteins rely on protein function annotation. Thus, identifying and annotating them with their set of functions is an extremely important task for further understanding of biology.

2.5 Biological Data Sources for Functional Genomics

The recent advancements in omics technologies have produced an abundance of data, which can be greatly leveraged for in silico protein function prediction. Genomic data enables sequence analysis in terms of their homology and other genomic context information such as motifs. Sequence comparison tools such as BLAST can be used to identify homologous protein sequences. Moreover, protein domain data are available from databases such as InterPro, CDD, ProDom and PROSITE. Protein structure data can be obtained from online sites such as PDB (Protein Data Bank), CATH and SCOP. The structure data are usually difficult to be processed and analyzed. Transcriptomic data captures differential gene expression, enabling co-expression analysis. It helps in identifying functionally related genes, due to having similar expression profiles. Stanford Microarray database, ArrayExpress and Gene Expression Omnibus provide gene expression data repository platforms, in addition to individual research study platforms. Interactome data includes physical interactions and genetic interactions. A physical interaction can refer to an interaction between two proteins, either to form a protein complex or for performing a certain function together. However, physical interaction data may often have false positive and false negative interactions. Genetic interactions are evidential of gene pairs which exhibit either a suppression or an enhancement of a phenotype, in the presence of mutations in both of the genes. It would give an indication that the pair is involved in the same biology process. However, relative to physical interactions, only a small amount of genetic interaction data are publicly available due to the limited amount of genetic interaction mapping studies. BioGRID (Biological General Repository for Interaction Datasets) is a well-known public data repository that archives interaction data for model organisms, as well as for humans. Other databases include DIP (Database of interacting proteins), MIPS Mammalian protein-protein interaction database, BINDING, STRING etc. There also exists organism specific databases (e.g. Saccharomyces Genome Database (SGD) and FlyBase) and protein family specific databases (e.g. GPCRDB for G protein-coupled receptors; and BRENDA

for enzymes). [36]

In addition to the above, more data types such as subcellular localization; phylogenetic profiles; transcriptional regulatory networks; gene co-expression networks and biomedical literature can be useful as well. These various biological data types could give a strong insight to protein functions, significantly supporting the functional context analysis through their interrelationships. UniprotKB (Universal Protein Resource) is an ideal resource for browsing many common protein data types at once. It has two main sections: SwissProt and TrEMBLE. SwissProt has 549,008 proteins reviewed and manually annotated using information extracted from literature and curator evaluated computational analysis, whereas TrEMBLE contains 50,011,027 proteins with un-reviewed records that await full manual annotation.

Many of the previously mentioned data sources allow researchers to collaboratively deposit new data, continuously bio-curate them and electronically record their findings. Most of such sites also contain statistics on their data. However, an important consideration should be given to the reliability of data when selecting the data sources. Many electronically annotated data are available for use, but it is always important to retrieve only the bio-curated data.

2.6 Microarray Gene Expression

Microarray technology is a widely used experimental approach to study the expression of an entire genome in a tissue of interest at one go. Many biological researchers conduct different genomic studies with the use of gene expression microarrays in order to analyze gene expression and regulation of a wide variety of organisms under varying conditions (i.e. control vs. treatment/ normal vs disease/ phases of a biological process such as cell cycle/ time stamps). These experiments allow biomarker identification, treatment response analysis, pathway analysis etc. through differential expression analysis, by identifying genes whose

regulation is evidential of their engagement in a certain biological process [18].

A DNA microarray is a chip, having a matrix of microscopic spots on a solid surface. Each spot contains a cluster of oligonucleotide sequences (i.e. many copies of the same genomic DNA sequence), uniquely representing a gene. Affymetrix Yeast Genome S98 array is an example for a commercial microarray used for Yeast microarray experiments. There are different types of microarray experiments for different purposes. For differential expression analysis, dual channel microarray experiments are carried-out, where two conditions are represented by two color channels: Red (R) and Green (G). Figure 2.6.6 gives an overview to the process. For instance, two tissue samples are obtained from normal (i.e. reference/control) and experimental conditions. Then for each sample, mRNA samples are extracted and cDNA is synthesized by reverse transcription, while labeling them with the corresponding fluorescent dye color (i.e. Cy5 for R; and Cy3 for G). The cDNAs are then hybridized onto the microarray, where each cDNA molecule gets binded to the spot containing the corresponding complementary DNA sequence. Then the microarray is excited with a laser to see the R and G fluorescent spots. The amount of fluorescence emitted upon excitation corresponds to the amount

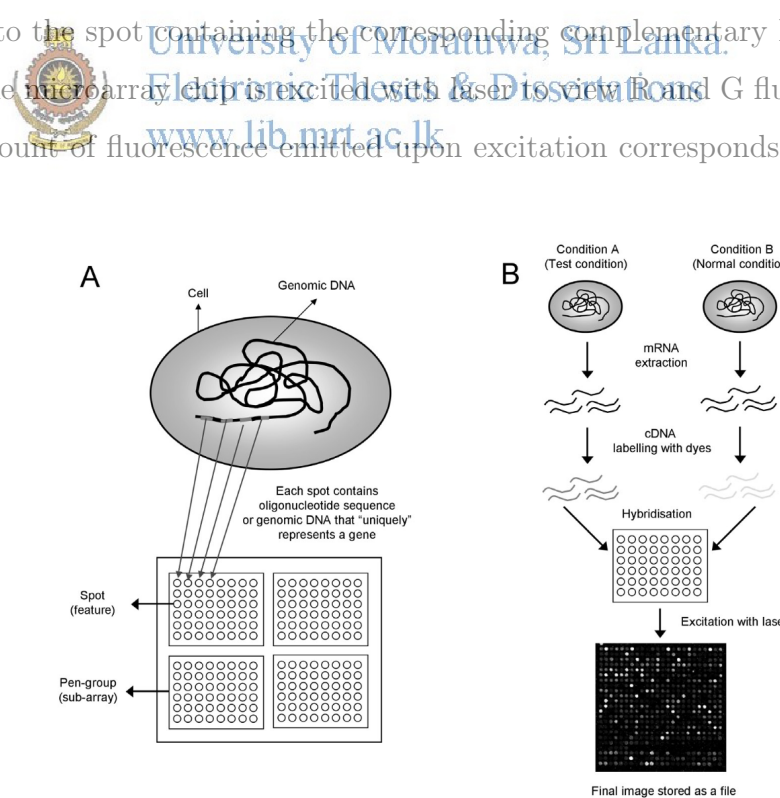


Figure 2.6.6: Microarray Technology [37]

of bound cDNA, which in turn is directly proportional to the initial number of mRNA molecules present for the particular gene. This can be captured as an image using a microarray scanner and can be preprocessed as necessary, in order to obtain a numeric matrix of raw pixel intensity values; each value accords to the amount of fluorescence emitted upon excitation and corresponds to the amount of cDNA in the spot. This raw data set should then be background corrected to remove background fluorescence from the spot signal fluorescence, because the spot signal is believed to be a sum of the fluorescences due to background and the hybridized target cDNA.

Further steps require a quality control and normalization procedure for eliminating systematic biases and artifacts. Such biases account for non-biological variance which masks the true biological variance. Systematic biases can be caused by dye bias (due to differences in heat/light sensitivity or in the efficiency of dye incorporation), varying amounts of starting mRNA in the two samples, variation across replication slides, variation across hybridization conditions, variations in scanning conditions and technicians. These can be introduced at different stages of the experiment (e.g. sample collection; preparation; and hybridization), under different experimental conditions and experimenter bias. Biasing factors are dependent on spotting, scanning and labelling technologies. Such variations result in overestimated or underestimated gene expression level values. For instance, when an experimenter compares the gene expression levels of a gene that should not change in reality between several conditions (i.e. housekeeping genes), he may find that the mean expression ratio of such genes deviates from 1. Also a deviation can be seen from the biological assumption that the majority of the genes are not differentially expressed, and the proportion of the up-regulated and down-regulated genes are almost the same. Thus, to avoid biological assumption invalidations and to obtain correct, reliable measures for detecting true biological differences, the raw data requires a quality control and normalization before moving into further analysis. Such preprocessing can ensure that the systematic variation is minimized and the ob-

served expression differences reflect only the true differences. Gene expression visualizations such as side-by-side boxplots, scatter plots and MA plots help in identifying systematic biases and artifacts. [17, 18, 37, 38, 39, 40, 41, 42, 43]

2.7 Gene Ontology (GO) Functional Classification Scheme

Gene Ontology (GO) [3] is a structured, precisely defined, common and controlled vocabulary for describing the roles of genes and gene products in any organism. It has been developed by the Gene Ontology Consortium. The underlying fact for the formation of such an ontology is the sharing of genes/proteins (i.e. orthologs) among a diverse range of organisms and having common core biological processes (e.g. DNA replication, transcription and metabolism). It was presented as to fulfill the requirement of a common language for annotation, which is to be done in a species-independent and an inter-operable manner between different genome databases. This kind of an ontology enables groupings, comparisons and inferences to be made at different functional granularities [36]. Pandey et. al [32] suggests GO as ideal for annotation due to its wide coverage, standardized format and the hierarchical structure. GO consortium is concerned of three main aspects: the development and maintenance of ontologies; annotation of proteins; and the development of tools that facilitates the creation, maintenance and the use of ontologies.

Gene Ontology database is a relational database with GO ontologies and gene/protein GO annotations. Originally, the GO was formed as a joint project of three model organism databases: FlyBase, Mouse Genome informatics and Saccharomyces genome database. Later on, more databases joined the effort, while extending the focus of coverage from general eukaryotic cell to both eukaryotes and prokaryotes. GO presents three independent ontologies, as means of defining the minimum information necessary for defining gene/protein functions [36]. The structure of each GO ontology reflects a directed acyclic graph G , where each node V refers to a GO term. An edge E between two nodes represent their rela-

tionship. The root ontology term represents each GO domain out of the following three.

Biological Process (BP) A biological process aims to achieve a certain biological objective through an ordered assembly of molecular functions. It is often involved in a physical or chemical transformation. Cell growth and maintenance, signal transduction are some examples for high level processes, while translation, Pyrimidine metabolism are examples for more specific processes.

Molecular Function (MF) This refers to a biochemical activity of a protein. Examples range from broad functional terms such as enzyme, transporter, ligand to narrower functional terms such as adenylate cyclase, toll receptor ligand.

Cellular Component (CC) This refers to the place in the cell where a protein is active. The terms reflect our understanding of the cell structure. (e.g. ribosome, nuclear membrane, golgi apparatus)



2.7.1 Standard Format

Every GO term has a unique, zero padded seven digit identifier (GO:XXXXXXXX) and a term name. The other essential elements are the namespace (which denotes the ontology), the definition and relationships to other GO terms. Apart from that, a set of optional elements including secondary IDs (in case of identical term merge), synonyms, database cross references (pointers to the same entity in other databases), comments and obsolete tag may be present.

2.7.2 Hierarchical Structure

The GO structure is hierarchical in which, the top level nodes are more general and bottom level nodes are very specific. These nodes are connected to other nodes through different biological relationships. Some of the commonly present

relationships are *is-a*, *part-of*, *has-part*, *regulates*, *negatively regulates* and *positively regulates*.

The set of well defined GO terms and their relationships represent the current biological knowledge, as organized into a well defined structure. This hierarchy is somewhat complex because, further to a parent node having multiple children, a single child node can also have multiple parents. For instance, ‘mitochondrion’ has two parents: cytoplasm and organelle, through *part-of* relationship and *is-a* relationship, respectively. Nodes can have any number of and any type of relationships to other nodes. The three ontologies are disjoint in terms of *is-a* relationships. However, *part-of* and *regulates* relationships can occur between ontologies. For instance, the molecular function term ‘cyclin-dependent protein kinase activity’ is *part-of* the biological process ‘cell cycle’. Relationship of a protein to a biological process/molecular function/cellular component is one to many, reflecting the fact that a single protein can involve in several processes; contains domains that carry out diverse molecular functions; and participates in multiple alternative interactions with other proteins, organelles or locations in the cell. [3]



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

2.7.3 Annotations

GO consortium stores GO gene product annotation data in tab delimited plain text. Annotations are contributions from all around the world, with different reliabilities. GO annotation process allows the submission of annotations from two types of groups: bioinformatics database research groups and groups without any long term commitments. First type of contributors are those who take an ongoing responsibility to update annotation data upon policy changes and ontology structure changes. The second type does not have any established database nor funding for long term maintenance.

Each GO annotation should be tagged with an evidence code to indicate how

the annotation was made in the first place. There are 18 different evidence codes under experimental evidence code, computational analysis evidence code, author statement evidence code, curatorial statement evidence code and automatically assigned evidence code. Except for the last type which has only the code IEA (Inferred from Electronic Annotation), all the others are assigned by curators. IEA is assigned automatically without any curatorial judgement. However these evidence codes are not evidential of the annotation quality. [44]

2.8 The ‘mitochondrion organization’ GO Term

The ‘mitochondrion organization’ is defined in the GO Biology Process ontology. The term has 5 ancestors as shown in Figure 2.8.7. This is a level 5 function for which, the GO node has 163 offspring BP terms, including 21 direct child nodes, comprising of 14 *is-a*, 4 *part-of*, 1 for each *negatively regulates*, *positively regulates* and *regulates* relationships. The term has a synonym: ‘mitochondrion organization and Biogenesis’. An example of a protein annotated with this GO term is Dynamin-like GTPase-MGM1. In addition, the following annotations are present for MGM1 through manual curation.

- GTPase activity (MF - GO:0003924)
- membrane fusion (BP - GO:0061025)
- mitochondrial fusion (BP - GO:0008053)
- mitochondrial genome maintenance (BP - GO:0000002)
- extrinsic component of mitochondrial inner membrane (CC - GO:0031314)
- intrinsic component of mitochondrial inner membrane (CC - GO:0031304)
- mitochondrial crista (CC - GO:0030061)
- mitochondrial inner boundary membrane (CC - GO:0097002)
- mitochondrial intermembrane space (CC - GO:0005758)

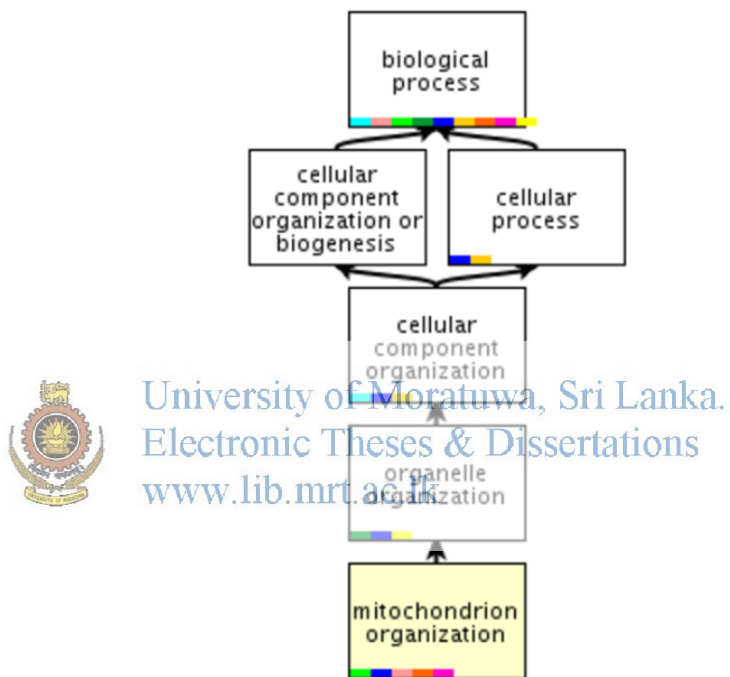


Figure 2.8.7: GO ancestor chart for 'mitochondrion organization' [45]

Chapter 3

LITERATURE REVIEW

This chapter presents a detailed outline over the existing methods for protein function prediction. Numerous computational models continue to be introduced for protein function prediction by many researchers. Often this problem is addressed in the form of gene function prediction as well. The only difference is that the protein space is much more larger than the gene space, due to splice variants and post-translationally modified proteins. Nearest Neighbour (NN) models, network-based models, kernel-based methods, decision tree models, Bayesian approaches and Support Vector Machines (SVM) along with ensemble based approaches are widely used in this context, either to obtain a local prediction of an individual protein function class or a global prediction of multiple protein function classes. Most approaches focus on addressing the multi-class hierarchical classification need and data heterogeneity.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

3.1 Homology based Protein Function Prediction

Earlier approaches considered sequence homology based annotation, which is not always accurate in occasions such as gene duplication [9]. When considering homology based annotation transfers, BLAST is quite an accurate and efficient global prediction approach compared to local prediction approach, to be used with highly similar sequences [46]. Another widely applied concept is guilt-by-association. A function of a protein is predicted through a direct transfer of functions, based on the functions of other proteins that it directly associates with. This association could either be physical or conceptual (i.e. having a shared feature). The functionally known protein is the knowledge donor and the functionally unknown protein is the knowledge acceptor. [36]

3.2 Multi-class Classification and Data Heterogeneity

The recent models focus on hierarchically consistent function annotation in the context of multi-class classification. There are two ways to work around this context: local prediction (also known as flat prediction) and global prediction. Local prediction simply involves in building a binary classifier for each function class. On the other hand, a single classifier is built for global prediction, where all functions of a protein can be predicted at once. The problem also requires hierarchically consistent classification which adheres to the True Path Rule (TPR). The True Path Rule specifies that an annotation for a class in the hierarchy is automatically transferred to its ancestors, while genes/proteins unannotated for a class cannot be annotated with its descendant classes [47]. Various models have been proposed to enforce this hierarchical consistency in gene/protein function classification.

Further, a wide variety of biological data such as subsequences (i.e. motifs and domains), amino acid features (i.e. molecular weights, isoelectric point, sequence length, residue occurrences etc.), protein structure, subcellular localization, pairwise interactions, gene expression, phylogenetic profiles, post-translational modifications and biomedical literature, can be leveraged in the context of protein function prediction. Valentini [47] emphasizes on the need of integrating multiple data sources through methods such as an ensemble training with a base classifier on each data source, a classifier training based on a weighted sum of kernels or a simple concatenation of vectors from different data sources for training a single classifier. This section describes some of the existing studies that address the data heterogeneity in this problem context.

3.2.1 A True Path Rule Hierarchical Ensemble Approach

Valentini [48] proposes a TPR hierarchical ensemble approach for multi-label, multi-path, tree structured hierarchical classification based on the true path rule. The methodology firstly constructs a local base classifier, independently special-

ized for each class in the hierarchy. Each base classifier outputs a probability of a gene belonging to the corresponding class. At the next stage, those base models exchange information among them to arrive at a global consensus ensemble decision, by correcting the local probabilities. This information flow is two-way asymmetric in order to grant a node for influencing its ancestors upon its positive prediction, as well as for influencing its offsprings upon its negative prediction. The proposed approach scans the tree structure of classes in a bottom up fashion, with a per level traversal. At each node scan, the algorithm checks if it is a leaf node. If so, the local probability is presented as the consensus probability for that particular class. If it is an internal node, firstly all the child nodes with positive predictions are considered and the consensus probability is computed based on both local probability and the child node probabilities. If the decision is negative, all child nodes are set to negative. The prediction is taken to be positive if the probability value is above a predefined threshold. In case of a positive local prediction for the current node, the consensus global estimate is computed through Equation 3.1, where Φ set refers to the child nodes of the current node which demonstrate positive predictions for the instance. In case of a negative local prediction for the current node, the decision is propagated to its sub tree. [47]

$$p_i(x) = \frac{1}{1 + |\Phi_i(x)|} \left(\hat{p}_i(x) + \sum_{j \in \Phi_i(x)} p_j(x) \right) \quad (3.1)$$


On the other hand, a hierarchical top down approach [47] classifies an example x with label y_i , where $d_i(x)$ is the output at node i and $root(T)$ denotes the set of nodes at the first level of the tree T .

$$y_i = \begin{cases} d_i(x) & \text{if } i \in root(T) \\ d_i(x) & \text{if } i \notin root(T) \text{ AND } y_{par(i)} = 1 \\ 0 & \text{if } i \notin oot(T) \text{ AND } y_{par(i)} = 0 \end{cases}$$

In [48], the author evaluates the performance of three different ensembles (i.e. flat ensemble, hierarchical top down ensemble and this true path rule hierarchical bottom up ensemble), with 2nd degree and 3rd degree polynomial SVMs as the

base classifiers. The approach has been tested for yeast gene function prediction using 200 functional classes from FunCat (forming a tree of depth = 5) and using four types of biomolecular data: protein domains; phylogenesis; gene expression; and protein protein interaction data. The significant results have been observed only with gene expression data. The author also claims that the results are not significant since the class imbalance problem was not addressed.

Further, this TPR ensemble has been evaluated on *S. cerevisiae* upon seven biomolecular data sources [47]. The method is able to enforce consistency in both GO and FunCat. Valentini [47] also introduces a variant of the TPR called weighted TPR (TPR-w), in order to balance the local prediction with positive predictions from offsprings through a weight value. In this approach, if the weight is 1, the node decision will solely depend on the local predictor. Otherwise, the prediction is shared proportionally between local predictor and the set of its offspring predictors, in values w and $(1-w)$, respectively. Here, the Equation 3.1 gets modified into Equation 3.2



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mpu.ac.lk

$$p_i(x) = w p_i(x) + \frac{1-w}{|\Phi_i(x)|} \sum_{j \in \Phi_i(x)} p_j(x) \quad (3.2)$$

The final observation is that the hierarchical methods largely outperform flat methods. TPR-w performs better with both linear and gaussian SVMs. However, Top down and TPR results are not significant. Flat methods tend to have the highest recall, whereas the top down method results in the highest precision. TPR-w performance is in the middle. It is also possible to tune the TPR-w by changing the weight. With large weights, it has improved precision, whereas with small values, it has improved the recall. The TPR concept can also be applied with probabilistic classifiers other than SVMs.

3.2.2 Hierarchical Classification of G Protein-Coupled Receptors

Davies et al. [49] proposes a selective top down classifier for G Protein-Coupled Receptors (GPCR) classification, with the objective of incorporating hierarchical

structure relationships between predicted classes. Even though this involves in assigning a GPCR protein into the correct GPCR family, the basic requirement of hierarchical consistency is the same. The focus has been on human GPCR, fungi (for class D) and Dictyostelium (for class E), with up to three levels in the hierarchy (i.e. 8354 protein sequences in 5 classes A-E, 40 classes at subfamily level and 108 classes at sub sub family level. Class F has been ignored due to the low number of sequences).

The model is a tree of classifiers that reflects the structure of classes. The root classifier is trained upon all training data. The sub classifiers are trained upon specific train data subsets. In the presence of an unknown GPCR protein sequence, it will be firstly classified by the root classifier and then passed down to the appropriate next level classifier, until it is assigned with all possible sub family class labels. At each node, the training data is split into a train subset and validation subset randomly. Then, eight different classifiers (Naive bayes, Bayesian net, SVM, Nearest Neighbour model using euclidean distance, Decision list, decision tree, Naive bayes tree, Linear layer neural net with back propagation, AIRS2 classifier based on artificial immune system paradigm and conjunctive rule learner) are trained upon the train subset data and are tested upon the validation subset data. The node classifier is selected to be the one which demonstrates the highest classification accuracy. Then the selected type of classifier will be trained upon the original train dataset. The authors have compared the method performance with standard top down approach and the results have showed that this selective top down classifier performs better. 3-nearest neighbour classifier has been chosen at the top level. Moreover, the comparisons with three publicly available GPCR classifiers have also showed results in favour of this approach. However, according to the authors, a notable disadvantage in this approach is that the misclassified instance at one level has no possibility of being correctly classified at deeper levels. Thus, the misclassification rate increases with the depth.

3.2.3 Predictive Clustering Trees and their Ensembles

Unlike the common, local approach of constructing multiple binary classifiers, CLUS-HMC [50] method uses a single tree structure known as the predictive clustering tree (PCT) for multi-class, multi-label classification. The objective is to predict all class labels associated with a gene at once, in a hierarchically consistent manner.

A PCT regards the decision tree as a hierarchy of clusters. It can be constructed by a standard top down decision tree induction algorithm (i.e. C4.5, CART). The root node of a PCT represents a single cluster, containing all training examples. The root cluster is then recursively partitioned into smaller clusters. A partitioning criteria which depicts the attribute splitting criteria in a decision tree, is used to split a node cluster into several clusters. The best split is chosen to be the split which results in a significant maximization of the variance reduction (as measured using a statistical F test). The key concept is that the maximum variance reduction could maximize the cluster homogeneity. If no test split provides a significant variance reduction, the node is marked as a leaf.


In this method, the class memberships are presented as a binary vector for each example. For instance, let $[0, 0, 1, 1, 0.1, 1, 0]$ be a binary vector of size N . The i^{th} position specifies whether the example belongs to class i or not (i.e. 1 or 0, respectively). The arithmetic mean of such binary vectors is an aggregate binary vector, with each position giving out the proportion of total examples belonging to the class corresponding to that position. The variance of a set of examples S (given in Equation 3.3) is defined as the average squared distance between each examples' class vector v_k and the sets' mean class vector \bar{v} .

$$Var(S) = \frac{\sum_k d(v_k, \bar{v})^2}{|S|} \quad (3.3)$$

The distance measure is the weighted euclidean distance as calculated by Equation 3.4. The weight is chosen according to the node depth in the hierarchy, in order to give a higher importance to top level similarity than the lower level similarity. Also $w(c) = w_0 \cdot avg_j(w(p_j(c)))$, where $p_j(c)$ denotes the j^{th} parent of class c and $0 < w_0 < 1$.

$$d(v_1, v_2) = \sqrt{\sum_i w(c_i) \cdot (v_{1,i} - v_{2,i})^2} \quad (3.4)$$

This weight value contributes to ensure the structured nature of class labels according to the hierarchy. In this tree, each node keeps an account of the mean class vector of all corresponding cluster examples. At the prediction phase, when a new instance arrives at a leaf node, a threshold is applied to the mean vector to determine the class vector for that particular instance. Moreover, whenever a class is predicted, its super classes are also predicted at the same time for ensuring hierarchical consistency.

 University of Moratuwa, Sri Lanka.
 Schietgat et al. [51] extends CLUS-HMC approach into an ensemble of PCTs called CLUS-HMC-ENS, through bagging. Training examples are randomly sampled with replacement in order to obtain the bootstrap samples of the same size as the train set, upon which a PCT base classifier is trained. The base predictions are combined by taking the average of all n class vectors predicted by n base predictors in the ensemble. The threshold is then applied to arrive at the final decision. When compared to CLUS-HMC, the performance is better and the results also indicate that it performs particularly better for the less frequent classes. However, the training time of the model is considerably high due to the extra burden of bagging on top of PCTs.

3.2.4 Bayesian Hierarchical Correction

Barutcuoglu et al. [52] presents a Bayesian framework for achieving the hierarchical consistency over a local prediction approach for *S. cerevisiae* gene function prediction. It comes as a way of collaborative error correction over all nodes.

Firstly an SVM based flat prediction is done and secondly a hierarchical Bayesian correction over the hierarchically inconsistent predictions is performed in order to improve the accuracy. The idea is to find the most probable set of consistent class label set, given the inconsistent class label set. Equation 3.5 gives the equation for posterior probability calculation. \hat{y} denotes an output by flat prediction (possibly inconsistent class labels), whereas y denotes a most probable output. Z is a constant normalization factor.

$$P(y_1, y_2, \dots, y_N | \hat{y}_1, \hat{y}_2, \dots, \hat{y}_N) = \frac{P(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N | y_1, y_2, \dots, y_N) P(y_1, y_2, \dots, y_N)}{Z} \quad (3.5)$$

In this proposed framework,

- y nodes are the binary valued hidden nodes, representing actual membership to the class. They are conditioned on their child nodes.
- \hat{y} nodes are the corresponding observed classifier outputs for y nodes. They are conditioned on the corresponding y nodes.

$$P(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N | y_1, y_2, \dots, y_N) = \prod_{i=1}^N P(y_i | ch(y_i)) \quad (3.6)$$

Hierarchical constraint imposition is made by having the appropriate conditional probability values. A label will be 1 if any of its child nodes is 1. $P(\hat{y}_i | y_i)$ can be estimated by validation. It is assumed that the distribution of classifier outputs for both positive and negative examples separately is Gaussian. $P(y_i | ch(y_i))$ can be inferred from train data by counting.

In [52], an ensemble of 10 hard margin linear SVM classifiers are learned over 10 bootstrap samples, for each class. The output is taken to be the median. The focus has been over the GO Biological process ontology with a coverage of 105 selected GO terms. The model evaluation has shown a performance increase in terms of AUC for 93 nodes out of the 105, while implying larger improvements at deeper nodes. The authors state that this is a generic ensemble method to be used with any type of base classifier other than an SVM.

3.2.5 HML Boosting

HML Boosting was introduced by Alaydie et al. [53] for outperforming local protein function prediction. It is another ensemble approach that exploits hierarchical class dependencies for performing class membership inconsistency correction. Two versions: top down and bottom up prediction approaches, have been evaluated in this context.

HML Boosting is a recursive algorithm from which, the hierarchy of nodes is traversed. At each node, it checks whether it is an internal node or not. If so, an ADABOOST.MH binary model is trained for each of its child nodes. Otherwise, it will skip to the next node, as the authors state that there is no need of a classification if the leaf nodes are reached. The base classifier is a decision stump. At the classification phase, the prediction is made based on the local prediction of that class, as well as on the descendant node predictions. Equation 3.7 is the formula for computing the consensus probability at each node.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

$$P(x) = \frac{P_{local}(x) + \sum_{children(x)} P_{child}(x)}{|children(x)|} \quad (3.7)$$

Also each node classifier filters out the unsuitable examples from going down to the lower levels of the hierarchy, by blocking the negative label assigned genes. Thus, during the classification phase, a node classifier will only be presented with instances that have been classified as positive by the parent node.

The authors have trained the ensemble over yeast bio-molecular data from [47] and evaluated the method in comparison to flat ADABOOST.MH multi-label classifiers. They have also analyzed the performance of HML Boosting at each FunCat hierarchy level, covering the top 4 levels. With the increase in the number of boosting iterations, HML Boosting outperforms flat classification with respect to all data sets. Moreover, HML Boosting top down approach has outperformed the other in most cases, in terms of precision and F measure. However, flat

methods tend to show best results in terms of the recall. Moreover, the top down HML Boosting accuracy gets reduced, when moving down from higher levels to the lower levels. The main reason for such behavior is the effect made by higher level classifiers on the lower level classifiers with regards to the instance propagation; especially in the presence of a higher level classifier misclassification. The authors also state that they are interested in minimizing this misclassified instance propagation by developing a mechanism to correct parent node misclassification at a child node.

3.2.6 Label Similarity Incorporated kNN Algorithm

Pandey et al. [54] presents a way of incorporating the distant functional relationships, by not only attempting to leverage the hierarchical structure. This is because the functional relationships might not always be hierarchical. The authors recognize this as a very challenging task than the hierarchical consistency enforcement problem, since there are many types of relationships between functional class nodes that exist just the ancestor-descendant relationships. The key idea is to identify the semantically similar GO terms and use the information for better prediction. Semantic similarity in an ontology can be measured using an appropriate method (e.g. an information theory based measure such as Lin's) on the basis of their relative position in the hierarchy and the associated content, or both. In this research, the authors present a method that firstly evaluates the semantic similarity between the nodes of the ontology, and then quantifies and incorporates the interrelationships into a weighted variant of the K nearest neighbour classifier. The focus has been made upon functional classes from GO Biology process ontology.

Lin's measure is used to form a label similarity matrix with dimensions $|\text{Labels}| \times |\text{Labels}|$. The computed raw label similarity matrix is then preprocessed by applying a filter to avoid the observed significant deterioration of label similarity incorporated classifier performances. For each class label, the experimenters have

used leave-one-out cross-validation and grid search over the interval [0,1] in 0.05 steps, in order to decide on a filtering threshold based on the AUC score. The threshold producing the highest AUC is chosen and all the labels with lower similarity are converted into 0s. Another matrix: the likelihood score matrix with dimensions $|\text{Proteins}| \times |\text{Labels}|$, is derived using direct kNN. Finally, the product of the likelihood score matrix and the label similarity matrix is taken as the final likelihood score matrix.

The evaluations have been conducted by constructing classification models for 138 functional classes that have no parent-child relationships among them and are convenient for testing in wet lab. A comparison has been made between the Label similarity incorporated kNN and the base kNN. The results have shown an average improvement for all classes, as well as for each class, without suffering a significant loss of accuracy at any of the class nodes. The primary target to improve the prediction accuracy of classes with insufficient training data, has been achieved through this method. The concept can be extended to incorporate all the useful GO classes for prediction, by computing the label similarity matrix of $|\text{Labels}| \times |\text{all GO terms}|$.

3.2.7 SVM based Ensemble Framework

Guan et al. [55] presents an ensemble approach for gene function prediction in the context of GO, focusing on unicellular organisms (i.e. *S. cerevisiae*), as well as multicellular organisms (i.e. lab mouse). Their model has been applied successfully to reveal functions of a novel Mitochondrial protein, which got experimentally confirmed as well. The proposed ensemble consists of 3 different classifiers as follow.

Bagged SVM classifier for each GO term of interest: A set of linear kernel SVMs are trained for each GO term, upon bootstrap samples that were derived from a single dataset, which is comprised of different datasets. As the direct concatenation of all datasets may give more weight to datasets

with more features, each of their contributions are normalized to the Gram matrix. That is done by separating all feature vectors from different datasets for a particular gene, normalizing each vector by its size, and then concatenating all normalized features for each gene in a single input matrix.

Bayesian hierarchical combination of SVMs: The bayesian network introduced by [52] is used to correct the flat classification predictions to impose hierarchical consistency across the GO terms. For the feasibility of bayesian network inference, GO ontology is divided into subgraphs for each term using one of the following two methods: HIER-MB and HIER-BFS. The subgraph preserves the local neighbourhood around each GO term.

- **HIER-MB**

For each node, its markov blanket (i.e. the set of nodes containing the parents, children and other parents of the children) is used as the subgraph to construct a bayesian network.

- **HIER-BFS**



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

From each node, a breadth first search is performed to include all descendants upto maximum of 30 GO node terms in the subgraph.

After obtaining the inconsistent set of GO term predictions, they can be corrected by taking the bayesian network into account.

A naive bayes combination of SVMs, each trained per different dataset

This is a meta classifier in which, for each GO node, SVMs are trained for each bootstrap fold, on each data set. A naive bayes classifier is built on a held-out set to combine the prediction outputs from them. Linear kernel SVMs have been used for the purpose, except for protein-protein interaction data, where a diffusion kernel has been used due to its superior performance.

For each GO term, the above three classifiers are trained and the AUC measures are taken. The best performing classifier for the particular GO term is selected to give out the final prediction. According to the results, majority of the GO term

predictions depend on 2nd or the 3rd classifier. Naive Bayes is more likely to get selected for larger GO terms, whereas the hierarchical classifier performs well for every size. The results have also indicated that the bagged SVM performance is robust across a wide range of GO terms, thus confirming its robustness as a baseline method for the ensemble.

An approach is also presented by [64] for the same purpose of calibrating and combining predictions to come up with a GO topology consistent set of probabilistic predictions. In the presence of heterogeneous data sets, firstly per each GO node, a kernel matrix is computed for each data set. Then an SVM is trained with each kernel of the particular node. After obtaining predictions from those base SVMs, they are combined and calibrated using a collection of logistic regressions. Finally, the calibrated predictions are reconciled as to adhere with the GO structure.

3.2.8 Hierarchical Bayesian Integration Algorithm

Alaydie et al. [56] modifies the approach in [53] to a Hierarchical Bayesian Integration Algorithm (HiBiN) for addressing data heterogeneity, through the integration of different data sources using Bayesian Reasoning. The authors attempt to address the source diversity problem, multi-label classification problem and hierarchical consistency in one go.

With the assumption of each dataset being independent, each boosting classifier computes the likelihood of observing a particular gene associated with a certain dataset, given a specific functional class. In HiBiN, firstly the prior probabilities for all classes are computed. Then at traversal, an ADABOOST.MH model is built for each child class of the current class node, per each dataset. The integrated Bayesian posteriors for each child class of the current class are then calculated. The posterior probabilities for multiple independent data sets are integrated and computed by using the Bayes formula.

The method has been compared with flat prediction and also with the hierarchical, single source method applied for each dataset separately. The results have demonstrated that HiBiN provides a considerable improvement over the others. Also there is no significant difference between hierarchical single source vs. flat prediction. Authors have also observed that the performance degradation due to the classifier depth, significantly decreases when data integration and hierarchical constraint is imposed.

3.2.9 Semi Supervised Multi-label Collective Classification

Collective classification performs a joint classification of interrelated instances, unlike the conventional supervised learning which takes unclassified instances independent of each other for classification. Such classifier makes the use of not only the attribute features, but also the relational features. In this context, the data is viewed as a graph and the task is to come up with a function that is capable of predicting the classes of unlabeled nodes, by using the labeled nodes. The basic concept is to make a prediction about an unannotated protein, in the presence of known functional properties of its labeled neighbours.

Wu et al. [57] diversifies an ensemble upon different types of latent graphs that can be built upon both labeled and unlabeled protein instances. They use 3 types of latent graphs: protein-protein interaction latent graph; random walk latent graph; and prediction similarity latent graph (i.e. two nodes are linked if one of them is among the k nearest neighbours of the other). This is a way of exploiting the data representation diversity. In a latent graph, the nodes which are closer to each other often tend to have the same functional labels, whereas the nodes disconnected from each other have different ones. Those latent linkages can be used to perform a knowledge propagation from labeled nodes to unlabeled nodes, in terms of their functional classes. This is because a latent linkage may exhibit a pairwise similarity.

For each latent graph, a base learning model called GM-SMCC is learned. GM-SMCC is a model which is designed to use probabilistic latent semantic analysis with a network regularizer for exploiting network linkages and label correlations effectively, in order to compute a label probability distribution. The authors have not limited the application of this approach to protein function prediction; it has been applied to protein localization prediction as well. The performance of the method has been compared with four baseline methods, including SVM base classifiers (LibSVM with linear kernel). The results show a better performance.

3.2.10 Ensemble based GPCR Class Prediction

Gu et al. [27] focuses on predicting classes for low homologous GPCR data (with just 40% identity) by using an ensemble of 12 euclidean distance based fuzzy kNN classifiers. Fuzzy kNN is a model that combines fuzzy set theory with kNN. The output of each base predictors is a fuzzy membership matrix (for each class membership). The final output is taken to be a weighted output fusion. Here, each fuzzy kNN classifier is trained over pseudo-amino acid composition (PAAC) data representation of proteins with different lambda values. The weight factors of the PAAC have been determined using an Immune Genetic Algorithm (IGA). In addition, the model also uses a hybrid approach of predicted secondary structural features, and approximate entropy as the feature selection method. Figure 3.2.1 is from [27], illustrating their framework.

3.2.11 Transductive Multi-label Ensemble Classification

Yu et al. [58] proposes and presents a Transductive Multi-label ensemble classification framework for data integration. In this method, a kernel is formulated to represent each data source. It is then used for training a graph based Transductive Multi-label Classifier (TMC). TMC is a model that views proteins and their functions as a directed bi-relation graph (e.g. Figure 3.2.2). This graph has 2 subgraphs: a protein graph and a function graph. These bi-relational directed

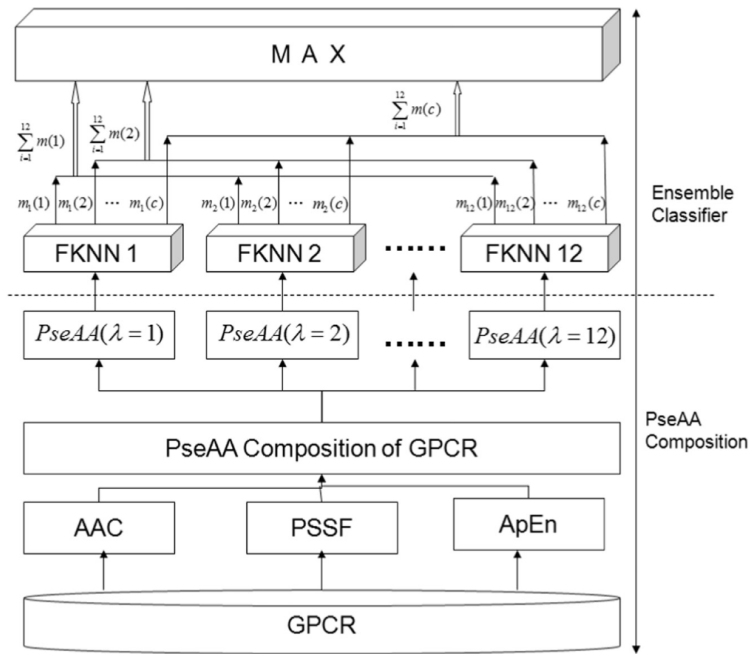


Figure 3.2.1: Fuzzy kNN ensemble model by Gu et al. (2015) [27]

graphs have been used to capture relationships between protein-protein pairs, function-function pairs and protein-function pairs. Thus it is evident that this approach attempts to achieve hierarchical consistency imposition implicitly.



The model firstly transforms each kernel into a directed bi-relation graph and then trains a TMC (graph based multi-label classifier) on each graph. Finally the TMC predictions are combined using a weighted majority vote, by using the confidence of the prediction. For an unknown protein instance to be annotated,

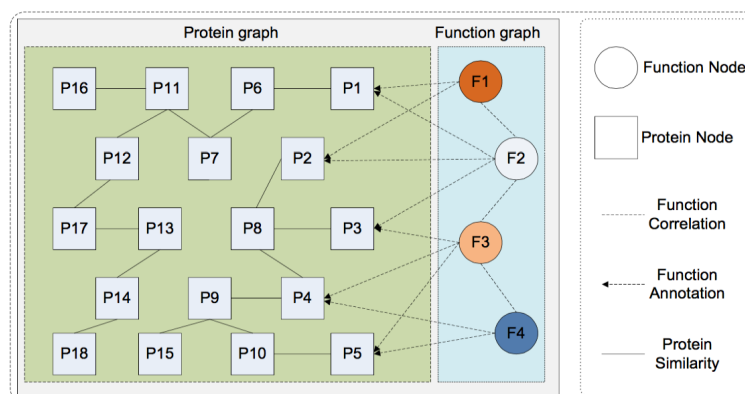


Figure 3.2.2: Directed bi-relation graph [58]

the predicted likelihood vector which is outputted by the approach have to be converted into a hard function assignment. The top k most probable function assignments (i.e. functions associated with the k largest probability scores) are selected for the purpose.

The common and traditional data source integration approach is to represent each data source with a kernel function and integrate the kernels into a composite kernel. In contrast, this method integrates the predictions of each base model, built upon an individual data source. The results have shown that this method outperforms the traditional method. They have also shown that the use of directed graph achieves better accuracy than the use of undirected graph. An advantage of this procedure is that a new data source can be utilized without redoing the entire training process. In overall, the authors have been able to justify the advantage of integrating classifiers instead of data.

3.2.12 MS-kNN for Multiple Data Integration

Lan et al. [59] presents a multi-source k -Nearest Neighbour (MS-kNN) algorithm for multiple data source integration in protein function prediction. It is also an ensemble approach in which, a base k NN classifier is built over a single data source. Three data types (i.e. protein sequence data, protein-protein interaction data and microarray expression data) have been used for the purpose. The base classifier outputs a prediction score by taking the weighted average of its k -nearest-neighbour functions. Then the authors have tested three combination strategies (i.e. averaging, weighted averaging and GO term cluster-specific weighted averaging) to combine base prediction scores, for arriving at an ensemble prediction. Weights for different data sources are obtained by solving a convex optimization problem. The authors conclude that the k -nearest neighbour algorithm is an efficient and effective model for protein function prediction, while being helpful in integrating multiple protein data sources.

3.2.13 Functional Association Network based Approaches

A typical approach for function prediction of proteins or genes is by applying the principle of guilt-by-association. Even though this does not always work well, the diverse association linkages between proteins/genes in terms of different data types could give a good insight regarding their functional context.

This principle is usually used by constructing a functional association network for representing each dataset from a certain biological datasource [60]. In such a network, nodes denote proteins or genes, while the edges denote an evidence of co-functionality as implied by the data. It is often represented by a kernel. The edges can also be weighted according to the intensity of the functional association. Usually the protein-protein interaction networks and gene co-expression data are ideal to be represented through a functional linkage network. And most of the time, the set of functional association networks are combined together in order to generate a composite network. [60] Several approaches have been introduced to utilize functional association networks in protein/gene function prediction.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

GeneMANIA [60] is a gene function prediction server, which focuses on real time protein function prediction with local prediction. It has a concern over the running time and the requirement of updating static databases with new functional association information. For each function of interest, this model builds a functional association network for each data source. Such network is assigned with a positive weight to reflect its usefulness in predicting the function of interest. Then a function specific, composite association network is constructed (using an algorithm based on linear regression) by taking the weighted average of individual networks. Different genomic and proteomic datasets are used for the purpose. There is also a seed list of genes which are known to have the particular function. The next step is to predict the gene functions using a variation of a label propagation algorithm called Gaussian field label propagation algorithm. A label propagation algorithm assigns a score named ‘discriminant value’ to each node

in the network. The score reflects a nodes' degree of association to the seed list of genes that define the given function. A threshold value can be used upon this discriminant value in order to make the prediction of whether the unknown gene has the function of interest or not.

Zhao et al. [61] presents an algorithm for Annotating Genes with Positive Samples (AGPS), while integrating several heterogeneous data sources. The types of data include yeast protein-protein interactions, gene expression profiles and protein complex data. The PPI network is represented as a functional network, having edges with distance values estimated using the Czekanowski-Dice (CD) distance (i.e. an estimate of the functional similarity between a pair of genes). Protein complex data is also represented by a functional network upon computing the CD distances. For gene expression profiles, firstly a binary network is built by calculating the Pearson correlation coefficients (i.e. an edge is added between two nodes if the absolute value of the correlation coefficient between the pair of genes is over 0.7). Then the CD distance is computed over the network to obtain the functional network. Finally, the above two functional networks are merged to obtain the final functional linkage graph. After that, a method known as singular value decomposition is applied for extracting the dominant structure of the graph. It reduces the dimensionality, while removing the noise from data. Then the AGPS algorithm is performed. AGPS is further explained in section 3.3.1. This approach also divides the multi-class classification problem into a set of binary classification problems and computes a radial basis kernel for each SVM classifier. The proposed method has been applied for *S. cerevisiae* gene function prediction with annotation data under 13 general functional classes from FunCat.

3.2.14 BLAST based Local Prediction

Eisner et al. [46] presents CHUGO: a system which exploits the GO hierarchical structure in predicting the GO molecular function of proteins from sequence data. CHUGO also uses local predictors to predict each single GO term. Their

initial focus is on creating a hierarchically aware training set and then utilizing it for reducing the computational cost of classification. In CHUGO, an ensemble of the following 4 classifiers is used as the local predictor for each GO functional class node of interest. They are: an SVM with PFAM features; an SVM with proteome analyst features; Probabilistic suffix trees; and BLAST. A simple majority voting scheme has been used to give the final output of the ensemble. The authors have also tested a weighted scheme by learning weights through SVMs, but have not seen a performance difference from the simple voting scheme. At overall prediction of functions, the positive predictions are upwardly propagated for imposing the True Path Rule.

The method is compared with BLAST-Nearest Neighbour (BLAST-NN) classifier. BLAST-NN is a global predictor, as it is capable of assigning a protein into multiple functions at one go. In the presence of highly similar sequences, BLAST search is quite accurate and computationally efficient when compared to local prediction. Each BLAST hit is assigned a score named the E value, which reflects the similarity between two sequences. A smaller E value indicates a lesser match to the query protein. One important observation made by the authors is that, in case of incorrect nodes being returned by BLAST-NN, they tend to be much closer to the correct nodes. Thus, BLAST can be used to obtain a set of candidate nodes in order to run the local predictors on. In other words, BLAST results can be used as a seed to begin the search for correct annotations. According to the authors, there are two options to decide the seed list.

- **B-N-Union:** taking the union of the top most BLAST hit annotations
- **SearchN:** searching in the neighbourhood of the top BLAST hit annotations

The goal is to run local predictors only for the most likely GO nodes, which in turn will reduce the computational cost.

3.2.15 An Ensemble for ‘mitochondrion organization’ Prediction

Hibbs et al. [12] has carried out an experimental evaluation of an ensemble of three computational methods (i.e. bioPIXIE, MEFIT and SPELL) to predict gene/proteins involved in the ‘mitochondrion organization’ Biology Process. Each method integrates high-throughput data sources and knowledge from GO and SGD. The bioPIXIE and MEFIT methods perform Bayesian integration, targeting genomic data and microarray expression data, respectively. SPELL attempts to identify groups of related genes through a similarity search algorithm over the same microarray dataset. The model results are combined based on their estimated precision. The authors have validated gene predictions using a laboratory technique.

3.3 Selection of Positive and Negative Examples

In supervised learning, the definition of positive and negative examples is important for learning how to distinguish an unknown instance from one class to another. Learning a model with only positive examples will not reach the best approximation of the true hypothesis. This is evident from the fact that, two class SVMs outperform one class SVMs. Also with a small number of positive examples, the model is most likely to underfit. [61]

In general, a positive example is a protein/gene which is known to be belonging to the function class of interest (i.e. the positive class). Negative example for a particular function class is a protein that is known to be not performing, nor engaged in the corresponding function [62]. However, a major challenging aspect of this learning process is the proper and an accurate definition of positive and negative examples. While a positive example could be selected by intuition, the same is not quite possible for negative example selection, as the current annotation databases mostly and explicitly provide positive examples, but rarely stores negative examples [61, 62]. This is because, if a protein is not annotated with a function, it is either a true negative example or a protein yet to be discovered

and to be annotated under the function of interest. Moreover, due to the multiple annotations of a single protein, negative selection becomes more complex [63].

Negative example verification is much difficult due to experimental constraints [62]. Moreover, with varying reliabilities in existing annotations (i.e. electronically inferred annotations), there can be false positive examples as well. Thus, a learning model could suffer from both false positive and false negative examples [62]. Hence, taking the annotations as the complete truth might not work [46]. Many researchers are being careful to select reliable positive examples.

The accuracy of a prediction model may rely on the reliable representation of positive and negative examples. Thus it is important to make a wise positive and negative example selection for supervised learning. The problem at hand is commonly known as the positive unlabeled (PU) problem, where the known labels are only the positive class ones. This leads the entire problem towards somewhat a semi supervised learning problem [62]. In literature, most of the gene/protein function prediction methods ignore this fact and choose all examples that are unannotated to the class of interest as negative examples. Much attention has not been made to deal with this important decision making step, prior to model learning. Some researchers have proposed several schemes and heuristics for selecting negative examples. Most of them have been influenced by text classification literature. NoGO database by Youngs et al. [62] is one such attempt to come up with a reliable negative example set generation for protein function. This database is said to contain high quality negative examples for GO terms of humans, mouse, worm, yeast, rice and arabidopsis. However the server is not currently available online.

3.3.1 Negative Example Selection Methods

The common positive example selection is to select the direct set of annotations under the class of interest. All non positive examples can be taken as negative

examples (e.g. [55]). Obozinski et al. [64] selects protein instances annotated with the target GO term or its descendant, as positive examples. The remaining will be the negative examples. Alaydie et al. [53] selects positive examples for an internal node, by taking the superset of the positive example sets union of all of its descendant classes. Negative examples can also be the examples that are not being positive to the class of interest, but are being positive to the parent class. They are considered as the most informative ones for training [53] .

Eisner et al. [46] introduces the notion of exclusive and inclusive classifiers, with respect to the way of selecting positive and negative examples. An exclusive classifier takes all targeted class examples as positives and the rest as negatives, whereas in less exclusive method, the descendant nodes are removed out of the negative examples. Inclusive classifier considers target class examples and their descendants to be the positive examples. Inclusion of ancestor terms as positive examples might be misleading, since there is no guarantee that a child node will be positive when the parent node is positive. Exclusive classifiers are more likely to have TPR violations. When being more inclusive, the number of positive examples is increased and the noise is removed from the train set. Thus, it may lead to a better negative example selection. [46]

Following are some other methods for negative example selection, as described in [62].

ALBNeg with ALBias Youngs et al. [63] selects high confidence negative examples by using prior functional biases. A parameterizable Bayesian technique is used for prior functional biases computation of a gene (i.e. for using available data to form prior beliefs about biological functions of a gene). An empirical conditional probability of seeing an annotation c , given the anno-


tation presence of m , is computed by Equation 3.8,

$$\hat{p}(c|m) = \frac{n_{mc}^+}{n_m^+} \quad (3.8)$$

where n_{mc}^+ = the number of proteins with both m and c annotations; and n_m^+ = the number of proteins with m annotation. For protein i , let D_i be its set of GO term annotations. For a given function c , the conditional prior probability of gene i having the function c is approximated by $prior_i$ score, as calculated from Equation 3.9.

$$prior_i = \frac{1}{|D_i|} \sum_{m \in D_i} \hat{p}(c|m) \quad (3.9)$$

To avoid redundant information from this score, all GO terms that have a child in D_i are removed from D_i . Thus, only the most specific annotations are used for calculating the bias. To avoid the large bias introduced by ancestral relationships, a weighted pseudocount is introduced to the empirical conditional probability calculation, as in Equation 3.10:



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

$$p'(c|m) = \frac{n_{mc}^+}{n_m^+ + \gamma e^{\lambda n_m^+}} \quad (3.10)$$

Parameter values can be selected by tuning with cross validation. For the genes with no previous annotations in GO, the mean of all label biases are computed for genes with GO annotations. Their label bias algorithm is called ALBias. In this technique, the negative examples are chosen based on the label biases calculated for each function. All gene instances with an annotation in the same GO branch as the term being predicted, and which have a priori score of 0 (computed across all the 3 GO branches) for the function of focus, are considered as negative examples. If a specific annotation does not ever appear alongside the function annotation of focus for any other gene, it is a negative example for that function. The method restricts the negative examples to have an annotation in the same branch as

the GO term being predicted. The choice of pseudo counting has no impact over the negative examples, as only the magnitude of the label bias will be affected.

Selection of Negatives through Observed Bias Youngs et al. [62] also describes an extension to the ALBNeg algorithm. Given a function annotation a , each gene instance is scored with the average of the conditional probabilities $\hat{P}(a|f_1), \hat{P}(a|f_2), \hat{P}(a|f_3), \dots, \hat{P}(a|f_n)$, where $\hat{P}(a|f_i)$ is the conditional probability of a gene being already annotated with a function f_i given a . $\vec{\alpha}_a$ is the vector having such score for all genes, with respect to the function a . It can be easily computed through Equation 3.11,

$$\vec{\alpha}_a = W^{-1}AP \quad (3.11)$$

where W = the diagonal matrix, with W_{ii} being the total number of annotations for protein i ; A = annotation matrix of the dataset (rows representing genes and columns representing GO terms); and P = the conditional probability matrix.



University of Moratuwa, Sri Lanka
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The scores in the vector are then ranked according to their value. The lower scores represent genes with a low probability of being positive examples, which in turn makes them to be more probable as negative examples.

Negative Examples from Topic Likelihood: In this method, a protein is viewed as a document, while GO annotations are considered as the words in the document. Then, Latent Dirichlet Allocation (LDA) is applied in order to obtain the topic distribution for each protein. In Natural Language Processing (NLP), LDA is a method for discovering topics of a set of sentences. It takes documents as a mixture of topics, where each topic has its own word distribution.

In the context of protein function prediction, LDA is performed on the

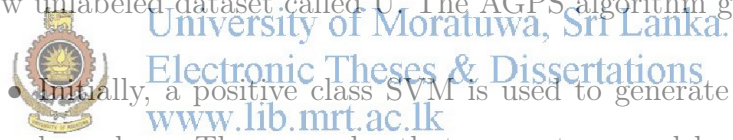
protein data set (i.e. analogous to the document corpus), to identify the parameters of Dirichlet topic distribution. Then the posterior topic distribution is inferred from each protein, given its set of GO annotations. The number of topics for each protein is taken as the total number of annotated direct descendants of the root ontology terms. This is done to increase the quality of latent topics, as well as to preserve the coverage of all GO categories. However, the correspondence between topics and GO terms are not always 1 to 1. As some topic/GO term combinations may relate to each other, it could be difficult for the LDA model to infer the exact probability of a protein being associated with a certain GO term. One solution is to represent the positive class by the average of Dirichlet posterior vectors resulted for all the positive protein examples of the function of interest. Next, a distributional-overlap score is computed for each unlabeled protein. This score represents the similarity of the unlabeled proteins' topic distribution and the positive class average topic distribution. It is a symmetric simplification of the Kullback-Leibler divergence metric and the value is in $[0, 1]$. A low score indicates the most probable negative examples, as they are least likely to share topics with the positive class protein instances. Unlabeled protein instances are ranked by the distributional-overlap score and the negative examples can be selected accordingly.

Rocchio algorithm This again comes under NLP and Information Retrieval context. However, it can be adapted to the protein function prediction domain by treating each protein as a document, and the set of GO terms as the lexicography. An annotation of a protein is a word. The term frequency - inverse document frequency (tf-idf) vectors are produced for each protein instance. A representative vector is then formed for the positive examples, as well as the unlabeled examples. Next, the cosine similarity scores between the tf-idf vector of each unlabeled protein and the representative vectors are computed. The algorithm can then take the proteins that have a higher similarity score with unlabeled example representative vector, compared to

that with the positive example representative vector.

1-DNF This also comes under the context of text classification in which, the enriched words among the positive class documents are identified. All unlabeled documents without those enriched words, are taken as negative documents. Similarly, enriched words can be defined as the GO terms which appear more frequently in positive examples than in unlabeled examples. Negative examples are all the proteins that do not belong to positive class, nor containing any term from the enriched word set.

Annotating Genes with Positive Samples AGPS [61] defines negative examples, solely using positive examples. The idea is to obtain a subset of negative examples from unlabeled data, which can best recover the hidden positive examples in unlabeled data. The approach uses SVMs for the task. Firstly, the positive example set is divided into positive training set P_1 and validation set P_2 . Then P_2 is combined with unlabeled data K_u to form a new unlabeled dataset called U . The AGPS algorithm goes as follow.

- 
- Initially, a positive class SVM is used to generate an initial decision boundary. The examples that are not covered by this boundary are taken as initial negative examples.
 - Then, a two class SVM is trained over the initial negative example set and the positive example set. It is used upon the unlabeled data to retrieve a more refined and an expanded negative example set.
 - This step is redone by using the positive train set and the negative example set obtained from the previous iteration, to retrieve a new negative example set. The procedure continues until it reaches a stopping criteria.
 - The SVM classifier and the generated negative set at each iteration are recorded.
 - Finally, the best classifier; and the negative example set, which recovers the largest number of positive examples from the unlabeled dataset;

are chosen.

However, an error in a previous step can affect the current step, as the procedure is sequential. Cross-validation method can be used to arrive at a more accurate set of negative examples.

Sibling Heuristic In this method, a protein is negative, if it is annotated only with the parent of the function of interest. This can also include the proteins that are annotated with the sibling categories. This heuristic produces different number of negative examples for different GO terms.

3.4 Class Imbalance

In a binary classification problem, the class of interest is usually the positive class, whereas the other class is referred to as the negative class. When the class proportions do not match, it results in a class imbalance, which is a challenge to traditional classifiers [65]. Usually the positive class is the minority class, implying a rare case. Such scenario can result in a misleading figure of accuracy. A classifier may correctly classify all the negative examples due to learning all the relevant classification rules from the majority class train instances, but may give false negatives for almost all truly positive majority class instances. Yet it will produce a false higher accuracy figure. Hence, overcoming class imbalance and providing the learning model with the opportunity to learn positive and negative instances in a well balanced manner is very important. Training over balanced data improves performance [61]. Also when it comes to model evaluation techniques such as cross-validation, the fold balance is the case of having the same positive-example:negative-example ratio at each fold. This can ensure that, each classifier trained upon each fold behaves as closely as possible to the final classifier, which is being constructed by training all the folds [46].

Various approaches have been introduced in literature, to handle and reduce the class imbalance effect. Some of them are as follow.

- Biasing a classifier towards learning the minority class more accurately
- Preprocessing data to convert them into balanced data using a technique such as,
 - random undersampling (random elimination of majority class instances)
 - random oversampling (random addition of more copy instances of the minority class by sampling with replacement) or creating novel synthetic instances to represent the minority class (i.e. SMOTE)
 - hybrid approach of jointly reducing the majority class instances, while increasing the minority class instances
 - random subsampling from majority class instances together with minority class instances
- Cost sensitive learning method by assigning a different cost to each class (i.e. assigning a higher penalty for a mistake made on minority class, while not modifying the data distribution [66])
- Ensemble learning [65]
 - Partitioning majority class train data into disjoint segments and constructing a model for each positive class segment, joined with the entire set of minority class instances
 - Random Balance ensemble: randomly varies the imbalance ratio of train data for each base model using two techniques: SMOTE and random under-sampling without replacement
 - RB-Boost: combines Random Balance with AdaBoost.M2 (random class proportion enforcement plus the instance reweighting). It can also be combined with bagging (BaggingRB).
 - SMOTEBagging (combines bagging with SMOTE), SMOTEBoost (combines boosting with SMOTE), RUSBoost (modification of Adaboost.M2 where random undersampling is applied at each iteration)



Unlike algorithmic and cost sensitive methods, data level and ensemble approaches can be applied independent of the base classifier and thus, they are more versatile [67]. It should also be noted that, instance synthesis for minority class augmentation could result in adulterating the train data, while elimination might lead to discarding important and potentially useful data.

3.4.1 Class Imbalance and Feature Selection

Often the class imbalance can adversely affect feature selection, as applying feature selection methods might result in a set of features that favours a single class over another. Yang et al. [66] presents an ensemble solution for wrapper feature selection, in the presence of a highly imbalanced class distribution. A wrapper algorithm generally consists of three main components: a search algorithm; a fitness function; and an inductive algorithm [66]. The key idea in their proposed approach is to sample several balanced datasets from original train data and to evaluate feature subsets through an ensemble of base models, each trained over a balanced dataset. For sampling a balanced dataset, a hybrid approach is followed from which, a simultaneous increase and decrease is done to achieve a balanced distribution (i.e. the minority class is increased using SMOTE and the majority class is decreased using random undersampling). At each feature set evaluation stage, an ensemble is trained over the balanced train data. The base classifier predictions are then normalized and combined. The fitness function for the search algorithm procedure is taken to be the area under the ROC curve. The method has been tested upon five highly dimensional and imbalanced datasets for greedy forward selection, as well as for the genetic algorithm. The results have demonstrated that the ensemble with greedy forward selection is more robust in the presence of high dimensionality and class imbalance. Improvements with genetic algorithm are mostly moderate, being evident of the fact that it is less sensitive to the ensemble component. According to the overall results, the ensemble method has been significantly better than the feature selection wrapper with single inductive algorithm upon imbalanced data.

Chapter 4

METHODOLOGY

The primary focus of this study was to assess the effectiveness of a weighted heterogeneous data ensemble, for the purpose of classifying *Saccharomyces cerevisiae* proteins under ‘mitochondrion organization’. All preprocessing, model construction and experimental work were carried out using R programming environment (version 3.2.2).

The goal was to employ as many important functional aspects as possible. Thus, six data types: sequence data; protein domains; peptide chain properties; gene expression; secondary structure; and interactions, were used along with existing annotations to construct a supervised learning model. This chapter describes how each type of dataset was retrieved and preprocessed, followed by methods used for building the prediction model that utilizes those heterogeneous data types.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

4.1 Data Retrieval and Preprocessing

An important consideration was given to select a reliable and an appropriate set of data to begin the model construction with. Reliable data selection and preprocessing is crucial in any knowledge discovery process. Unreliable sources or data instances should be avoided, as usage of such error prone data could lead to error propagation, resulting in even more erroneous results. Moreover, biological datasets produced by high-throughput experiments may suffer from high error rates and systematic variations. Thus they require data type specific, standard preprocessing and quality control procedures to be performed prior to feeding them into a learning model [16, 17, 38]. Following sections describe how data were retrieved and preprocessed for this study. Preprocessing was also guided by the produced data visualizations available in Appendix A.

4.1.1 Protein Annotation Data

A reliable set of 32 current *S. cerevisiae* protein annotations under GO:0007005 (i.e. positive examples) were obtained from the Gene Ontology Consortium [68]. They include only the manually curated annotations, excluding the electronically inferred annotations (IEA). In addition, annotations were also obtained from a benchmark gold dataset published by Huttenhower et al. (2009) [69]. The original gold set includes 135 annotations guided by literature curation and 100 experimentally validated proteins which participate in ‘mitochondrion organization’. Moreover, it provides a confident list of 4500 proteins which have been verified as not being engaged in the process (i.e. negative examples). The following steps reduced the amount of positive examples and negative examples up to 239 and 3880, respectively.

- Seven redundant and un-reviewed negative examples were discarded by manual inspection
- Five annotations found to be in both positives and negatives, were removed
- Negative proteins without protein domain information were discarded for later experimental purposes

The final set is a reliable set of annotations, as they include only the manually curated annotations, excluding the electronically inferred annotations.

4.1.2 Sequence Data

Amino acid sequence data of proteins were obtained from the Saccharomyces Genome Database (SGD). They were available in FASTA format, as shown through an example in Figure 4.1.1. R Package *seqinr* [70] allows sequence extraction from a FASTA file. In addition, domain specific sequences were extracted by using start and end positions specified in domain data which will be described in the next section.

4.1.3 Domain Data

Domain data were retrieved from the InterPro database [71] which has InterPro domain annotations for all UniProtKB proteins. Initially the *S. cerevisiae* ORF specific domain data were downloaded from SGD. However only 618 proteins seemed to have InterPro domains. Thus the complete InterPro domain data file with 299,626,886 lines was downloaded to extract only the yeast protein domain data. The file is a tab delimited file containing the domain annotations for each protein, in the form of their InterPro IDs and names, along with the start and end loci within the corresponding amino acid sequence. This large file of a size around 3.9GB was accessed by splitting it into portions of 1,000,000 lines. InterPro uses UniProt IDs and thus, 'Retrieve/ID mapping' online tool from UniProt was used to manually obtain the InterPro - SGD ORF mappings for the focused list of protein SGD IDs. Further, for 23 proteins (i.e. with UniProt IDs P16547, P16965, P32858, P36038, P38305, P38325, P40207, P40451, P40491, P47157, P50945, P53140, P87275, Q01926, Q02783, Q02888, Q03429, Q03798, Q04964, Q06820, Q08223, Q3E731, Q99299) domain data had to be obtained from Uniprot that presents domains based on publications, as InterPro did not have any domain records for them.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mru.ac.lk

```
>YAL005C SSA1 SGDID:S000000004, Chr I from 141431-139503, Genome Release 64-2-1, reverse complement, Verified ORF, "ATPase involved in protein folding and NLS-directed nuclear transport; member of HSP70 family; forms chaperone complex with Ydj1p; localized to nucleus, cytoplasm, and cell wall; 98% identical with paralog Ssa2p, but subtle differences between the two proteins provide functional specificity with respect to propagation of yeast [URE3] prions and vacuolar-mediated degradations of gluconeogenesis enzymes; general targeting factor of Hsp104p to prion fibrils"  
MSKAVGIDLGTTYSCVAHFANDRVDIANDQGNRTTPSFVAFTDTERLIGDAAKNQAAMN  
PSNTVFDARLIGRNFNDPEVQADMKHFPPKLDVDGKPKIQVEFKGETKNFTPEQISSM  
VLGKMKETAESYLGAKVNDVAVTPAYFNDSQRQATKDAGTIAGLNVLRIINEPTAAAIY  
GLDKKGGKEEHLIFDLGGGTFDVSLLSIEDGIFEVKATAGDTHLGGEDFDNRLNVNHFIEF  
KRKKNKDLSTNQRALRRLRTACERAKRTLSSSAQTSVEIDSLFEGIDFYTSITRARFEELC  
ADLFRSTLDPVEKVLDAKLDKSQVDEIVLGGSTRIPKQKLVTDYFNGKEPNRSINPDE  
AVAYGAAVQAAILTGDESSKTQDLLLLDVAPLSLGIETAGGVMTKLIPRNSTIPTKSEIFST  
YADNQPGLIQVFEGERAKTNDNLLGKFEKSGIPAPRGVPIEVTFDVSNGILNVSVA  
EKGTGKSNKITITNDKGRLSKEDIKMAEAEKFEDEKESQRIASKNQLESIAYSLKNTI  
SEAGDKLEQADKDTVTKKAETISWLDSENTTASKEEFDDKLELQDIANPIMSKLYQAGG  
APGGAAGGAPGGFPGGAPPAPEAEAGPTVEEVD*
```

Figure 4.1.1: Example FASTA sequence of a protein

4.1.4 Properties Data

A protein properties dataset with important peptide chain features, was obtained from the SGD. They include molecular weight, isoelectric point (pH value at which a molecule carries no net charge), protein length, N terminal sequence, C terminal sequence, GRAVY score (hydropathicity of the protein), Aromaticity score (frequency of aromatic amino acids; Phe, Tyr, Trp), codon adaptation index, codon bias, FOP score (frequency of optimal codons), Composition of Carbon, Hydrogen, Oxygen, Nitrogen, Sulphur elements, Instability index, Aliphatic index, 'assuming all Cys residues appear as half Cystines' and 'assuming no Cys residues appear as half Cystines' for all proteins. Molecular weight has a highly skewed distribution which ranges from min 1978 to max 559,100 with a mean of 50,860. Thus, \log_{10} transformation was performed.

4.1.5 Gene Expression data

Four types of microarray gene expression datasets [72, 73, 74, 75] often used in literature, were downloaded from their corresponding sites.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Mnaimneh et al. (2004) [72] Expression data: This study is the exploration of essential gene functions via titratable Promoter Alleles. Gene expression data are from cDNA microarrays which measure the expression of all *S. cerevisiae* genes under a set of 215 titration experiments. The values are log expression ratios of mutant vs wildtype. Authors have examined 215 of the TetO7-promoter strains by microarray expression profiling (where 215 mutant strains represent 215 different TET mutants).

Chu et al. (1998) [73] Expression data: This study is the transcription program of sporulation in budding yeast. Authors have used DNA microarrays containing 97% of the known or predicted genes of *S. cerevisiae*, to explore the temporal program of gene expression during meiosis and spore formation. Changes in the mRNA transcript concentrations from each gene, have been measured at 7 successive intervals (i.e. t0, t0.5, t2, t5, t7, t9,

t11.5) after transfer of wild type (SK1 strain) diploid yeast cells to a Nitrogen deficient medium which induces sporulation. In their experiment, the sporulation channel is R and the wild type channel is G. For each time stamp: red; red background; green; and green background intensity values are provided. They have also examined consequences of expressing Ndt80 (i.e. YHR124 - Meiosis specific transcription factor), ectopically in vegetative cells; and of eliminating Ndt80 during sporulation, during 3 stages: gal.ndt80; Ndt80.delete.early; and Ndt80.delete.mid.

Gasch et al. (2001) [74] Expression data: This study is on genomic expression responses to DNA-damaging agents, and the regulatory role of the Yeast ATR Homolog Mec1p. The authors have used DNA microarrays to observe yeast gene expression in the presence of 2 different DNA damaging agents: methylating agent methyl methanesulfonate; and ionised radiation. Purpose is to characterize the role of MEC1 pathway in modulating cellular response to DNA damage. Their analysis includes wild type and mutant cells (defective in MEC1 signaling) under normal growth conditions vs in response to the 2 DNA damaging agents, identifying specific features of gene expression responses that depend on MEC1. The expression dataset includes time course data. Used mutants are *mec1*, *dun1*, and *crt1*.

Spellman et al. (1998) [75] Expression data: This study is on how transcript levels of each gene vary within the yeast cell cycle. According to the authors, the cell cycle regulated genes belong to 5 classes during yeast cell cycle: M/G1, G1, S, G2, and M. There are 4 experiments with time course data for each. Three independent methods: alpha factor arrest; elutriation; and *cdc15* temp-sensitive mutant arrest, have been used with *cdc28*. Average fluorescence intensities have been taken, background corrected and finally the ratios have been included in the dataset.

These four datasets are also referred by the names: Expressions 1; Expressions 2; Expressions 3; and Expressions 4; respectively, for results reporting purposes

in Chapter 5. As explained in Chapter 2, microarray gene expression data has to undergo a quality control and normalization procedure.

Preprocessing Methods

The raw expression data should undergo specific quality control and normalization with following methods, as clearly described in [17, 18, 38].

- **Background correction**

The simplest form of background correction is to subtract the background intensities from the foreground intensities. One way to do this is by subtracting the background pixel median value from the spot pixel median value. Bioconductor Package *Limma* [76] provides a more sophisticated correction through an adaptive background correction method known as ‘normexp’. It adjusts the foreground adaptively for the background intensities and results in positive adjusted intensities. This is done by fitting a convolution of normal and exponential distributions to the foreground intensities using the background intensities as a covariate. The expected signal given the observed foreground is taken as the corrected intensity. Resultant is a smooth monotonic transformation of the background subtracted intensities such that, all the corrected intensities are positive. An offset is used to damp the variation of log ratios for very low intensity spots towards zero. [76]

- **Expression ratio transformation**

Expression levels across two conditions (i.e. experimental condition (R) and control/reference condition (G)) cannot be compared in their absolute units, as the starting amounts of mRNA might be different. This becomes a major concern, especially for dual channel microarray experiments. A relative expression measure should be computed to intuitively represent expression differences. Thus, gene expression level for the experimental condition is normalized by that of the reference condition, as in Equation 4.1,

$$T_k = \frac{R_k}{G_k} \quad (4.1)$$

where R_k = spot intensity of red channel; and G_k = spot intensity of green channel.

However, the expression ratio alone cannot give much information about differential expression. For instance, a gene up-regulated by a factor of 4, has a ratio of 4; but if it is down-regulated, the ratio is 0.25. Thus, the ratios have to be transformed for having a more intuitive and consistent differential expression measure. One way is to perform inverse transformation, where it simply changes the ratio to fold-change. However, its mapping space is discontinuous. The best and the common method is to perform logarithmic transformation with base 2, where the magnitude directly gives the fold-change, and the sign represents whether it is an up-regulation (+) or a down-regulation (-). Log transformation also makes the expression value distribution to be closer to a normal distribution, while the multiplicative noise becomes an additive noise. However, this transformation is not in favour for analyzing differential regulation of expression.

- **Normalization**

Within-array and between-array normalization are required to remove systematic biases in expression data. Global normalization assumes the red and green intensities to be related by a constant factor, and shifts the center of the log ratios to 0 [38]. Equation 4.2 presents the normalization step. Common choice for c is the mean or median of log ratios.

$$\log_2\left(\frac{R}{G}\right) \leftarrow \left[\log_2\left(\frac{R}{G}\right) - c = \log_2\left(\frac{R}{kG}\right) \right] \quad (4.2)$$

For this study, median centering was preferred, as it is a more reliable measure than the mean in the presence of outliers. The goal is to enforce all sample expression profiles to have the same median. The exact median

value is not important, but the conventional choice is to have zero median, as any value above-zero and below-zero reflects an up-regulation and down-regulation, respectively.

A stochastic process may cause variance of the measured log ratios to differ from one region of array to another, or between arrays [17]. Thus variance regularization (scale normalization) has to be performed to adjust the measures for gaining the same variance. Yang et al. [38] recommends scale normalization for expression ratio values after median centering.

As [17] describes, firstly a scaling factor a_k should be computed for each subgrid k, by dividing the variance σ_n^2 of the particular subgrid n by the geometric mean of the variances for all subgrids. Afterwards, all log expression ratios within each subgrid k is divided by the corresponding scaling factor a_k , as in Equation 4.3.



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
 www.lib.mrt.ac.lk

(4.3)

[41] describes the following procedure. Consider the expression matrix X with i genes and j samples. Let x_{ij} denote the expression of i^{th} gene at the j^{th} sample.

1. Determine sample medians $m_j = median(x_{1j}, x_{2j}, \dots, x_{ij})$

2. Find median absolute deviations

$$MAD_j = median(|x_{1j} - m_j|, |x_{2j} - m_j|, \dots, |x_{ij} - m_j|)$$

3. In order to scale normalize each sample j, multiply all of its values by C/MAD_j where $C = \sqrt[n]{a_1 * a_2 * \dots * a_n}$ (geometric mean of median absolute deviations)

In the case of this study, individual sample expression profiles were taken as subgrids and the variances were adjusted across the samples by performing

scale normalization.

Another important normalization known as loess normalization should be performed to remove dye effects from dual channel expression values. Yang et al. [38] recommends this prior to median centering and scale normalization. The non linear bias due to systematic artifacts such as dye effect and print tip effect, can be identified through a log fold-change (M) vs average expression (A) plot (i.e. MA plot). M and A are defined in Equation 4.4.

$$M = \log_2(R) - \log_2(G) \quad \text{and} \quad A = \frac{\log_2(R) + \log_2(G)}{2} \quad (4.4)$$

In a realistic MA plot, genes with similar expression must appear around $M = 0$ line, and a cloud should symmetrically distribute all data points. This is expected under the biological assumption; the majority of genes are not differentially expressed, and the proportion of the up-regulated and down-regulated genes are almost the same. Any deviation from that nature is evidential of a systematic artifact. Loess method can identify such systematic deviations and correct them through a local weighted linear regression as a function of \log_{10} expressions, and subtracting the calculated best-fit average \log_2 ratio from the experimentally observed ratio for each data point [17]. If $x_i = \log_{10}(R_i * G_i)$ and $y_i = \log_2(\frac{R_i}{G_i})$, the method initially estimates $y(x_k)$: the dependence of the \log_2 ratio on the \log_{10} intensity. Then the aforementioned function is used to correct the measured \log_2 ratio values, point by point as in Equation 4.5.

$$\log_2(\hat{T}_i) = \log_2(T_i) - y(x_i) = \log_2(T_i) - \log_2(2^{y(x_i)}) \quad (4.5)$$

As a more specialized method, cyclic loess normalization [77] method normalizes two arrays at a time, by applying a correction factor obtained from the loess curve fit through the MA plot of the two arrays. This can be applied for between-array normalization. Moreover, Robust spline normal-

ization method is said to be more robust than lowess at lower and upper intensities. Bioconductor Package *aroma.light* [78] has an implementation for Robust spline, which can be applied as a curve fit normalization between R and G channels for within-array normalization.

Lastly, another technique called quantile normalization can be applied as a calibration process, which improves the expression array comparability [41]. It enforces each sample profile to have the same quantiles. The procedure is to find the smallest \log_2 intensity on each channel at first; average those values; replace each value of those earlier smallest \log_2 values with the computed average; and repeat the steps again for the second smallest, third smallest and so on.

Preprocessing Steps

Preprocessing steps were guided mainly by the side-by-side box plots and MA plots. The plots were obtained for visualizing the non-biological and systematic artifacts present within the expression datasets. To remove those non-biological variance and systematic artifacts, methods explained in [17, 18, 38] should be applied over these expression datasets prior to model construction. Thus, we performed median centering, followed by scale normalization and quantile normalization, as a calibration process that improves the comparability among microarray experiments. Missing values contained in both Expressions 3[74] and Expressions 4[75] data were imputed using kNN imputation from R Package *Impute* [79]. Here, rows with more than 50% missing entries were imputed using mean imputation. For Expressions 2 [73] dataset, some initial quality control was done as follow. YDR273W negative ORF had an additional expression profile, where the euclidean distance between the two profiles is ~ 0.2198 . They were averaged into a single expression profile. Moreover, the foreground intensities were adjusted adaptively for the background intensities using Bioconductor Package *Limma* [76]. This method adjusts the foreground adaptively for the back-

ground intensities and results in strictly positive adjusted intensities. Further, within-array normalization was done using Robust spline method in Bioconductor Package *aroma.light* [78]. Between array normalization was done using cyclic loess normalization in *Limma* [76] for two arrays at a time. All expression values are \log_2 fold-changes.

4.1.6 Secondary Structure Data

The secondary structure for all positive proteins and negative proteins were predicted using NetSurfP-1.1 [82]: an online server developed and hosted at the Technical University of Denmark. Given a set of amino acid sequences in FASTA format, the server process them and outputs the probabilities of a residue belonging to one of the 3 main secondary structural elements: alpha helix (H); beta strand (E); and Coil(C). These data can be used to obtain a 3 category secondary structure assignment, by annotating each residue with H,E and C, creating a secondary structure sequence. The results also carry more predictions over the surface accessibility (relative surface accessibility with Z-score, and absolute surface accessibility). Residues are getting classified as ‘buried’ or ‘exposed’ with a 25% exposure threshold. NetSurfP uses: a neural network approach introduced previously in [83] for secondary structure prediction; an ensemble of neural networks over Position Specific Scoring matrices; and the predicted secondary structure to predict the relative exposures.

4.1.7 Interaction Data

Protein interaction data were downloaded from BIOGRID through SGD. There are two interaction types: genetic interactions and physical interactions. The data have been generated from 27 different bait and prey experimental approaches. The complete dataset has 336,198 interaction records. 16.65% interactions are manually curated, and 83.35% have been identified from high throughput interactions. 66.38% are genetic interactions, while the rest (33.62%) being physical interactions. Some genetic interactions (24.3%) are tagged with 3 phenotypes:

inviable; decreased vegetative growth; and normal vegetative growth. Physical interactions do not have any phenotypes. In this study, phenotype was not taken into account. There are 0.60266% self interactions (i.e. bait = prey), which account for proteins whose two or more copies can interact with each other. Some self interactions are in both physical and genetic level (e.g. YAR019C). 69,213 duplicates (in terms of bait, hit, genetic/physical interaction) were identified and removed.

Complementary interaction record pairs exist in which, both (bait A, hit B) and (bait B, hit A) pairs refer to the same interaction. A reasonable thing is to retain only one of them and remove the complementary as a duplicate. However, if the complementary interactions are from 2 types of interactions, they should be kept. The complementary duplication removal was performed separately for the genetic and physical interaction sets. After removing the complementary interactions, we have 79,575 (34.42%) physical interactions and 151,599 (65.58%) genetic interactions. For physical interactions, the average number of neighbours for a vertex is 4.9482 and the maximum number of neighbours is 128. Network object creation was done using package *igraph* [84].

4.2 Protein Instance Representation Methods


For effective supervised learning, each individual positive or negative example should be properly represented. Selection of the most suitable and accurate protein instance representation is important and challenging. This depends on the data types at hand. Primary structure based and secondary structure based representation methods usually focus on discrete modeling of the protein sequence. Different sequence features can be taken into consideration.

4.2.1 Pseudo Amino Acid Composition (PAAC)

Amino Acid Composition (AAC) is the simplest discrete model for representing an amino acid sequence. Let P be the protein with the amino acid sequence

$R_1R_2R_3\dots R_L$ of length L , where R_i represents residue i . Then the AAC vector is $[f_1, f_2, f_3, \dots, f_{20}]$, where f_{AA} is the normalized occurrence frequency of each of the 20 types of amino acids within the sequence. This was commonly used to model sequences in earlier approaches. However, it does not contain any sequence order information which may be evidential of a certain biological importance. Pseudo Amino Acid Composition (PAAC) numerical representation scheme was introduced by Chou et al. [80], to represent an amino acid sequence by considering different amino acid properties. It is an attempt to retain the sequence order information, unlike AAC which is solely based on amino acid frequencies. The final output is a numeric vector with each value representing a quantity based on combining the aforementioned amino acid properties.

Suppose there are n amino acid property types. A generic correlation function $\Theta(R_i, R_j)$ can be defined for two amino acid residues: R_i and R_j , where $H_k(R_i)$ is the k^{th} property of R_i , as in Equation 4.6.



$$\Theta(R_i, R_j) = \frac{1}{n} \sum_{k=1}^n [H_k(R_i) - H_k(R_j)]^2 \quad (4.6)$$

Then, a set of descriptors called sequence order-correlated factors are defined as in Equation 4.7,

$$\theta_\lambda = \frac{1}{n - \lambda} \sum_{i=1}^{n-\lambda} \Theta(R_i, R_{i+\lambda}) \quad \text{for } \lambda = 1, 2, \dots, \lambda < L \quad (4.7)$$

where λ is a parameter to define the maximum distance between a pair of residues [85]; and L is the length of the sequence.

The correlation pairs are determined according to the λ value. When $\lambda = 1$, all the most contiguous residue pair correlations are taken along the sequence. When $\lambda = 2$, it will be the second most contiguous residue pairs. As λ increases, the pattern continues with third most, fourth most and so on. The PAAC numeric vector contains these $20 + \lambda$ descriptors (X_c for $c = 1, 2, \dots, 20, \dots, \lambda$), as given in

Equation 4.8. The first 20 descriptors ($X_1 - X_{20}$) are based on the normalized occurrence frequencies (f_c) of each amino acid type. The rest (i.e. λ descriptors) are the sequence order-correlated factors. A weight factor w is applied to the sequence order correlation effect.

$$X_c = \frac{f_c}{\sum_{r=1}^{20} f_r + w \sum_{j=1}^{\lambda} \theta_j} \quad (1 < c < 20) \quad \text{and} \quad (4.8)$$

$$X_c = \frac{w\theta_{c-20}}{\sum_{r=1}^{20} f_r + w \sum_{j=1}^{\lambda} \theta_j} \quad (21 < c < 20 + \lambda)$$

R Package *protr* [81] provides an implementation to compute the PAAC vector for a peptide chain by considering 3 key amino acid residue properties: hydrophobicity; hydrophilicity; and side chain mass. Let $H_1^O(i)$, $H_2^O(i)$ and $M^O(i)$ for $i = 1$ from $i = 20$ be the original hydrophobicity, hydrophilicity and side chain mass values, respectively, corresponding to each amino acid type out of the 20 types. Firstly, they undergo normalization/standardization as per the Equation 4.9, 4.10 and 4.11.



$$H_1(i) = \frac{H_1^O(i) - \frac{1}{20} \sum_{i=1}^{20} H_1^O(i)}{\sqrt{\frac{\sum_{i=1}^{20} [H_1^O(i) - \frac{1}{20} \sum_{i=1}^{20} H_1^O(i)]^2}{20}}} \quad (4.9)$$

$$H_2(i) = \frac{H_2^O(i) - \frac{1}{20} \sum_{i=1}^{20} H_2^O(i)}{\sqrt{\frac{\sum_{i=1}^{20} [H_2^O(i) - \frac{1}{20} \sum_{i=1}^{20} H_2^O(i)]^2}{20}}} \quad (4.10)$$

$$M_1(i) = \frac{M_1^O(i) - \frac{1}{20} \sum_{i=1}^{20} M_1^O(i)}{\sqrt{\frac{\sum_{i=1}^{20} [M_1^O(i) - \frac{1}{20} \sum_{i=1}^{20} M_1^O(i)]^2}{20}}} \quad (4.11)$$

The correlation function is defined by the Equation 4.12.

$$\Theta(R_i, R_j) = \frac{1}{3} ([H_1(R_i) - H_1(R_j)]^2 [H_2(R_i) - H_2(R_j)]^2 [M(R_i) - M(R_j)]^2) \quad (4.12)$$

4.2.2 Quasi-Sequence-Order Descriptor (QSOD)

This is another representation proposed by Chou [86], which can be derived using a physicochemical distance matrix that has been defined over the 20 amino acid types. Pairwise distances are calculated for each pair of amino acid residues that

are at most separated by a maximum lag (*maxlag*). The separation is reflected by rank *d* which ranges from 1 to *maxlag*. The sequence-order coupling number is defined as in Equation 4.13, and is calculated for each *d*.

$$\tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2 \quad d = 1, 2, \dots, \text{maxlag} \quad (4.13)$$

The QSOD numeric vector contains $20 + \text{maxlag}$ values. Just as in PAAC, the first 20 values correspond to the 20 amino acid types, and are computed based on their normalized occurrence frequencies (f_r). The rest refer to the quantities computed using sequence-order coupling numbers for each *d*. Equation 4.14 and 4.15 define the two QSOD sub vectors: X_r and X_d .

$$X_r = \frac{f_r}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{\text{maxlag}} \tau_d} \quad r = 1, 2, \dots, 20 \quad (4.14)$$

$$X_d = \frac{w\tau_{d-20}}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{\text{maxlag}} \tau_d} \quad r = 21, 22, \dots, 20 + \text{maxlag} \quad (4.15)$$

R Package *protr* [81] provides implementation of this descriptor by incorporating Schneider-Wrede physicochemical distance matrix, as well as the Grantham (1974) distance matrix.

4.2.3 Conjoint Triad Descriptors

Shen et al. [87] proposed conjoint triad method, which firstly classifies the 20 amino acid types into 7 classes based on their dipoles and side chain volumes. Then, a numeric vector with features called conjoint triads is obtained for a sequence. The goal is to reflect the residues which participate in electrostatic and hydrophobic interactions with other proteins. Figure 4.2.2 gives the classification table, with dipole scale in Debye (Dipole < 1.0 (-), 1.0 < Dipole < 2.0 (+), 2.0 < Dipole < 3.0 (++) , Dipole > 3.0 (+++), Dipole > 3.0 with opposite orientation (+'+'+')); and volume scale in angstrom (Volume < 50 (-), Volume > 50 (+)). It should be noted that the amino acid residue Cys is separated from class 3 due to its ability to form disulfide bonds.

No.	Dipole Scale ¹	Volume Scale ²	Class
1	—	—	Ala, Gly, Val
2	—	+	Ile, Leu, Phe, Pro
3	+	+	Tyr, Met, Thr, Ser
4	++	+	His, Asn, Gln, Tpr
5	+++	+	Arg, Lys
6	+'+'+'	+	Asp, Glu
7	+ ³	+	Cys

Figure 4.2.2: Amino acid residue classification [81]

A triad is a unit of three consecutive amino acid residues. Triads are differentiated as illustrated in Figure 4.2.3, according to its composition in terms of the previously defined 7 classes. Since there are 3 residues, $7 \times 7 \times 7 (= 343)$ different combinations of classes can be found. In this representation, the sequence features are considered to be the triad frequencies. Following gives mathematical notion of the method. Define sequence feature vector $V = (v_i; \text{for all } i = 1 \text{ to } i = 343 \text{ triad type})$; and frequency vector $F = (f_i; \text{for all } i = 1 \text{ to } i = 343 \text{ triad type})$. A protein can be represented by its F

Since f_i could correlate to the sequence length, the F vector should be normalized using Equation 4.16, in order to compare 2 proteins in terms of their F representation. This normalization enforces the final representation vector D to have values in $[0,1]$ interval, where $D = (d_i; \text{for } i = 1 \text{ to } i = 343 \text{ triad type})$.

$$d_i = f_i - \frac{\min(f_1, f_2, \dots, f_n)}{\max(f_1, f_2, \dots, f_n)} \quad (4.16)$$

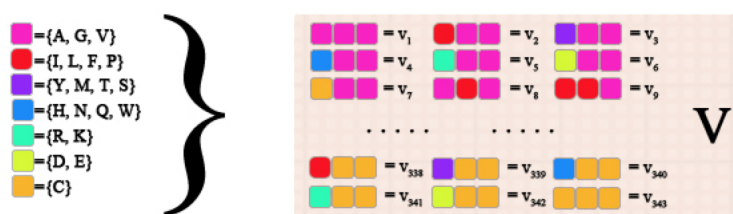


Figure 4.2.3: Conjoint triads [87]

The authors in [87] have used CTD representation to model PPI between two proteins: A and B, by concatenating D_A and D_B . R Package *protr* [81] also provides an implementation to compute the CTD vector.

4.2.4 Secondary Structure based Representation

Another numeric representation for a protein can be constructed with 11 widely applied secondary structure based features, as used in [27]. The vector S contains features computed upon a predicted secondary structure sequence, as defined in Equation 4.17.

$$S = (Con_H, Con_E, \frac{Avg_H}{N}, \frac{Avg_E}{N}, \frac{Max_H}{N}, \frac{Max_E}{N}, \frac{Alt_n}{N}, \frac{P_{NE}}{P_{NE} + AP_{NE}}, \frac{AP_{NE}}{P_{NE} + AP_{NE}}, CMV_H, CMV_E) \quad (4.17)$$

Con_H	=	alpha helix residue occurrence
Con_E	=	beta strand residue occurrence
$\frac{Avg_H}{N}$	=	normalized average length of Alpha helices
$\frac{Avg_E}{N}$	=	normalized average length of Beta strands
$\frac{Max_H}{N}$	=	maximum normalized length of Alpha helices
$\frac{Max_E}{N}$	=	maximum normalized length of Beta strands
$\frac{Alt_n}{N}$	=	alternating frequency of Alpha helices and Beta strands
$\frac{P_{NE}}{P_{NE} + AP_{NE}}$	=	normalized parallel beta sheet count
$\frac{AP_{NE}}{P_{NE} + AP_{NE}}$	=	normalized anti-parallel beta sheet count
CMV_H	=	composition moment vector of alpha helices
CMV_E	=	composition moment vector of beta strands

4.2.5 Latent Dirichlet Allocation (LDA) Topic Representation

For sequence representation, a common approach is to exploit k-mers, as they retain sequence order information [88]. Since a protein sequence is a text string over an alphabet of 20 letters corresponding to the 20 amino acid residue types, the k-mers can be considered as words. Thus, a protein instance can be regarded

as a document under the ‘bag of words’ concept, creating the possibility to apply text document mining approaches.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model of a document corpus, proposed by Blei et al. [89] for text mining. It is a method to identify the most recurrent, but hidden topics shared by a corpus [90]. Following its success in the document mining domain, the model has made its way into Computational Biology very recently. For example, Blei et al. [91] uses LDA for biomedical text mining. Konietzny et al. [92] identifies Biological Process functional modules of protein families from microbial genome annotations, by using LDA topic modeling to model a gene as a document, and the words as its gene family annotations.

As for genomic sequence modeling using topic modeling, some recent studies suggest different ways of word segmentation. For instance, Rosa et al. [90] applies LDA with 8-mers, for predicting the taxonomic class of barcode DNA sequences belonging to 16S housekeeping bacterial and archaeal DNA genes. Upon LDA model building, a one-to-one mapping of a topic to a taxonomic label is performed, by assigning the label among the majority of sequences which have the particular topic as their highest probable. Then the fitted model is used upon a new DNA sequence, by firstly decomposing it into k-mers and then retrieving the topic distribution, as to assign the most probable topic to the sequence. Final output is the taxonomic label corresponding to the particular topic. Compared to DNA/RNA sequences, the k-mer space is much larger for amino acid sequences. Nevertheless, several researchers have used the approach, while looking for ways to reduce the k-mer space. For instance, Yang [88] models protein sequences for predicting type III secreted effectors (T3SEs) using non-overlapping 2-mers and 3-mers. They also pick only the informative k-mers by referring to their term frequencies or the tf-idf value. Pan et al. [93] used 3-mers as well. However Yang et al. [94] refers to the fact that, the typical longest distance between amino acid local interactions is four. It is also important to note that the k-mers should not

be too long, as longer ones tend to be less informative.

LDA topic modeling process

The LDA topic modelling process can be described as follow. An LDA model considers a text document as a random mixture of latent topics, where each topic has its own word distribution. Suppose a document W is a sequence of N words (w_1, w_2, \dots, w_N) ; a corpus D is a collection of M documents (W_1, W_2, \dots, W_M) ; θ_d is the topic proportions vector for document d ; and k is the term distribution for topic k . Blei et al. [89] defines the following generative process for each document w in the corpus D .

Choose $N \sim \text{Poisson}(\xi)$

Choose $\theta \sim \text{Dir}(\alpha)$

For each of the N words w_n ,

(i) Choose a topic $z_n \sim \text{multinomial}(\theta)$

(ii) Choose a word w_n from $\phi(w_n|z_n)$



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The topic proportion vector θ of a document is drawn from Dirichlet distribution given its parameter α , as in Equation 4.18.

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \dots \theta_k^{\alpha_k-1} \quad (4.18)$$

Given the parameters α and β , the joint distribution of θ (a topic mixture), z (a set of N topics) and W (a set of N words), is defined by the Equation 4.19.

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^N p(z_n|\theta) p(w_n|z_n, \beta) \quad (4.19)$$

By integrating over θ and summing over z , the marginal distribution of the document is obtained, as in Equation 4.20.

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta \quad (4.20)$$

Finally the probability of the document corpus can be defined through Equation 4.21. It is the product of the marginal probabilities of single documents.

$$p(D|\alpha, \beta) = \prod_{d=1}^M \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d \quad (4.21)$$

Only w_{dn} is fully observable. Hidden variables are inferred often using Gibbs sampling [88].

In the protein function prediction context, a positive annotation and a negative annotation can be taken as two different topic representations. An LDA model represents each document by a posterior Dirichlet over the topics [88]. Thus it can be used to obtain a probability vector for a protein under the 2 cases: ‘the protein belongs to the function class’ and ‘the protein does not belong to the function class’.

4.2.6 Gene Expression Profile Representation

Gene expression profiles obtained from microarray experiments allow the identification of differentially expressed genes between two or more conditions (i.e. control vs. treatment/ normal vs disease/ phases of a biological process such as cell cycle/ time stamps), or genes whose regulation is evidential of their engagement in a certain biological process [18]. A protein instance can be represented in terms of its gene expression profile, which may carry information regarding differential expression and differential expression regulation. As explained in Chapter 2, a microarray experiment outputs a set of images with spot colors, reflecting the level of gene expression. They are processed to obtain a raw expression data matrix of spot intensity data, where rows represent genes; and columns repre-

sent samples. A gene expression profile is simply a raw expression vector in the expression data matrix, corresponding to the gene.

4.3 Ensemble based Classification

An Ensemble learning model amalgamates several diverse base predictors with expertise in different input regions. That is to attain more accuracy than at least the best performing base predictor. The behind motivation for such an approach is the human nature of drawing conclusions based on several expert opinions. In the context of social sciences, it is evident that if a committee is consisted of individuals with a reasonable competence, the overall judgement of the committee is superior to those of the individuals [95]. Following is proof by Dietterich [96], that an ensemble has better results than each of its single classifiers. Suppose a dichotomic classification problem for which, there are L hypotheses constructed with each of their error being less than 0.5. Then, as long as the errors of the base learners are uncorrelated, a majority voting ensemble shows an error lower than that of single classifiers. The overall ensemble error rate P_{error} is given by the Equation 4.22. It is the area under a binomial distribution, where more than $L/2$ hypotheses are wrong.

$$P_{error} = \sum_{i=\lceil L/2 \rceil}^L \binom{L}{i} p^i (1-p)^{L-i} \quad [95] \quad (4.22)$$

As there is no single learning algorithm which could achieve the best accuracy for all kinds of scenarios [97], it is a wise approach to develop a multiple learning system, which can make each constituted base learner to cover a different aspect of the problem domain. A different hypotheses space can be obtained with each classifier [98], leading to an ensemble of the complementary prediction models that misclassify completely separate parts of the input space [99]. Hence individual errors can be corrected by the other base models. When base models have different data learning focuses, their combination tends to handle data changes in a much more relaxed manner. Thus, unlike a single model which may overfit,

an ensemble tends to reduce the variance of classifiers, which in turn reduces its generalization error [100]. In certain situations, they can also reduce the bias error [101].

To achieve an effective ensemble model with a good generalization capability, the base model diversity should be enforced carefully. Re-sampling methods; different feature sets; different parameter choices; different architectures; and different learning algorithms; can be used for the purpose. Diversity determines how capable an ensemble would be to outperform its best base model. However, it should be balanced with accuracy, in order to gain a well-performance [102]. Fusion of base model predictions is the stage where the final prediction is presented by the ensemble. Various combination strategies (i.e. uniform/weighted voting methods, arithmetic aggregation methods such as weighted mean, Bayesian probabilistic methods, fuzzy methods [95]) have been introduced for the purpose. Simple combination schemes are much effective than sophisticated schemes in certain contexts [55]. In addition, an ensemble model can be enforced to lean on an optimal set of base classifiers, by selecting them using an appropriate optimization algorithm (exhaustive or heuristic). The goal can be to achieve a significantly high accuracy and diversity.

Nowadays, many Computational Biology applications rely on widely used ensemble methods such as bagging, boosting, random forests, meta learning methods (e.g. stacking, arbiter trees and combiner trees), and Error Correcting Output Code [101, 103, 104]. Yang et al. [98] presents an excellent review on the most widely used ensemble learning methods and their future trends in Bioinformatics. However, utilizing an ensemble requires careful design in terms of the construction method (i.e. serial, parallel, hierarchical), diversity enforcement, and the combination strategy. A general theoretical framework for ensemble methods have not yet been developed, since the hidden commonalities among different approaches are still unknown [95].

4.3.1 Heterogeneous Data Ensemble

This study focuses on implementing a heterogeneous data ensemble, comprised of 12 base models. They include 3 affinity-based neighbourhood models and 9 nearest neighbour models.

4.3.2 Affinity-based Neighbourhood models

This type of model takes a proteins' affinity with another protein into account, when predicting the protein function. Affinity between two proteins can indicate an engagement in the same biological process. Thus, this is an application of 'guilt-by-association' concept. A protein is classified under 'mitochondrion organization', only if the majority of its interacting proteins are also engaged in the same functional process. In a probabilistic sense, the posterior probability that a protein engages in the biology process of interest, is simply the proportion of positive proteins in its interaction neighbourhood. Such a model was built for the case of combined physical and genetic interactions, as well as for each case of individual interaction types. At training, only the interactions among proteins in the train dataset were taken into consideration.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mru.ac.lk

In addition, another affinity-based neighbourhood model called the Domains model was also evaluated. This model firstly obtains the list of InterPro domain annotations for all the proteins. At prediction, the neighbourhood will be the other proteins that share protein domains with the protein of focus. The posterior probability is simply the fraction of positive neighbourhood proteins. If there are no positive proteins in the neighbourhood, the function is not predicted. This is because, a perfect neighbourhood cannot be guaranteed due to missing information.

4.3.3 Nearest Neighbour Models

A nearest neighbour (NN) model outputs for a particular protein, the frequency of its positive example neighbours, as the posterior probability of belonging to

the ‘mitochondrion organization’ class. The neighbourhood is defined by a distance function and the number of neighbours (k) to be considered. The distance function for this study was taken to be the Euclidean distance. The Euclidean distance d between two data vectors: $x = (x_1, x_2, \dots, x_n)$ and $y = (y_1, y_2, \dots, y_n)$, is defined by the Equation 4.23.

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (4.23)$$

During experimentation, the optimal number of neighbours was empirically decided. The most closest k proteins according to the euclidean distance are taken to be a proteins’ neighbourhood. Using R Package *kknn* [37], NN models were trained individually over the peptide chain properties, each of the four expression datasets, secondary structure data and each of the three amino acid sequence representations (i.e. PAAC, QSO and CTD). Further, Epanechnikov kernel was specified to weight the neighbours according to their distance.

PAAC representations (i.e. based on hydrophobicity, hydrophilicity and side chain mass) for the PAAC-NN model, were obtained using package *protr* [81] for all positive and negative protein instances. The sequence of target is the protein domain specific amino acid sequence, which is the concatenation of all domain specific subsequences in the protein primary amino acid sequence. The *protr* PAAC implementation uses $w=0.05$ by default as originally used by Chou et al. [80]. λ was set to 12. Unlike in PAAC model, QSOD vectors were computed for the protein subsequences made out solely of structurally exposed residues. This was decided upon the observation made by Naani et al. [85], suggesting that a residue-couple scheme is better at encoding surface amino acids. The surface exposure sequence was extracted from the original sequence, based on the exposure classification given by NetSurfP-1.1 [82]. An interesting observation was that, all exposed sequences start from residue M. CTD model was also built upon exposed amino acid sequences. Again, package *protr* [81] was used to obtain QSOD and CTD representations.

For the secondary structure model, the 11 length vector was computed, considering only the alpha helices and beta strands. Coil elements are ignored. Figure 4.3.4 illustrates how the two beta sheet based features are derived. If two beta strands are separated by alpha helices, the pair is considered to be forming a parallel beta sheet. Otherwise they are taken to be forming an anti-parallel beta sheet. [27] The alternating frequency was computed as the number of times $\alpha \rightarrow \beta$ or $\beta \rightarrow \alpha$ happens. The 4 different gene expression NN models were built upon Mnaimneh et al. [72], Chu et al. [73], Gasch et al. [74] and Spellman et al. [75], covering some important aspects of *S. cerevisiae* gene expression which lead to differential expression of proteins in certain circumstances. Expressions 1 measures the expression of all *S. cerevisiae* genes under a set of 215 titration experiments. Expressions 2 covers the transcriptional program of sporulation in budding yeast. Expressions 3 provides expression measures upon how yeast genes respond to DNA-damaging agents and how the regulatory role of yeast ATR homolog Mec1 is performed. Expressions 4 dataset gives the yeast cell cycle related expression through 4 time series experiments.



LDA based Model

In addition to the above main base models, an attempt was made to use LDA topic modeling for amino acid sequence representation. The number of topics was set to be two, as this is a binary classification task. All 2-mers, 3-mers and 4-mers (both overlapping and non-overlapping) were considered. It should be noted that, only the protein domain specific amino acid sequences were used for k-mer segmentation. Moreover, Yang et al. [94] incorporates motif patterns as words for further domain knowledge incorporation. Following a similar concept, this

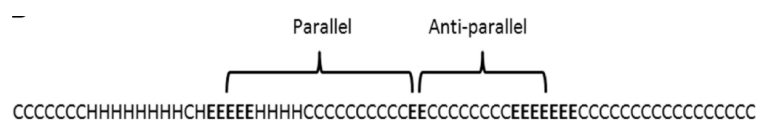


Figure 4.3.4: Parallel and anti-parallel β sheet formation [27]

study also tested the usage of protein domain tags and motif patterns as words, obtained from Gibbs sampling method. Gibbs Sampler method from package *tcR* [105] can find frequent motifs, by splitting each string in a string set, to k-mers of a given k. For this study, motifs of length 4 were considered.

All together, 8 word representations were individually applied for LDA topic modelling. Furthermore, the document term matrix was subjected to weighting with term frequencies (normalized). The decision was made upon the observation made by Yang [88], that the term frequencies give better results compared to tf-idf. R Package *TopicModels* [106] provides an implementation for constructing the term document matrix and modelling LDA. For each protein, the LDA model outputs two topic probabilities. These two topic probabilities along with true protein class labels were fed into an SVM classifier, for meta learning. SVM was decided to be the meta classifier due to its usage in [88]. It gives the final probability of a protein instance belonging to the positive class.

4.3.4 Base Model Combination Scheme



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Initially, five combination schemes: equal-weighted scheme; genetic algorithmically weighted scheme; AUC-weighted scheme; maximum posteriori probability hypothesis selection; and Bayesian network based combination scheme; were applied, in order to see each of its effectiveness in the context. Weight schemes assign a weight to each base model and take the weighted average of their individually outputted posterior probabilities as the final ensemble output. Maximum posteriori probability hypothesis selection scheme simply outputs the highest posterior probability given by a base model. Bayesian network based scheme combines base model outputs by referring to a Bayesian network. Finally, the 5 combination scheme ensembles are taken as individual base models for a second level ensemble (i.e. Ensemble of different combination schemes). It is simply the equal-weighted average of the base ensemble models.

Genetic Algorithm based Base Model Weighting

Genetic Algorithm (GA) [107] is an evolutionary computing approach that mimics evolution with basic darwinian concept of natural selection. Just as the species population of several variants undergo a natural test of fitness from generation to generation, where only the most fittest variants survive, a population of solutions to a problem can be made to go through a similar process. Chromosome crossovers and mutations decide a new generation to be tested for survival, allowing new parts of the target regions in the solution fitness landscape to be tested in successive generations. A crossover operation combines two parent solutions according to a certain combination scheme, in order to produce an offspring solution(s). The algorithm ensures that, only the offsprings of the fittest solutions will be passed on to the next generation. A mutation causes random part(s) of a solution to get mutated as to introduce a solution to be tested from a completely new region in the solution space. Ultimately, the possible solution space is searched in a heuristic manner to attain the optimal solution.



University of Moratuwa, Sri Lanka.

Electronic Theses & Dissertations

www.lib.mrt.ac.lk

The basic steps of the algorithm go as follow. Firstly a random set of solutions are generated as the initial population. Then, each solution is tested for fitness using an appropriate fitness function. Solutions with a low fitness value will not be forwarded to the next generation. The rest undergoes crossovers and mutations, in order to have new offsprings (variants) present in the new generation. The same procedure is repeated with new generational population until a best or a good enough solution is reached within the population. The pseudocode is as follow.

1. Generate a population of random chromosomes
2. Repeat (for each generation)
3. Calculate fitness for each chromosome
4. Rank the chromosomes according to their fitness
5. Repeat

6. Select pairs of parents
7. Generate offspring with crossover and mutation
8. Until a new population gets produced

Using GA as an optimization algorithm, the set of base models of an ensemble can be weighted to arrive at the final prediction [108]. In this context, a solution is a weight vector. The fitness function can be a performance evaluation function such as accuracy, precision, recall or the area under the ROC curve. An optimal weight vector can be obtained by running GA over the possible weight space.

In this study, the *ga* function in package *GA* [109] was used to search through the base model weight space for an optimal weight vector. The fitness was defined to be the Area under the ROC curve value. For this real valued solution space, following experimental setting was applied.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

- Local arithmetic crossover with 0.8 crossover probability: Given two weight vectors w_1 and w_2 , local arithmetic crossover firstly comes up with a random weight vector v from a uniform distribution. Then, the offspring weight vectors o_1 and o_2 are calculated as shown in Equation 4.24.

$$o_1 \leftarrow vw_1 + (1 - v)w_2 \quad \text{and} \quad o_2 \leftarrow vw_2 + (1 - v)w_1 \quad (4.24)$$

- Uniform random mutation with 0.05 probability: It selects a random weight position in the parent vector and changes the corresponding weight into a new weight, drawn from a uniform random distribution.
- Fitness proportional selection with fitness linear scaling
- 1000 maximum number of generations
- 100 population size

- 5% elitism (the percentage of the best solutions in a generation to be carried out to the next generation without any alteration)
- Minimum base model weight = 0; and maximum model weight = 1
- Optim = TRUE, specifying the algorithm to perform local search using a general-purpose optimization algorithm (i.e. L-BFGS-B)

Bayesian Network based Base Model Selection

This combination scheme firstly constructs a bayesian network over discrete base model predictions (i.e. positive -1; negative -0). Here, the threshold probability of a model is taken to be the value corresponding to the best optimal ROC point (i.e. closest top left point the in ROC curve). For a threshold t , if base model output $> t$, the protein instance is given 1 as the prediction. Otherwise it is given 0. The Bayesian network infers relationships between base models and the true label, considering their outputs for all the proteins in a training data set. R Package *bnlearn* [110] provides a variety of bayesian network learning methods. For this study, a network was learned using *bnlearn* [110] via a score based method in which, the score is the Bayesian Information Criterion; and the learning algorithm is the tabu search metaheuristic local optimization method. This network was used to select the set of base models which should be incorporated at ensemble combination. It selects only the direct parents and children of the true label node. For example, Figure 4.3.5 shows a Bayesian network derived over a train set. Here, all models except for PAAC, PPI_g and SS, are selected for giving a prediction to a test protein. The final probability is the average of: the mean of selected base model outputs; and the conditional probability of the true label being 1, given those selected base model discrete predictions.

4.3.5 Performance Measures

Following basic performance measures were reported for individual base models and ensemble models. They take into the account the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN), at classification.

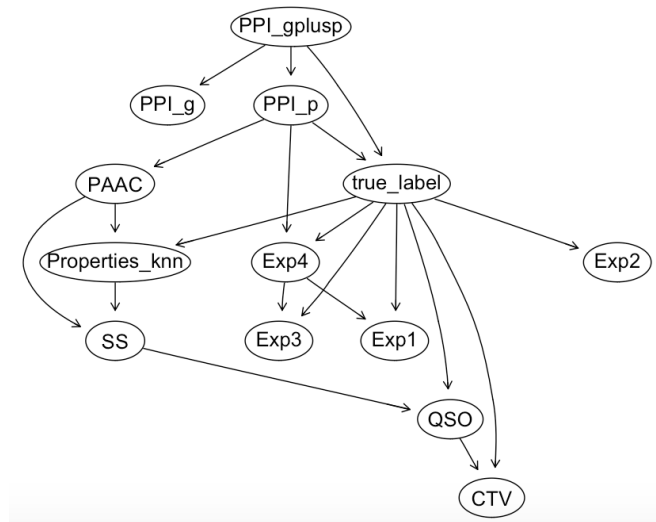


Figure 4.3.5: Example bayesian network

- Specificity/true negative rate = $\frac{TN}{TN+FP}$
- Sensitivity/true positive rate/Recall = $\frac{TP}{TP+FN}$
- Accuracy = $\frac{TP+TN}{TP+FN+TN+FP}$
- Positive predictive value/Precision = $\frac{TP}{TP+FP}$
- False positive rate = $\frac{FP}{FP+TN}$
- False negative rate = $\frac{FN}{TP+FN}$
- Negative predictive value = $\frac{TN}{TN+FN}$

University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
 www.lib.mrt.ac.lk

The primary performance evaluation measures were the Receiver Operating Characteristic and kappa statistic.

Receiver Operating Characteristic (ROC) [111]

When a binary classification model gives a prediction in the form of a probability, it is a continuous value which is also an estimation of the class membership. Given this classifier and a set of instances, the classifier probability outputs can be discretized upon a certain threshold, in order to decide the class of the instance. The confusion matrix upon classifier outputs specifies the number of true

positives, number of true negatives, number of false positives and the number of false negatives. When the threshold is varied, the confusion matrix values vary too. Thus, at each different threshold, a different TP rate/FP rate is observed.

An ROC plot is a two dimensional graph in which, TP (in y axis) is plotted against FP (in x axis). TP is the sensitivity, while FP is (1-specificity). As the threshold can vary from 0 to 1 over the classifier probability outputs, a distinct pair of $\langle TP, FP \rangle$ can be obtained for each threshold value. These pairs are reflected through points on an ROC plot. Since the threshold is also a continuous value, the final ROC plot contains an ROC curve made out of such points.

The optimal threshold corresponds to the ROC point with the best possible trade-off between TP rate and FP rate. There are certain regions on the ROC plot that recognizes different degrees of the classifier performance. The upper right most point accounts for $TP = 1$ and $FP = 0$, denoting a perfect classification. Any point on the diagonal denotes the case where $TP = FP$, which reflects a random performance. The upper triangular region includes points where a classifier performance is better than random chance, while the lower triangular region includes points where a classifier performs worse than random guessing. Therefore, only the classifiers which fall on the upper triangular region are considered as effective.

In overall, ROC curve shows the ability of a classifier to identify positive examples relative to negative examples. And the Area Under the Curve (AUC) reflects the classifier performance: whether it is better than random chance or not. An AUC value closer to 100% simply indicates a very good classification, with a high sensitivity and high specificity.

Kappa Statistic

Kappa statistic is a statistical measure to evaluate the inter-rater agreement between multiple raters. There are various versions of this measure to be used

in different scenarios. Kappa statistic value of 0 indicates poor agreement, while value 1 indicates a perfect agreement. Viera et al. [112] provides a common scale as in Figure 4.3.6. The commonly used statistic is the Cohens' kappa statistic which evaluates the agreement between two parties. Fleiss kappa statistic [113] is a generalization of the unweighted Cohens' kappa. This statistic is based on the difference between the observed agreement and the expected agreement among base models [112]. Suppose N = the number of proteins; m = the number of base models; n_{i+} = the number of base models that assign i^{th} protein to the positive class; and n_{i-} = the number of base models that assign i^{th} protein to the negative class. Then, the observed agreement P_o is defined by the Equation 4.25.

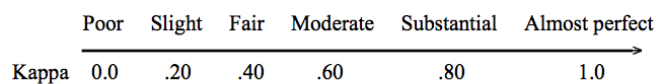
$$P_o = \frac{\sum_{i=1}^N p_i}{N} \quad \text{where} \quad p_i = \frac{n_{i+}^2 + n_{i-}^2 - m}{m(m-1)} \quad (4.25)$$

The expected agreement P_e is defined by the Equation 4.26.

$$P_e = P_{(+)}^2 + P_{(-)}^2 \quad \text{where} \quad P_{(+)} = \frac{\sum_{i=1}^N n_{i+}}{Nm} \quad \text{and} \quad P_{(-)} = \frac{\sum_{i=1}^N n_{i-}}{Nm} \quad (4.26)$$

Finally the Fleiss Kappa statistic is obtained by the Equation 4.27. [113]

$$kappa = \frac{P_o - P_e}{1 - P_e} \quad (4.27)$$



<u>Kappa</u>	<u>Agreement</u>
< 0	Less than chance agreement
0.01–0.20	Slight agreement
0.21–0.40	Fair agreement
0.41–0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–0.99	Almost perfect agreement

Figure 4.3.6: Kappa scale [112]

Chapter 5

EXPERIMENTATION, RESULTS, ANALYSIS AND DISCUSSION

This chapter discusses the results and observations, obtained from the carried-out experimentation to assess a properly engineered heterogeneous data ensemble, for classifying *Saccharomyces cerevisiae* proteins under ‘mitochondrion organization’ Biology process. After a thorough background study and literature review, the explained methodology in Chapter 4 was applied to evaluate the effect of diverse biological data incorporation, in formulating the particular functional context for supervised learning. Six data types: amino acid sequences; protein domain data; gene expression; peptide chain properties; secondary structure data; and interaction data, were utilized for the purpose. Specifically, a Genetic Algorithm based weight scheme was compared with four baseline combination schemes, in order to fuse base model probability outputs. In addition, an LDA topic modeling based sequence representation and a second level combination scheme ensemble were evaluated as well. The rest of the sections describe the experimental setup at first, followed by a result analysis and discussion of the performance evaluations.

5.1 Experimental Setup

The objective of setting up experiments was to come up with a technique that gains a fairly high accuracy in *Saccharomyces cerevisiae* protein classification under ‘mitochondrion organization’, utilizing a variety of data types. The concern was not on the computational complexity, but on the classification ability of the approach. Experiments were performed mainly to validate the Genetic Algorithmically weighted heterogeneous data ensemble. It was done with regards to a comparison with four other combination schemes and a second level combination scheme ensemble. Prior to ensemble model construction, a topic modeling based

approach was employed as described in Chapter 4, in order to validate its capability to accurately represent a protein instance (in terms of the amino acid sequence).

The experimental setup was decided upon its suitability to address the high class imbalance, the need for an appropriate performance measure, and the ability to provide a strong measure of the true classifier performance. As the initial step, the high class imbalance issue was addressed by preparing 10 annotation data samples with 1:1 positive to negative example ratio. Each sample contains all 239 positive examples and a randomly selected 239 negative examples out of the 3880, from the benchmark gold dataset published by Huttenhower et al. [69]. Thus, the samples vary only in terms of the negative example representation.

Experimental evaluation of a model was carried out by performing leave-one-out cross-validation for each of the 10 samples. A cross-validation technique supports model evaluation in terms of its generalization capability, avoiding over-fitting to some extent. In leave-one-out cross-validation, each data instance in the sample is left out and a model is trained over the rest. Then the trained model is used to predict the class for the particular left-out data instance. The model is evaluated upon such predictions made for all the data instances in the sample.

The primary performance measure was taken to be the Area Under the Curve (AUC) value of the Receiver Operating Characteristic (ROC). The measure is equivalent to the probability of ranking a randomly chosen positive instance higher than a randomly chosen negative instance [111]. It was decided to be used for several important reasons. Firstly, ROC is considered and widely used as a standard classifier performance measure nowadays. Thus, the results from this study could be used as a baseline for comparison by future researchers in the area. Moreover, the ROC space facilitates the exploration of different ROC points to obtain the optimal threshold for classification. Further, the measure is insensitive to the changes in class distribution. Davis et al.[114] proves that a

curve dominates in ROC space, if and only if it dominates in the Precision Recall (PR) space.

The other measures included specificity, sensitivity, accuracy, true negative rate (tn), true positive rate (tp), false positive rate (fp), false negative rate (fn), positive predictive value (ppv), negative predictive value (npv) and kappa statistic where applicable. The final performance measure vector in any model evaluation was taken to be the average measure vector, which was observed over the 10 samples.

Experiments were conducted for the following tasks using the described experimental setup.

- Evaluation of the LDA topic modeling based approach to incorporate domain specific amino acid sequences in protein function prediction: to see its effectiveness in giving an accurate representation of a *S. cerevisiae* protein in the context of mitochondrion organization.
- Finding the optimal number of neighbours to be used in nearest neighbour models: to ensure that a nearest neighbour classifier scans the most optimal set of neighbours which may strongly adhere to the ‘guilt-by-association’ concept.
- Evaluation of the individual base model performances and their inter-rater agreement: to check their eligibility in ensemble formation and to decide on individual contributions for collective decision making.
- Running the standard Genetic Algorithm to obtain the optimal weight vector for the heterogeneous data ensemble: to optimize the classification accuracy, while receiving an insight to the importance of each data type.
- Evaluation of the heterogeneous data ensemble under different combination schemes and the second level ensemble scheme: to compare the performance of Genetic Algorithm based weight scheme with a set of baseline

combination schemes, and to evaluate the extent of accuracy an ensemble of combination schemes can achieve.

5.2 LDA Topic Modeling based Approach

The Latent Dirichlet Allocation topic modeling approach for protein sequence representation was evaluated using the aforementioned experimentation setup. A model was constructed according to the method described in Chapter 4. Eight word representations: Domain tags; motif tags; and both overlapping and non-overlapping k-mers for k=2,3 and 4, were tested. Figure 5.2.1 presents ROC plots of each representation, for all 10 samples. Corresponding performance measures are presented in Table 5.1.

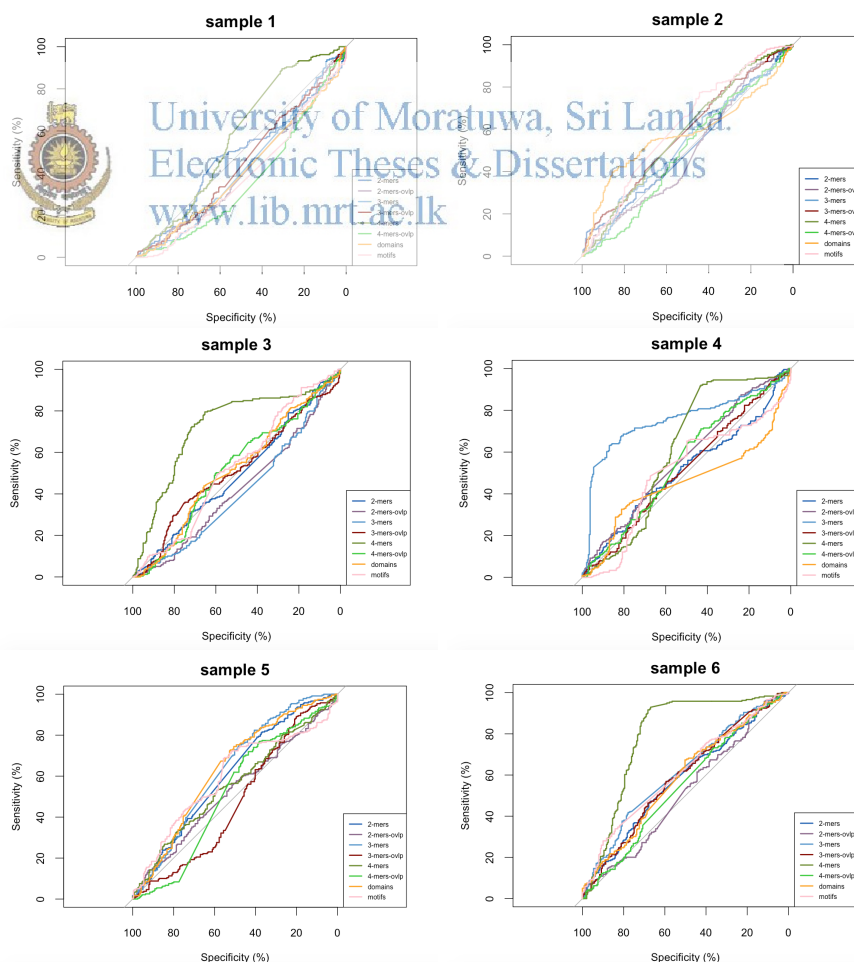
Accordingly, LDA topic modelling based approach has not been able to accurately model a protein sequence in the functional context of ‘mitochondrion organization’. Even though literature such as [88] suggests a higher sensitivity and specificity gain from LDA topic modeling for sequence representation, the results of this study suggest otherwise. All average AUCs are in the range of 50%-59%, indicating that the model is only slightly better than random chance. Non-overlapping 4-mer representation gives the highest mean AUC value of 58.74%, the highest sensitivity of 63.43% and the highest accuracy of 59.51%, implying the importance of considering 4 adjacent amino acid residues. However, it is not

Table 5.1: LDA model based approach evaluation results

Model	AUC	threshold	specificity	sensitivity	accuracy
Domains	51.8768	0.5000	57.5127	54.2260	55.8694
2-mers	54.3161	0.5015	56.3356	54.2260	55.2827
2-mers (ovlp)	50.7832	0.5001	49.1484	55.5230	52.3331
3-mers	57.1135	0.5009	52.0708	63.0962	57.5846
3-mers (ovlp)	53.9766	0.4970	51.6522	57.6569	54.6558
4-mers	58.7402	0.5057	55.5823	63.4310	59.5078
4-mers (ovlp)	51.5816	0.4996	47.0110	60.6695	53.8403
Motifs	52.3257	0.4973	52.3290	56.2762	54.2991

a significant AUC value for incorporating the model as a base model in an ensemble. The least performing model is the overlapping 2-mer topic model with a 50.78% AUC. It is almost a by-chance model. The highest specificity of 57.51% was gained by the Domains tag representation, while the least (i.e. 47.01%) was given by overlapping 4-mer representation. All threshold values are around 0.5. Further, some samples show unexpected improvements with some representation types. For instance, sample 3 and sample 6 non-overlapping 4-mer representations give a much higher AUC value than the rest. Similar observation can be made from sample 4 non-overlapping 3-mer representation as well.

Observations from this domain-specific sequence representation by LDA topic modeling is quite surprising, since similar proteins are somewhat expected to share functional domains. However, functionally unrelated proteins can also have



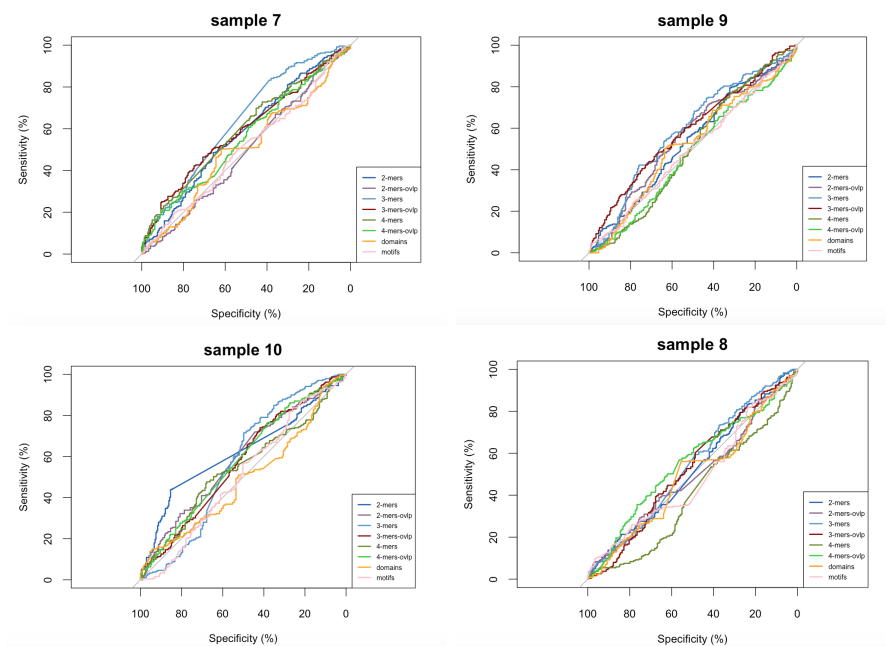


Figure 5.2.1: ROC plots for the LDA model based approach

the same domain. Such domains usually account for a common and essential function. Thus, not all domain information will be useful for function prediction. Moreover, some domains may act as lock-key pairs. When a certain biology process requires two proteins to interact with each other, those two proteins may have two different domains, which account for each of their interaction site. Thus, even though it is the same function, the proteins can have different domains for the purpose. The same explanation can be given with regards to motifs, as sequence and structural motifs tend to repetitively appear in functionally unrelated proteins as well. Furthermore, the selection of two topics might be inaccurate. It can be the case that the positive class and the negative class is represented by different number of topics.

5.3 Optimal Number of Neighbours

An experiment was also carried-out to empirically decide on the number of proteins k in the neighbourhood, to be covered by the nearest neighbour models as explained in Chapter 4. The optimal k for all nearest neighbour base models was decided upon the performance of their equal-weighted ensemble (with the

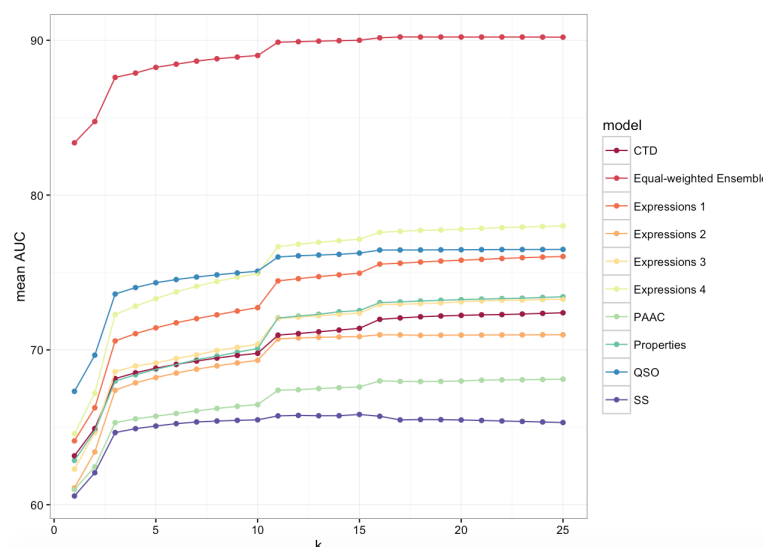


Figure 5.3.2: k vs mean AUC

weight of each base model as 1) when varying k from 1 to 25. In this case, the ensemble only consists of kNN models. Figure 5.3.2 shows the k vs mean AUC for each NN model, as well as for the Equal-weighted ensemble. As expected, the mean AUC increases with the increase of k. All models show a significant rise in their mean AUCs from k=1 to k=3. Starting from k=3 to k=25, all models gradually arrive at a stable AUC value. The significant gap between base model performances and the ensemble performance proves the promising nature of the ensemble approach. Expressions 4 model surpasses the QSO model at around 10 neighbours, and becomes the best performing nearest neighbour base model. The least performing NN base model is the Secondary Structure (SS) model. PAAC model is getting much better than the SS model with the increase of k, as their gap gets wider.

For the Equal-weighted ensemble, the highest mean AUC of 90.2140% with specificity 84.2200% and sensitivity 82.74557%, is given by 17 neighbours. The accuracy at that point is 83.4828%. Therefore 17 was selected as the optimal number of neighbours to be used in NN models.

5.4 Base Model Evaluation

An ensemble performance relies on the performances of its base models. Just as the literature states, a diverse set of base models with each of its error rate being less than 0.5 is required to arrive at a ensemble high performance. Therefore, the individual model performance measures were obtained prior to forming the ensemble. Table 5.2 and Table 5.3 present the individual model performances.

Accordingly, the best performing base classifier is the Genetic plus physical interactions model with an average AUC of 78.86%, specificity of 74.21% and a sensitivity of 76.21%. When comparing the three affinity-based neighbourhood models, the Physical interactions model shows a significant increase of AUC compared to the Genetic interactions model. This may indicate that physical interactions are better than genetic interactions at identifying the protein functional context. However, only a small amount of genetic interactions are publicly available due to limited genetic interaction mapping studies. Therefore, it may also be the case that, a set of genetic interactions that are good indicators of ‘mitochondrial organization’ were not present in the used interaction dataset.

Model	AUC	threshold	specificity	sensitivity	accuracy
CTD	72.15	0.27	70.92	62.89	0.67
Domains	50.36	0.96	27.53	75.98	0.52
Expressions 1	75.66	0.53	70.22	69	0.7
Expressions 2	70.97	0.52	62.47	68.12	0.65
Expressions 3	72.96	0.55	66.98	68.41	0.68
Expressions 4	77.73	0.55	70.01	72.59	0.71
PAAC	67.93	0.5	59.16	68.12	0.64
Genetic interactions	71.72	0.6	66.56	72.28	0.7
Genetic plus physical	78.86	0.59	74.21	76.21	0.75
Physical interactions	77.14	0.57	72.62	77.32	0.75
Properties	73.14	0.51	66.99	70	0.69
QSO	76.46	0.36	73.64	66.86	0.7
Secondary Structure	65.27	0.51	59.49	62.76	0.61

Table 5.2: Individual base model performance results I

Model	tn	tp	fn	fp	npv	ppv
CTD	169.5	150.3	88.7	69.5	65.67	68.63
Domains	65.8	181.6	57.4	173.2	53.47	51.18
Expressions 1	166.7	164.9	74.1	70.7	69.33	70.03
Expressions 2	149.3	162.8	76.2	89.7	66.28	64.54
Expressions 3	156.2	158.7	73.3	77	68.16	67.35
Expressions 4	162.3	168.4	63.6	69.5	71.88	70.86
PAAC	141.4	162.8	76.2	97.6	65.04	62.62
Genetic interactions	130.9	155.3	59.6	65.5	68.72	70.44
Genetic plus physical	165.9	176.7	55.2	57.6	75.07	75.55
Physical interactions	132	157.9	46.6	49.8	74.08	76.09
Properties	159.7	167.3	71.7	78.7	69.09	68.07
QSO	176	159.8	79.2	63	69.02	71.86
Secondary Structure	142.18	150	89	96.82	61.56	61.09

Table 5.3: Individual base model performance results II

The Genetic plus physical interactions model surpasses both of those interaction models as expected, because it takes into account both physical and genetic interactions which can give a much stronger clue. For further evaluation, one-way ANOVA test (DF1=2, DF2=27) was performed and the F-statistic of 15.39 was observed, implying significant difference among the three affinity-based neighbourhood models. The highest significance of difference is between the combined interaction model and the Genetic interactions model, with a 0.0000380 p-value according to the post hoc Tukey's test.

The least performing base model is the Domains model with a mean AUC of 50.3608%. It is worse than random chance and thus, it was not incorporated into the ensemble as a base model. The next least performing base models are the Secondary Structure model with a 65.27% mean AUC and PAAC model with 67.93% mean AUC. Secondary Structure model was expected to perform better, as it utilizes 11 types of secondary structure element information. Rest of all the models display a mean AUC above 70%.

Out of the sequence representation models, QSO is the best performer with a

mean AUC of 76.46%. PAAC may have been the least performing sequence model due to its full domain specific amino acid sequence coverage. Both QSO and CTD models are based on exposed structure based sequence. Thus, it is clear that the incorporation of amino acid residue exposure details is necessary to effectively represent functional details from a sequence. One-way ANOVA test (DF1=2, DF2=27) yielded a very high significant difference among the 3 sequence models, with an F statistic of 58.76. Tukey's HSD showed that the most significant difference lies between QSO model and PAAC model, with a p-value of 0. The difference between QSO and CTD is also significant with $p=0.0000248$.

Among gene expression data models, Expressions 4 model is the best performing model with an average AUC of 77.73%, implying the importance of yeast cell cycle details for 'mitochondrion organization' function. The rest of the expression models also show a considerably good mean AUC for a base model. One-way ANOVA test (DF1=3, DF2=36) gave an F statistic of 15.57 with a 0.00000118 p-value, indicating a very high significance of difference between the 4 expression models. Tukey's test showed that Expressions 4 and Expressions 2 models differed most significantly at $p = 0.0000014$. Expressions 3 and Expressions 2 are the least significantly differed pair with a p value of 0.2586336.

All the models except for the Domains model, show a diverse range of AUC values, indicating their potential to be incorporated into an ensemble that can give a significant improvement of performance. Moreover, CTD, Expressions 1 and QSO models are better at identifying true negative protein examples, than the rest of the models that are better at true positive protein example recognition.

5.5 Evaluation of the Inter-rater Agreement

The diversity between all base models can be measured through the Fleiss kappa interrater agreement statistic measure [113]. The inter-rater agreement was evaluated for the Equal-weighted ensemble as to obtain an unbiased measure of the

agreement between the base models. Fleiss kappa statistic was obtained for all 10 samples using R Package *irr* [115]. Table 5.4 presents the kappa values received for each sample. They were measured by considering only the protein instances, for which all raters give an output. On average, all 12 base raters give a prediction to 333.9 proteins (69.85356% of all proteins in a sample). At classification, the best threshold value which corresponds to the closest top-left ROC point, was taken for each base model. For the 12 base raters, a mean kappa of 0.2352 with standard deviation 0.0162, mean z value = 34.9020 and p-value = 0 were observed. The null hypothesis is that the kappa = 0. According to the common scale given in [112], a value between 0.21 to 0.40 implies a fair agreement between the base models. Hence it can be concluded that the heterogeneous data models in this ensemble are in a fair agreement.

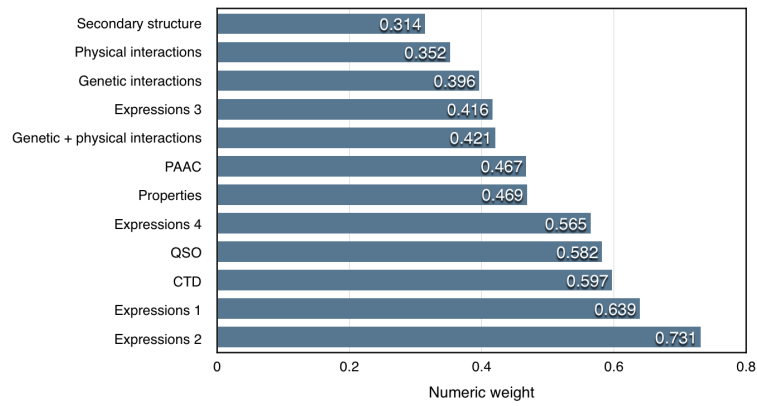
5.6 Genetic Algorithm based Weight Optimization

Aforementioned experimental setup was also used to retrieve the optimal weight vector, by running the standard genetic algorithm under the specification mentioned in Chapter 4. Figure 5.6.3 (a) presents the optimal weights allocated to each base model in the GA-weighted ensemble, accordingly. Figure 5.6.3 (b) presents the mean ROC AUC values of each base model.

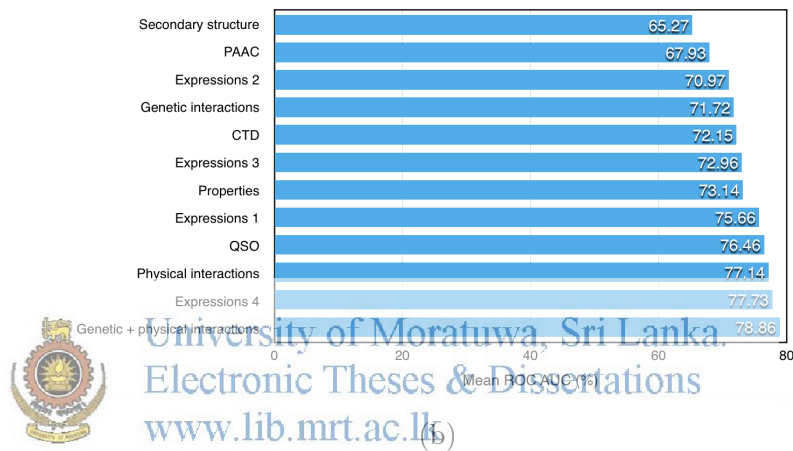
Sample	Number of proteins	Kappa	Z-value
1	334	0.2348	34.8618
2	327	0.2326	34.1640
3	359	0.2454	37.7744
4	321	0.2239	32.5847
5	326	0.2485	36.4507
6	328	0.2039	29.9946
7	312	0.2657	38.1286
8	350	0.2334	35.4796
9	335	0.2308	34.3180
10	347	0.2330	35.2641

Table 5.4: Kappa measure for individual samples

There were 21 weight vectors that got resulted from running GA over each



(a)



(b)

Figure 5.6.3: (a) GA optimized weights (b) mean ROC AUC of base models

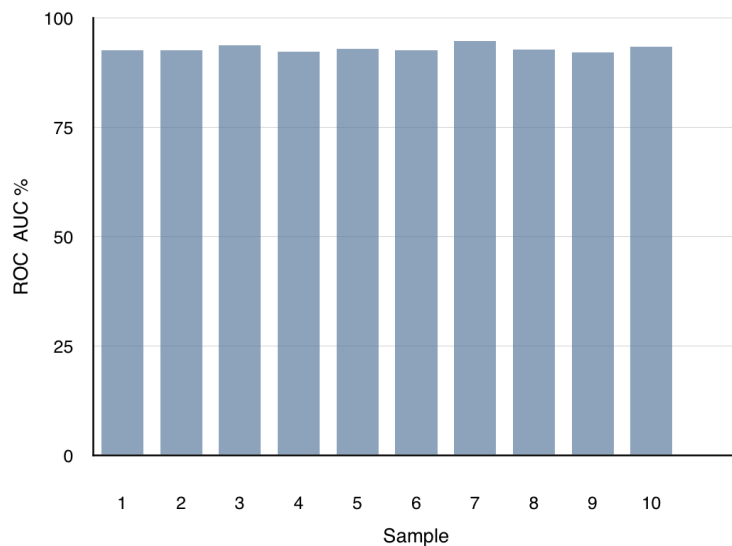


Figure 5.6.4: Best fitness value over each sample

of the 10 samples, as sample 1 and sample 7 gave multiple optimal weight vectors (i.e. 3 vectors and 10 vectors, respectively). Figure 5.6.4 shows the bar chart of best fitness values in each sample. Table 5.5 presents all sample-wise weight vectors. The GA experiment has taken 45.1 average number of iterations with 72 maximum and 20 minimum iterations. The fitness function is the AUC under ROC of the GA-weighted ensemble. The average, attainable fitness over the 10 samples is 92.939% with a 0.7677 standard deviation. The maximum fitness shows to be 94.6307%, while the minimum fitness is 92.038%. The final optimal weight vector for the base models is taken to be the mean weight vector of all 21 vectors. Figure 5.6.4 shows only slight variations among the attained best fitness values for each sample, positively supporting to take the average weight vector.

Accordingly, the most weighted model is the Expressions 2 model, while the Secondary Structure model being the least weighted model. The most weighted base model reflects the importance of the temporal program of gene expression during meiosis and spore formation when predicting proteins in the context of mitochondrial biogenesis. The least weighted Secondary Structure model, indicating the lowest support for the functional context formation from the alpha helices and beta sheet based 11 feature vector.

An important observation is the mismatch between the base model rankings that are visible in GA weights and the mean AUCs. It somewhat provides a contradiction, as both: numeric weight and performance rate can be taken as indicators of the biological data type significance. As the Figure 5.6.3 illustrates, only three base models (i.e. Secondary Structure, Properties and QSO) hold the same rank, leading to a fair acceptance of their relative standings. Others differ vigorously. The reason might be perhaps the lack of decision making ability in the affinity-based neighbourhood models; they do not always provide a prediction for a protein due to missing interactions. The weights might have been adjusted at GA optimization, in order to cater for that. Moreover, we cannot guarantee to receive a global optimal weight vector from GA. The resultant dif-

ference might have been caused due to that as well. The mismatch can also be due to the implicit difference between the two indicators (i.e. weight and AUC measure). While AUC provides each base models' individual capability to form the functional context, perhaps GA numeric weights are more evidential of each others' tendency to contribute towards collectively carrying-out the actual function. However, this is a very convoluted interconnection and needs more biology expertise, experiments and verification for gaining a fair understanding.

Nevertheless, a strong conclusion can be made regarding the Secondary Structure model, since it is the least performing base model, as well as the least weighted base model. The secondary structure feature vector does not effectively capture the biologically significant structural variations among the positive and negative protein examples. Moreover, according to the results from Section 5.4, the model sensitivity is greater than specificity (i.e. 62.76% > 59.49%), indicating that the feature vector is better at capturing positive structural variations than the negative structural variations.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

During the iterations of running GA over sample 7, it finds 10 weight vectors which give the highest AUC value (i.e. the maximum fitness) across all 10 samples. It should also be noted that the order of weights was always the same when multiple optimal vectors are present for a sample. If the average weight order is compared with the maximum fitness giving weight vector, Figure 5.6.5 illustrates a fair observation that can be made regarding the relative importance

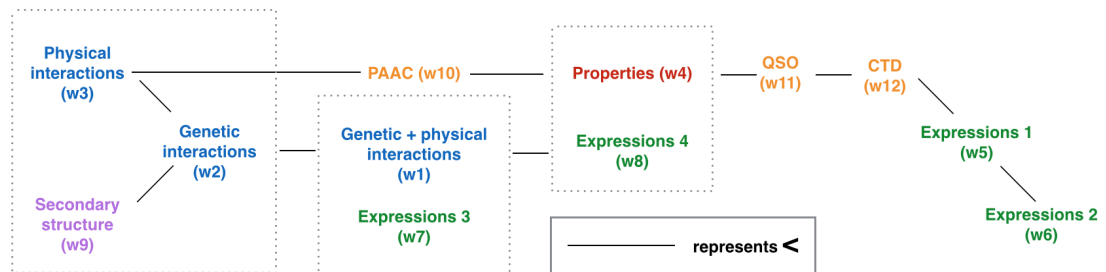


Figure 5.6.5: Order of data types with respect to both average and maximum fitness giving weight vectors

of each data type. Table 5.5 includes the 10 weight vectors obtained for the maximum fitness giving sample (i.e.sample 7). The average of those were taken for the comparison. It can also be argued that the average order of weights is more biased towards the order of weights resultant from sample 7. However, the bias could be justified under the assumption, that it has a good representation of positives and negatives, indicated by its highest performance.

When models of the same type are trained over the 10 samples, the varied behavior of each model can be considered due the random and varying negative example representations. Hence, the best fitness sample could be the best representation of a negative sample with respect to the positive set (current GO annotations). In a hypothetical scenario, the change in weight vectors will also be due to the same reason, given the assumption that the GA gives a global optimal solution. The weight of each base model is getting adjusted to arrive at the best fitness value. However, we cannot guarantee a global optimal from GA and thus, weights might not be completely reliable for decision making.



Sample	Fitness	Iterations	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12
1	92.4984	36	0.3472	0.3741	0.3141	0.3185	0.7081	0.7248	0.5833	0.5029	0.1569	0.5217	0.5545	0.7746
1	92.4984	36	0.3471	0.3716	0.3077	0.3215	0.7049	0.719	0.6011	0.4606	0.1553	0.5163	0.5564	0.7759
1	92.4984	36	0.346	0.3735	0.3234	0.3223	0.7156	0.7163	0.5568	0.4532	0.1592	0.5034	0.551	0.7775
2	92.6297	32	0.6058	0.5661	0.6105	0.2539	0.7036	0.633	0.2131	0.478	0.027	0.5631	0.689	0.7966
3	93.6888	20	0.2402	0.5054	0.3502	0.4606	0.7382	0.8402	0.2799	0.5901	0.447	0.4647	0.4748	0.529
4	92.2375	57	0.3856	0.2942	0.4486	0.5447	0.5292	0.6374	0.1397	0.7549	0.4421	0.5665	0.6179	0.5028
5	92.9378	48	0.3531	0.0906	0.4076	0.5074	0.5551	0.6606	0.2986	0.7308	0.3697	0.2309	0.6879	0.6243
6	92.6227	48	0.3119	0.3142	0.3726	0.6157	0.6787	0.6685	0.3499	0.5896	0.1897	0.4718	0.6247	0.6006
7	94.6307	56	0.4565	0.4553	0.3032	0.5299	0.6537	0.7718	0.4822	0.5252	0.4278	0.4156	0.5608	0.5395
7	94.6307	56	0.4568	0.455	0.3029	0.5286	0.662	0.7723	0.4821	0.5252	0.4271	0.4158	0.5613	0.5396
7	94.6307	56	0.4568	0.455	0.3029	0.5285	0.6537	0.7715	0.4814	0.5258	0.4274	0.4158	0.5614	0.5398
7	94.6307	56	0.4568	0.4551	0.303	0.5297	0.6614	0.7722	0.4821	0.5252	0.4272	0.4157	0.5612	0.5396
7	94.6307	56	0.4566	0.4551	0.3032	0.5309	0.6548	0.7723	0.4818	0.5254	0.4295	0.4154	0.5608	0.539
7	94.6307	56	0.4567	0.4553	0.303	0.5298	0.6606	0.7719	0.4821	0.5252	0.4273	0.4157	0.5609	0.5396
7	94.6307	56	0.4567	0.4554	0.303	0.5298	0.6537	0.7718	0.4822	0.5252	0.4272	0.4156	0.5608	0.5395
7	94.6307	56	0.4567	0.4552	0.3031	0.5308	0.663	0.7711	0.4818	0.5256	0.4274	0.4156	0.561	0.5394
7	94.6307	56	0.4567	0.4552	0.303	0.5287	0.6612	0.772	0.4821	0.5252	0.4272	0.4158	0.561	0.5396
7	94.6307	56	0.4568	0.4562	0.3031	0.5311	0.6566	0.7707	0.4834	0.5251	0.4273	0.4159	0.5613	0.5396
8	92.7697	41	0.1864	0.4803	0.4653	0.5964	0.6334	0.7353	0.2779	0.7069	0.048	0.4656	0.5714	0.6003
9	92.038	72	0.5529	0.3159	0.2913	0.1363	0.5261	0.6097	0.299	0.9108	0.0681	0.6058	0.6219	0.5132
10	93.337	41	0.5922	0.0824	0.473	0.4644	0.3474	0.6952	0.3203	0.4268	0.2656	0.7482	0.6573	0.6496

Table 5.5: GA optimized weights for all 10 samples

5.7 Ensemble Classification Performance

Firstly the 12 base models were incorporated into five ensembles, through the 5 combination schemes as described in Chapter 4. The performance of each ensemble was evaluated using the previously described experimentation. Furthermore, a second level ensemble of these five ensembles was also evaluated. Table 5.6 and Table 5.7 give out the obtained performance measures. Figure 5.7.6 illustrates the sample-wise ROC plots of each base model, compared with the GA-weighted ensemble. Figure 5.7.7 presents the sample-wise ROC plots for the 5 ensemble classifiers and the second level ensemble classifier.

Accordingly, the GA-weighted ensemble gains the highest mean AUC value of 92.52% with 86.44% specificity, 85.36% sensitivity and an accuracy of 86%. It improves the best performing base classifier (i.e. 78.86% AUC of Genetic + physical interactions model) by 17.32%. The same accuracy as the GA-weighted ensemble

Model	auc	threshold	specificity	sensitivity	accuracy
AUC-weighted	92.03	0.52	86.36	84.35	0.85
Bayesian net	91.3	0.5	85.4	83.26	0.84
Combination scheme	92.02	0.6	86.78	83.1	0.85
Equal-weighted	91.97	0.51	86.32	84.73	0.86
GA-weighted	92.52	0.5	86.44	85.36	0.86
Highest probability	77.92	0.9	68.28	83.43	0.76

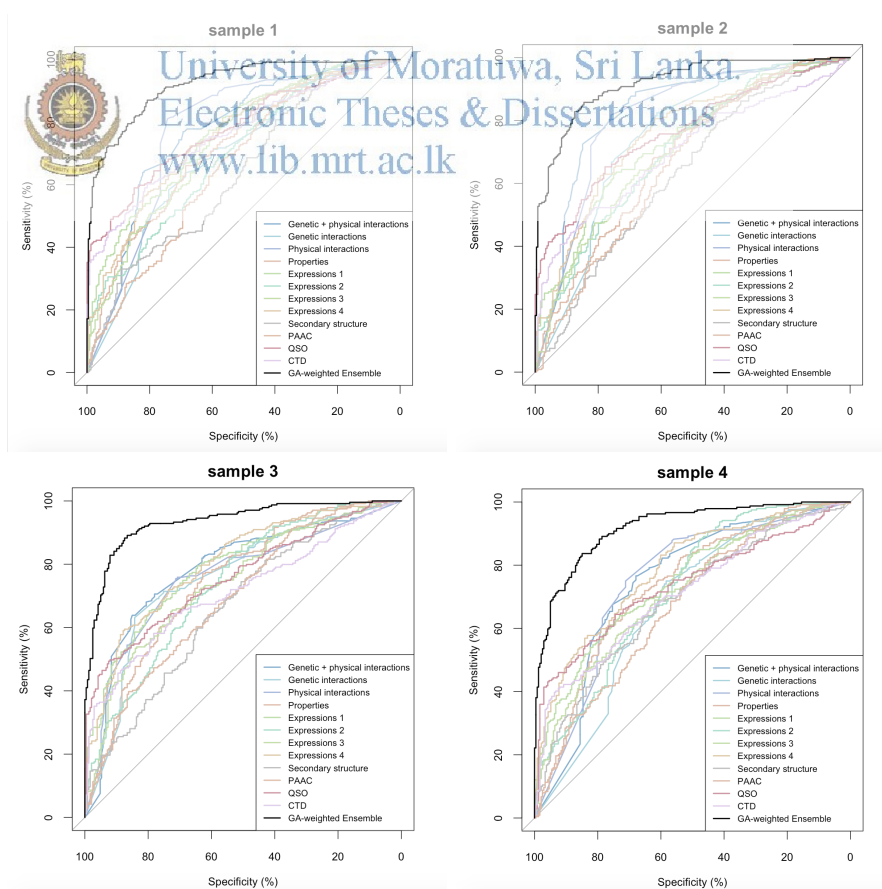
Table 5.6: Ensemble performance results I

Model	tn	tp	fn	fp	npv	ppv
AUC-weighted	206.4	201.6	37.4	32.6	84.69	86.16
Bayesian net	204.1	199	40	34.9	83.65	85.1
Combination scheme	207.4	198.6	40.4	31.6	83.75	86.42
Equal-weighted	206.3	202.5	36.5	32.7	85.03	86.21
GA-weighted	206.6	204	35	32.4	85.52	86.38
Highest probability	163.2	199.4	39.6	75.8	80.53	72.56

Table 5.7: Ensemble performance results II

ble, also obtained by the Equal-weighted ensemble, except for its 91.97% AUC. It improves the best base model by 16.62%. The least performing ensemble is the Highest probability ensemble with a 77.92% AUC, which is significantly lower than that of the rest and also lower than the best base model. AUC-weighted ensemble gives an AUC of 92.03%, which is even better than the equal-weight scheme. The improvement is 16.7%. However, its accuracy has dropped by 1%. Bayesian net ensemble improves the best base model by 15.77%. These results suggest that the best ensemble is the GA-weighted ensemble. However, the second level ensemble does not seem to be effective. Its mean AUC of 92.02% is lower than both, AUC-weighted and GA-weighted ensembles. This suggests that the ensemble models are not diverse in their predictions, leading to poor capability in correcting each others' mistakes.

The one-way ANOVA test ($DF1=4, DF2=45$) was performed to evaluate the sig-



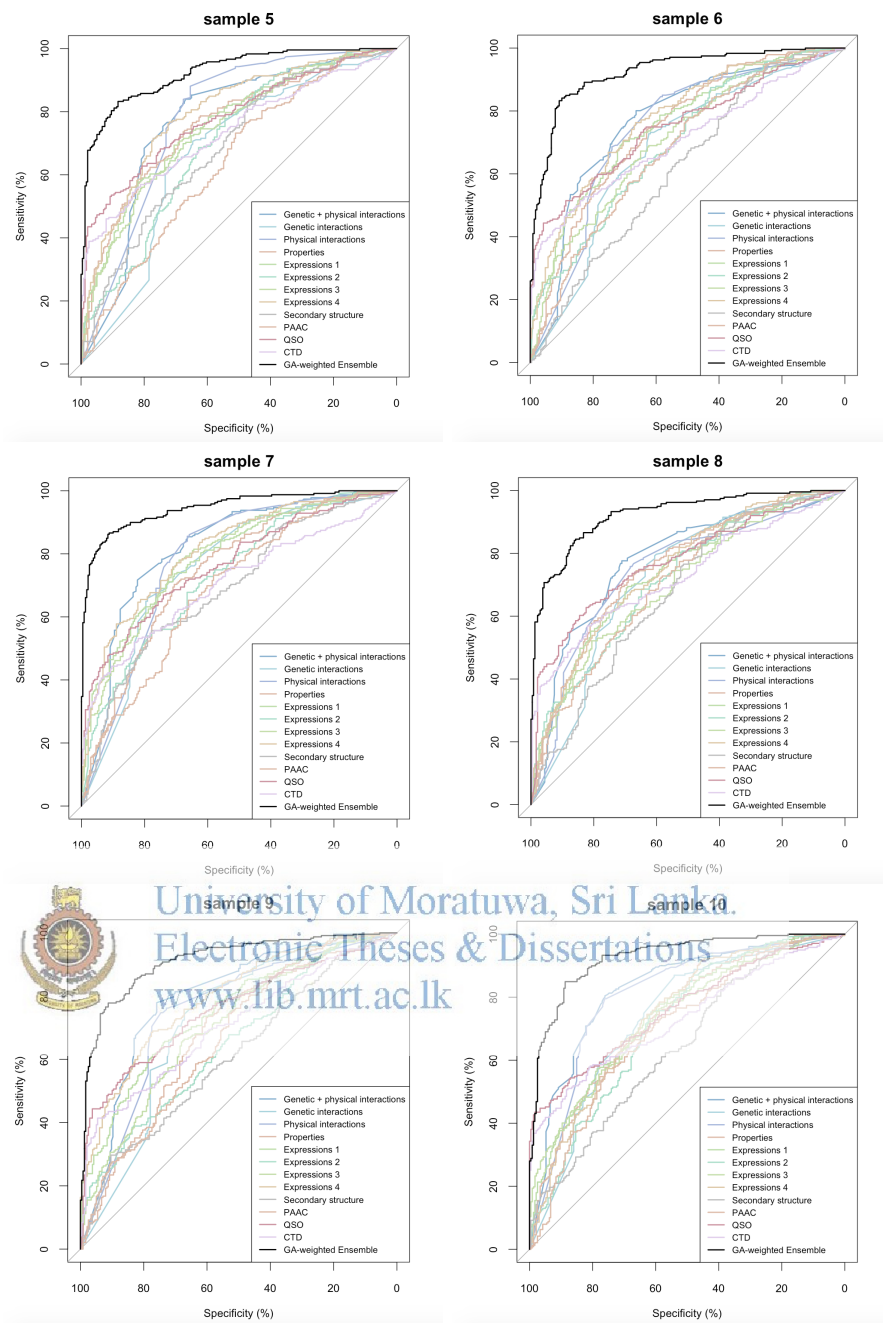
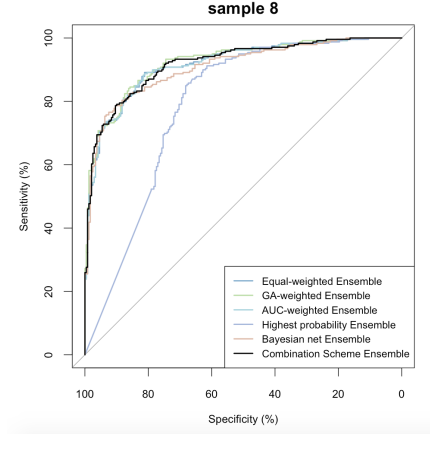
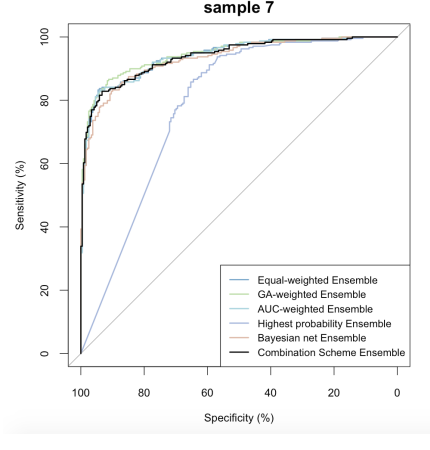
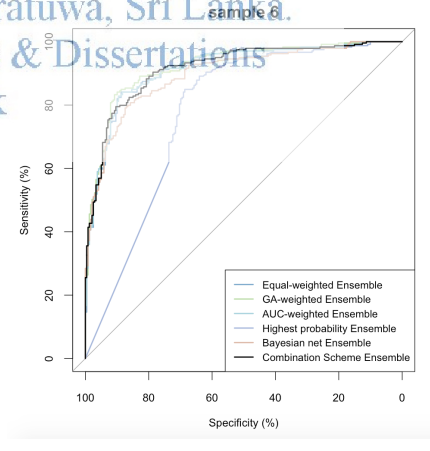
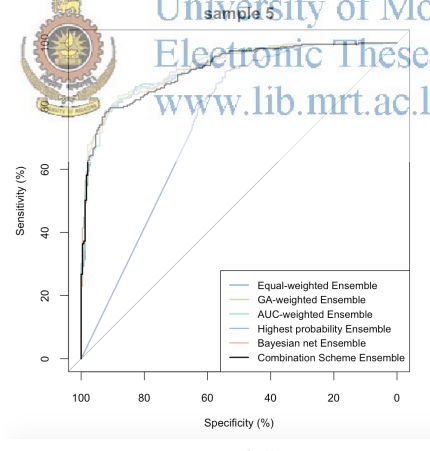
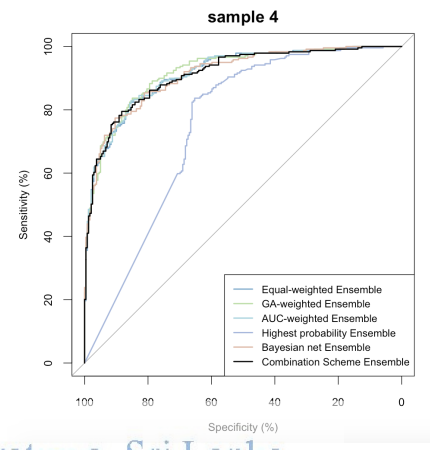
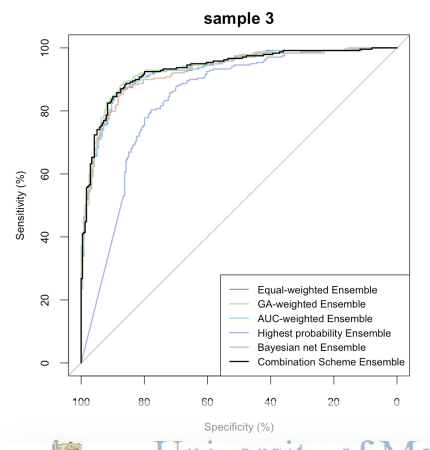
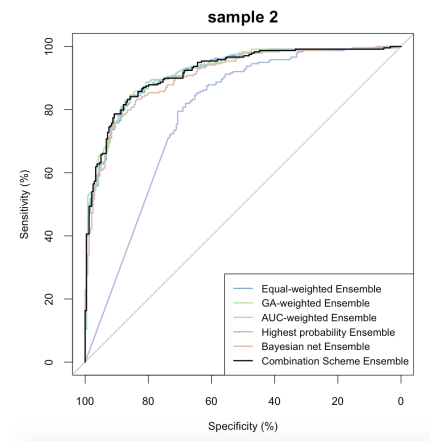
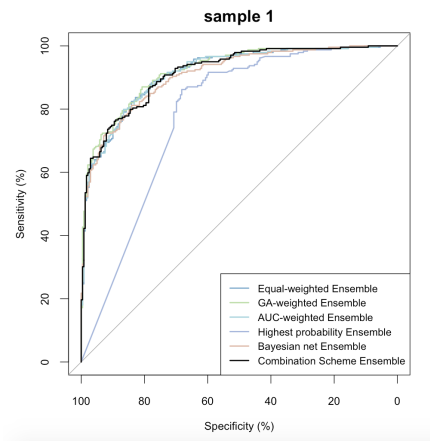


Figure 5.7.6: ROC plots of base models and GA-weighted Ensemble

nificance of difference between the combination schemes, except for the least performing ensemble. The test gave an F statistic of 1.841 with $\Pr(>F) = 0.138$, which did not yield an indication of a significant difference. This also support the observation made regarding the second level ensemble, proving the less diversity among the second level base models.



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

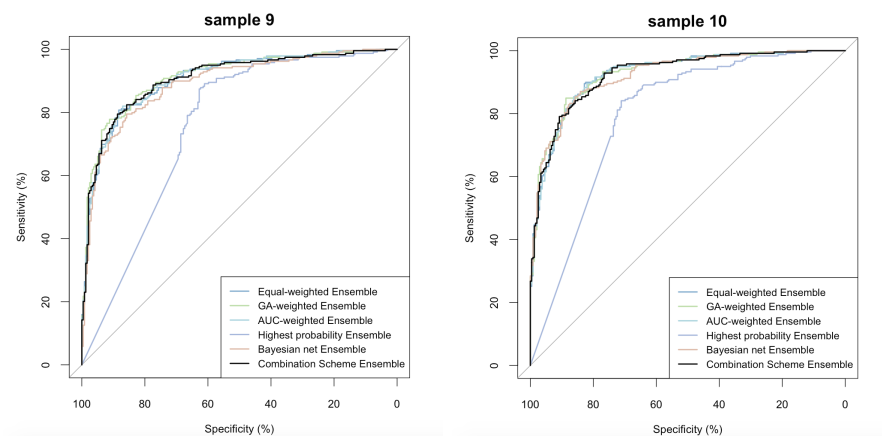


Figure 5.7.7: ROC plots of Ensemble models

Further, PR curve AUCs were obtained for all six ensemble classifiers. The average PR curve AUC values are presented in Table 5.8. The consistency of numerical order in AUC values between ROC and PR curves, partially solidifies the proof given by Davis et al. [114]. That is: a curve dominates in ROC space if and only if it dominates the PR space.

5.8 Identification of Disease Related Proteins

As mentioned in Chapter 1, this study evaluated nine disease related proteins that are identified to be involved in ‘mitochondrion organization’ biology process. They are COX10, SCO1, AFG3, AAC1, FUM1, MGM1, BCS1, SDH1 and CYC3. Figure 5.8.8 illustrates the disease related protein identification matrix, along with the number of samples in the experiment that lead to correctly identifying each disease related protein, by each base model. It is a heat map where

Model	Precision Recall AUC
AUC-weighted ensemble	92.46
Bayesian net ensemble	91.99
Combination scheme ensemble	92.6
Equal-weighted ensemble	92.39
GA-weighted ensemble	93.03
Highest probability ensemble	70.6

Table 5.8: PR curve AUC values of ensemble models

right-end of the color scale (i.e. green) indicates all 10 samples, while left-end (i.e. red) indicates 0 samples. Not all proteins are correctly identified as positive by all 10 samples. Also it should be noted that, a base model may not give any prediction to a protein. For instance, Physical interactions model does not give predictions for CYC3 and FUM1 in 6 samples and 9 samples, respectively. This is due to the absence of relevant data.

The matrix clearly shows a significant difference in identifying disease proteins by base models and their ensembles. For instance, three base models can strongly predict MGM1 to be negative, while four base models also give a considerable support towards the protein being negative. However, there are three other base models that can strongly suggest otherwise. The overall result can conclude that the ensemble models are able to strongly predict MGM1 as positive. This observation clearly depicts the effectiveness of a diverse ensemble in accurate classification.

Overall, ensemble models are significantly better at identifying all disease proteins, except the CYC3 protein. Although three base models (Genetic + physical interactions, Genetic interactions and Expressions 1) strongly suggest that it is positive, none of the ensembles could give a very strong prediction. Perhaps, the expert knowledge in Biology is required for a discussion regarding the reason behind this observation. It should also be noted that three disease related proteins (i.e. COX10, AFG3 and AAC1) are identified by every ensemble model in all 10 samples, suggesting a strong functional context formulation for them by the incorporated data types.

Figure 5.8.9 gives a bar chart of the average number of samples out of the 10 that recognizes each disease related protein. Accordingly, the GA-weighted ensemble and the AUC-weighted ensemble have the highest disease related protein identification rate of 9.6667, while the lowest rate of 5.2222 is held by the PAAC base model. The highest rate for a base model is given by the Expressions 1

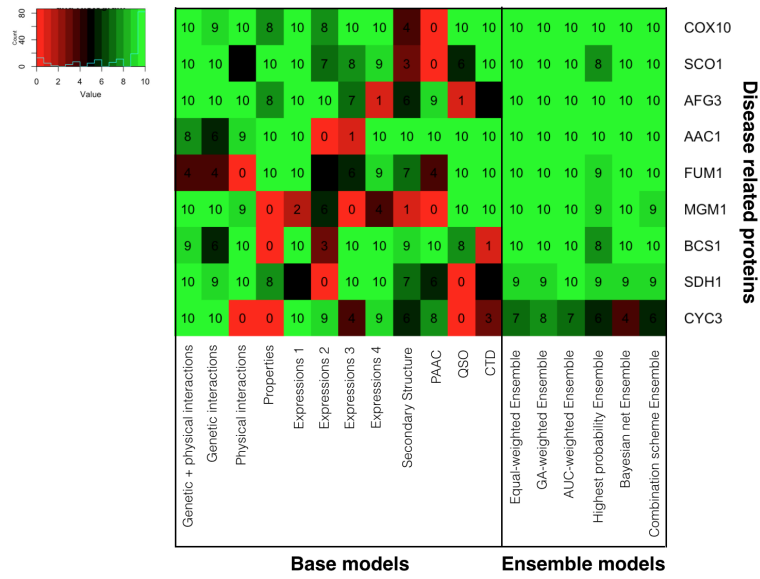


Figure 5.8.8: Disease protein identification matrix

model (i.e. 8.5556).

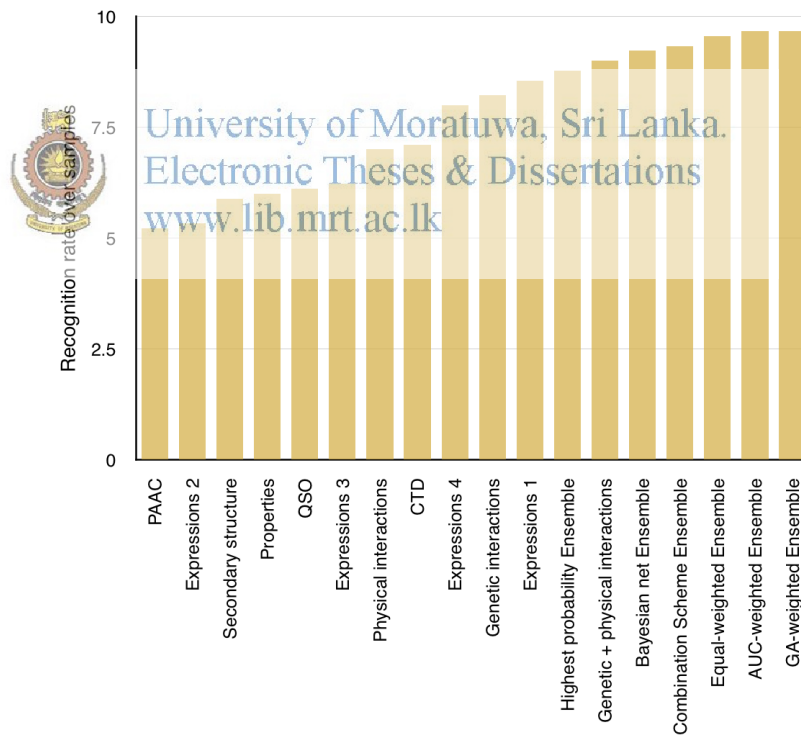


Figure 5.8.9: Disease related protein identification over the 10 samples

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

Protein Function Prediction is a supervised learning problem, focusing the annotation of a functionally unknown protein with its intended functions (i.e. biology processes, molecular functions). A key challenge is to constructively incorporate different biological data types that capture various functional aspects of the proteins, in order to formulate a strong functional context in classification. Hence, this study addressed the data heterogeneity in a single function prediction context, while eliminating the high class imbalance issue and the elusive nature of negative examples. The focus was to assess the effectiveness of a heterogeneous data ensemble approach for classifying *Saccharomyces cerevisiae* proteins under ‘mitochondrion organization’ biology process. Nine positive proteins are known to be human disease related and thus, it is important to annotate proteins under this particular Biology process.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Twelve base models were incorporated for the purpose, including nine euclidean-distance based nearest neighbour (NN) models and three affinity-based neighbourhood models. A reliable set of annotations were obtained from Gene Ontology, as well as from a publicly available benchmark gold dataset containing experimentally verified positive examples and negative examples. The class imbalance was addressed by preparing 10 annotation samples, each with a 1:1 positive to negative class ratio. Each base model was trained over a different biological data representation. The NN models contain 3 amino acid sequence models, 4 gene expression models, a peptide chain properties model and a secondary structure data model. Affinity-based models are the genetic interactions model; physical interactions model; and a combined genetic + physical interactions model. Five combination schemes were evaluated for fusing the base model outputs. The main purpose was to evaluate the standard Genetic Algorithm based weight scheme

with four other baseline combination schemes. Further, a second level ensemble was also evaluated, by taking the five different ensemble models as the base models. The results showed that a Genetic Algorithm based weight scheme is ideal for this classification purpose. Initially, the study also looked at representing a domain-wise protein sequence through an LDA topic model, and observed that it is ineffective in this context. Overall, the ensemble model substantially improves the prediction accuracy, due to the diverse set of heterogeneous data models. The approach has been able to give coverage to most of the important functional aspects, through the 12 different data representations. The kappa statistic depicts their potential to correct each others' mistakes at final prediction. More data types such as phylogeny profiles can be added in future for further performance enhancement. Moreover, the heterogeneous data ensemble is capable of identifying eight disease related *S. cerevisiae* proteins (i.e. COX10, AFG3, AAC1, SCO1, FUM1, MGM1, BCS1, SDH1) in a strong sense and one disease protein (i.e. CYC3) in a moderate sense. One aspect to look at in future, is the reason behind the moderate inaccuracy of ensemble models in identifying CYC3 as a disease related protein.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The four baseline combination schemes for an ensemble are the Equal-weighted scheme; AUC-weighted scheme; Highest probability scheme; and the Bayesian network based combination scheme. The GA-weighted Ensemble performance was compared to their performances, and the best combination scheme was observed to be the GA-weighted combination scheme, with an average ROC AUC of 92.52%. This gives a 17.32% improvement to the best performing base classifier (i.e. Genetic + physical interactions model). The GA-weighted approach is ideal for achieving an optimal classification of proteins under the targeted protein function class, while approximating the contributions of different biological data types. The most weighted base model reflects the importance of the temporal program of gene expression during meiosis and spore formation, when predicting proteins in the context of Mitochondrion biogenesis. However, we cannot completely rely on the average optimal weight vector for several reasons. Firstly,

we cannot guarantee to receive a global optimal weight vector from GA. More improvement may be gained by further experimental tuning of the GA hyperparameter settings. Moreover, modified versions of the standard Genetic Algorithm such as the Immune Genetic Algorithm can be tested for their effective usage in this context. Secondly, weights had been adjusted by the algorithm as to optimize the ROC AUC. These adjustments are affected by the base models that do not predict the functional class membership for some proteins, in the presence of missing data. Consequently, a mismatch was observed between the base model rankings in terms of the GA weights and mean AUCs. Since both are independent indicators of the biological importance of each data type, the mismatch could be justified. It can be concluded that, the optimized GA weights are more evidential of each others' tendency to contribute towards the biology process, while AUC provides each data types' individual capability to form the functional context. The complex interrelationships between these different data types is yet to be understood by Biologists.



University of Moratuwa, Sri Lanka.

GA-Weighted Scheme & Observations

www.lib.mrt.ac.lk

Even though the GA-weighted Scheme was observed to be the best ensemble classifier, the other combination schemes except for the highest probability ensemble, were also able to give high accuracy rates, implying no significant difference among their performances. The second-level combination scheme ensemble did not yield an improvement, due to the low diversity among the five ensemble models. It is also evidential from the result obtained through a one-way ANOVA test over the ensemble models, except for the highest-probability ensemble which did not improve the best base classifier. There is no significant difference between their average ROC AUC values over the 10 samples. Hence, it can be concluded that in general, more diverse combination schemes will be required for implementing a successful second-level ensemble.


One obstacle for this study was the lack of a benchmark study to compare the results with. There is a wide range of protein function prediction studies in literature. However they use diverse datasets, preprocessing methods and approaches,

making it somewhat difficult to carry-out a thorough comparison. Hence the primary focus of this study was to solely assess the heterogeneous data ensemble performance, along with the comparison of different combination schemes. A very abstract and generic level comparison can be made with model accuracies presented by Hibbs et al. [12]. They have evaluated an ensemble of three diverse computational methods for predicting genes/proteins that involve in ‘mitochondrion organization’. The overall prediction accuracy has been observed to be 67.21% (123/183). In comparison, this study observed an 86% mean accuracy. However, this cannot be taken as a good comparison, since the datasets and experimental setting are not the same.

This heterogeneous data ensemble approach however did not address the multi-class, hierarchical classification. In future, the GA-weighted heterogeneous data ensemble can be extended to a hierarchically consistent classification ensemble as well. Initially, the method can be tested upon the sub hierarchy, which includes offspring Biology Process GO terms of ‘mitochondrion organization’. The method can also be tested upon other GO terms with appropriate strategies for hierarchical multi-label classification. However, the performance could vary, as different functional contexts require different data utilizations. Further, data types can be more refined at selection. For instance, more domain knowledge can be consulted to select relevant microarray expression datasets. For interaction data, more network analysis can be incorporated to enhance the capability of achieving a neighbourhood that reflect the true functional context. Another important data type to be considered is the tertiary or quaternary protein structure. Utilizing 3D structural data for the purpose would indeed give more clues about the functional context, with respect to structural exposure, molecular dynamics, protein energy states and stability.

In conclusion, this ensemble based heterogeneous data mining approach enables an accurate classification of *Saccharomyces cerevisiae* proteins under ‘mitochondrion organization’ Biology Process in Gene Ontology.


References

- [1] P. Bork, T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan, "Predicting function: from genes to genomes and back", *Journal of molecular biology*, vol. 283, no. 4, pp. 707-725, 1998.
- [2] D. Botstein and G. R. Fink. "Yeast: an experimental organism for 21st Century biology", *Genetics*, vol. 189, no. 3, pp. 695-704, 2011.
- [3] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, ... and M.A. Harris, "Gene Ontology: tool for the unification of biology", *Nature genetics*, vol. 25, no. 1, pp. 25-29, 2000.
- [4] H.W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S Rudd and B. Weil, "MIPS: a database for genomes and protein sequences", *Nucleic acids research*, vol. 30, no. 1, pp. 31-34, 2002.
- [5] HER2 Status [Online]. Available:  www.lib.mrt.ac.lk
<http://www.breastcancer.org/symptoms/diagnosis/her2>
- [6] What is sickle cell disease [Online]. Available - <http://www.nhlbi.nih.gov/health/health-topics/topics/sca>
- [7] What are yeast [Online]. Available: http://wiki.yeastgenome.org/index.php/What_are_yeast%3F
- [8] A. Barrientos, "Yeast Models of Human Mitochondrial Diseases", *IUBMB Life*, vol. 55, no. 2, pp. 83-95, 2008.
- [9] By Masur-Own work, Public Domain [Online]. Available: <https://commons.wikimedia.org/w/index.php?curid=1069017>
- [10] By domdomegg-Own work, CC BY 4.0 [Online]. Available: <https://commons.wikimedia.org/w/index.php?curid=46468746>

- [11] M.R. Duchon and G. Szabadkai, "Roles of mitochondria in human disease", *Essays in biochemistry*, no. 47, pp. 115-137, 2010.
- [12] M.A. Hibbs, C.L. Myers, C. Huttenhower, D.C. Hess, K. Li, A.A. Caudy and O.G. Troyanskaya, "Directing experimental biology: a case study in mitochondrial biogenesis", *PLoS Comput Biol*, vol. 5, no. 3, e1000322, 2009.
- [13] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, "Analyzing protein structure and function", *Molecular Biology of the cell* (4th ed.), New York: Garland Science, 2002.
- [14] P. Radivojac, W.T. Clark, T.R. Oronn, A.M. Schnoess, T. Wittkop, A. Sokolov, K. Graim et al., "A large-scale evaluation of computational protein function prediction", *Nature methods*, vol. 10, no. 3, pp. 221-227, 2013.
- [15] G. Valentini, "Hierarchical ensemble methods for protein function prediction", *ISRN bioinformatics*, 2014.
- [16] R. Nielsen, J.S. Patil, A. Albrechtsen and Y.S. Song, "Genotype and SNP calling from next-generation sequencing data", *Nature Reviews Genetics*, vol. 12, no. 6, pp. 443-451, 2011.
- [17] J. Quackenbush, "Microarray data normalization and transformation", *Nature genetics*, vol. 32, pp. 496-501, 2002.
- [18] A. Sanchez and M.C.R. de Villa, "A Tutorial Review of Microarray Data Analysis", Universitat de Barcelona, 2008.
- [19] Bioinformatics: Introduction and Methods, Center for Bioinformatics, Peking University [Online Lecture]. Available: <https://www.coursera.org/learn/bioinformatics-pku>
- [20] T. Can, Introduction to Bioinformatics, Middle East Technical University [Online]. Available: <http://ocw.metu.edu.tr/course/view.php?id=37>

- [21] What is Functional Genomics [Online]. Available:
<http://www.ebi.ac.uk/training/online/course/functional-genomics-introduction-embl-ebi-resource/what-functional-genomics-1>
- [22] L. R. Engelking, "Textbook of Veterinary Physiological Chemistry", Chapter 1 [Online]. Available:
http://booksite.elsevier.com/samplechapters/9780123848529/02~Chapter_1.pdf
- [23] General structure of an amino acid [Online]. Available:
<https://en.wikipedia.org/wiki/File:AminoAcidball.svg>
- [24] Peptide bond [Online]. Available:
<http://www.ifa.hawaii.edu/UHNAI/article4.htm>
- [25] Amino Acid Properties [Online]. Available:
http://www.genscript.com/amino_acid_structure.html
- [26] Protein Structure [Online]. Available:
<http://courses.washington.edu/conf/protein/protein.htm>
- [27] Q. Gu, Y.S. Ding and B. Zhang "An ensemble classifier based prediction of G-protein-coupled receptor classes in low homology", *Neurocomputing*, vol. 154, pp. 110-118, 2015.
- [28] Protein Folds and Protein Fold Classification [Online]. Available:
<http://www.proteinstructures.com/Structure/Structure/protein-fold.html>
- [29] Supersecondary Structures (Motifs) and Domains [Online]. Available:
<https://biochemistryquestions.wordpress.com/2008/10/09/supersecondary-structures-motifs-and-domains/>
- [30] Protein Domains and Domain Classification [Online]. Available:
<http://www.proteinstructures.com/Structure/Structure/protein-domains.html>

- [31] Protein Three-Dimensional Structure: Structural Levels, Motifs and Folds [Online]. Available:
<http://www.proteinstructures.com/Structure/protein-structure1.html>
- [32] G. Pandey, V. Kumar and M. Steinbach, "Computational approaches for protein function prediction: A survey", Twin Cities::Department of Computer Science and Engineering, University of Minnesota, 2006.
- [33] What are protein families [Online]. Available:
<https://www.ebi.ac.uk/training/online/course/introduction-protein-classification-ebi/protein-classification/what-are-protein-families>
- [34] J. Adams, "The proteome: discovering the structure and function of proteins", *Nature Education*, vol. 1, no. 3, pp 6, 2008.
- [35] S. Halgamuge, Advanced workshop in Bioinformatics.
- [36] S.Y. Rhee and M. Mutwil, "Towards revealing the functions of all genes in plants", *Trends in plant science*, vol. 19, no. 4, pp.212-221, 2014.
- [37] M.M. Babu, "Introduction to microarray data analysis", *Computational genomics: Theory and application*, pp.225-249, 2004.
- [38] Y.H. Yang, S. Dudoit, P. Luu, D.M. Lin, V. Peng, J. Ngai and T.P. Speed, "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation", *Nucleic acids research*, vol. 30, no. 4, e15-e15, 2002.
- [39] B. Zhang, Gene Expression Data Analysis, Department of Biomedical Informatics, Vanderbilt University [Online]. Available:
http://bioinfo.vanderbilt.edu/zhanglab/lectures/BMIF310_geneExpression-_B_dataAnalysis_2009.pdf

- [40] Z. Louxin, Gene expression data analysis, Department of Mathematics, National University of Singapore [Online]. Available:
<http://www.bii.a-star.edu.sg/docs/education/lsm5192/zhangLec7.pdf>
- [41] D. Nettleton, Normalization Methods for Two-Color Microarray Data, Iowa State University [Online]. Available:
<http://www.public.iastate.edu/~dnett/microarray/05normalization2color.ppt>
- [42] Normalization of Microarray Data [Online]. Available:
<https://www.youtube.com/watch?v=FpuVrfMu-2U>
- [43] P. Hoen, Expression Array Normalization [Online]. Available:
http://dial.liacs.nl/Courses/MicroArrayDataAnalysis/Lectures/normalization_R-course_PB.pdf
- [44] GO Evidence codes [Online]. Available:
<http://geneontology.org/page/guide-go-evidence-codes>
- [45] QuickGO, A fast browser for Gene Ontology terms and annotations [Online]. Available:

<http://www.ebi.ac.uk/QuickGO>
www.lib.mrt.ac.lk
- [46] R. Eisner, B. Poulin, D. Szafron, P. Lu and R. Greiner, "Improving protein function prediction using the hierarchical structure of the gene ontology", *In Computational Intelligence in Bioinformatics and Computational Biology*, Proceedings of the IEEE Symposium, pp. 1-10, 2005.
- [47] G. Valentini, "True path rule hierarchical ensembles for genome-wide gene function prediction", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 3, pp.832-847, 2011.
- [48] G. Valentini, "True path rule hierarchical ensembles", In International Workshop on Multiple Classifier Systems, pp. 232-241, Springer Berlin Heidelberg, 2009.

- [49] M.N. Davies, A. Secker, A.A. Freitas, M. Mendao, J. Timmis and D.R. Flower, "On the hierarchical classification of G protein-coupled receptors", *Bioinformatics*, vol. 23, no. 23, pp. 3113-3118, 2007.
- [50] J. Struyf, S. Dzeroski, H. Blockeel and A. Clare, "Hierarchical multi-classification with predictive clustering trees in functional genomics". pp. 272-283. Springer Berlin Heidelberg, 2005
- [51] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev and S. Dzeroski, "Predicting gene function using hierarchical multi-label decision tree ensembles". *BMC bioinformatics*, vol. 11, no. 1, pp. 1, 2010.
- [52] Z. Barutcuoglu, R.E. Schapire and O.G. Troyanskaya, "Hierarchical multi-label prediction of gene function", *Bioinformatics*, vol. 22, no. 7, pp. 830-836, 2006.
- [53] N. Alaydie, C.K. Reddy and F. Fotouhi, "Hierarchical boosting for gene function prediction", *In Proceedings of the 9th International Conference on Computational Systems Bioinformatics*, vol. 9, pp. 14-25, 2010.
- [54] G. Pandey, C.L. Myers and V. Kumar, "Incorporating functional inter-relationships into protein function prediction algorithms". *BMC bioinformatics*, vol. 10, no. 1, pp. 1, 2009.
- [55] Y. Guan, C.L. Myers, D.C. Hess, Z. Barutcuoglu, A. Caudy and O. Troyanskaya, "Predicting gene function in a hierarchical context with an ensemble of classifiers", *Genome biology*, vol. 9, no. 1, S3, 2008.
- [56] N. Alaydie, C.K. Reddy and F. Fotouhi, "A Bayesian integration model for improved gene functional inference from heterogeneous data sources". *In Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pp. 376-380, 2011.
- [57] Q. Wu, Y. Ye, S.S. Ho and S. Zhou, "Semi-supervised multi-label collective classification ensemble for functional genomics". *BMC genomics*, vol. 15, no. 9, S17, 2014.

- [58] G. Yu, C. Domeniconi, H., Rangwala, G. Zhang and Z. Yu, "Transductive multi-label ensemble classification for protein function prediction". *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1077-1085. ACM, 2012.
- [59] L. Lan, N. Djuric, Y. Guo and S. Vucetic, "MS-kNN: protein function prediction by integrating multiple data sources", *BMC bioinformatics*, vol. 14, no. 3, p.S8, 2013.
- [60] S. Mostafavi, D. Ray, D. Warde-Farley., C. Grouios and Q. Morris, "GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function". *Genome Biol*, vol. 9, no. 1, S4, 2008.
- [61] X.M. Zhao, Y. Wang, L. Chen and K. Aihara, "Gene function prediction using labeled and unlabeled data", *BMC bioinformatics*, vol. 9, no. 1, pp.1, 2008.
- [62] N. Youngs, D. Penfold-Brown, R. Bonneau and D. Shasha, "Negative example selection for protein function prediction: the NoGO database", *PLoS Computational Biology*, vol. 10, no. 6, p.e1003644, 2014.
- [63] N. Youngs, D. Penfold-Brown, K. Drew, D. Shasha and R. Bonneau, "Parametric Bayesian priors and better choice of negative examples improve protein function prediction", *Bioinformatics*, p.btt110.
- [64] G. Obozinski, G. Lanckriet, C. Grant, M.I. Jordan and W.S. Noble, "Consistent probabilistic outputs for protein function prediction", *Genome Biology*, vol. 9, no. 1, pp.1, 2008.
- [65] J.F. Diez-Pastor, J.J. Rodriguez, C. Garcia-Osorio and L.I. Kuncheva, "Random Balance: Ensembles of variable priors classifiers for imbalanced data", *Knowledge-Based Systems*, vol. 85, pp. 96-111, 2015.
- [66] P. Yang, W. Liu, B.B. Zhou, S. Chawla and A.Y. Zomaya, "Ensemble-based wrapper methods for feature selection and class imbalance learning",

In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 544-555, Springer Berlin Heidelberg, 2013.

- [67] M. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H. and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp.463-484, 2012.
- [68] GO Annotations Download [Online]. Available:
<http://geneontology.org/page/download-annotations>
- [69] C. Huttenhower, M.A. Hibbs C.L. Myers, A.A. Caudy, D.C. Hess and O.G. Troyanskaya, "The impact of incomplete knowledge on evaluation: an experimental benchmark for protein function prediction", *Bioinformatics*, vol. 25, no. 18, pp. 2404-2410, 2009.
- [70] D. Charif and J.R. Lobry, "SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis" *Structural approaches to sequence evolution*, pp. 207-232, Springer Berlin Heidelberg, 2007.
- [71] R. Apweiler, T.K. Attwood, A. Bairoch, A. Bateman, E. Birney, M. Biswas, P. Bucher, L. Cerutti, F. Corpet, M.D.R. Croning and R. Durbin, "The InterPro database, an integrated documentation resource for protein families, domains and functional sites", *Nucleic acids research*, vol. 29, no. 1, pp. 37-40, 2001.
- [72] S. Mnaimneh, A.P. Davierwala, J. Haynes, J. Moffat, W.T. Peng, W. Zhang, X. Yang, J. Pootoolal, G. Chua, A. Lopez and M. Trochesset, "Exploration of essential gene functions via titratable promoter alleles", *Cell*, vol. 118, no. 1, pp. 31-44, 2004.

- [73] S. Chu, J. DeRisi, J. M. Eisen, J. Mulholland, D. Botstein, P.O. Brown and I. Herskowitz, "The transcriptional program of sporulation in budding yeast", *Science*, vol. 282, no. 5389, pp. 699-705, 1998.
- [74] A.P. Gasch, M. Huang, S. Metzner, D. Botstein, S.J. Elledge and P.O. Brown, "Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p", *Molecular biology of the cell*, vol. 12, no. 10, pp. 2987-3003, 2001.
- [75] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein and B. Futcher, "Comprehensive identification of cell cycle regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization", *Molecular biology of the cell*, vol. 9, no. 12, pp. 3273-3297, 1998.
- [76] G.K. Smyth, N. P. Thorne and J. Wettenhall, "LIMMA: Linear Models for Microarray Data Users Guide", 2003. [Online].
Available: <http://www.bioconductor.org>
- [77] K.V. Ballman, D.E. Ghim, A.L. Oberg and T.M. Therneau, "Faster cyclic loess: normalizing RNA arrays via linear models", *Bioinformatics* vol. 20, no. 16, pp. 2778-2786, 2004.
- [78] H. Bengtsson, K. Simpson, J. Bullard and K. Hansen, "aroma.affymetrix: A generic framework in R for analyzing small to very large Affymetrix data sets in bounded memory", Tech Report, no 745, Department of Statistics, University of California, Berkeley, February 2008
- [79] T. Hastie, R. Tibshirani, B. Narasimhan and G. Chu. "impute: impute: Imputation for microarray data", 2011.
- [80] K.C. Chou, "Prediction of protein cellular attributes using psuedo-amino acid composition." *Proteins: Structure, Function, and Bioinformatics*, vol. 43, no. 3, pp. 246-255, 2001.

- [81] N. Xiao, Q. Xu, and D. Cao. "protr: Protein Sequence Feature Extraction with R", R package version 0.2-0, 2013 [Online]. Available: <http://CRAN.R-project.org/package=protr>.
- [82] B. Petersen, T.N. Petersen, P. Andersen, M. Nielsen and C. Lundegaard, "A generic method for assignment of reliability scores applied to solvent accessibility predictions", *BMC structural biology*, vol. 9, no.1, pp. 1, 2009.
- [83] T.N. Petersen, C. Lundegaard, M. Nielsen, H. Bohr, J. Bohr, Brunak, S., ... and O. Lund, "Prediction of protein secondary structure at 80% accuracy", *Proteins: Structure, Function, and Bioinformatics*, vol. 41, no. 1, pp. 17-20, 2000.
- [84] G. Csardi and T. Nepusz, "The igraph software package for complex network research", *InterJournal, Complex Systems*, vol. 1695, no. 5, pp.1-9, 2006.
- [85] L. Nanni, S. Mazzara, L. Pattini and A. Lumini, "Protein classification combining surface analysis and primary structure", *Protein Engineering Design and Selection*, vol. 22, no. 4, pp. 267-272, 2009.
- [86] K.C. Chou, "Prediction of protein subcellular locations by incorporating quasi-sequence-order effect", *Biochemical and biophysical research communications*, vol. 278, no. 2, pp. 477-483, 2000.
- [87] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li and H. Jiang, "Predicting protein-protein interactions based only on sequences information", *Proceedings of the National Academy of Sciences*, vol. 104, no. 11, pp. 4337-4341, 2007.
- [88] Y. Yang, "Identification of novel type III effectors using latent Dirichlet allocation", *Computational and mathematical methods in medicine* 2012.
- [89] D.M. Blei, A. Y. Ng and M.I. Jordan, "Latent dirichlet allocation", *Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003.

- [90] M. La Rosa, A. Fiannaca, R. Rizzo and A. Urso, "Probabilistic topic modeling for the analysis and classification of genomic sequences", *BMC bioinformatics*, vol. 16, no. 6, pp. 1, 2015.
- [91] D.M. Blei, K. Franks, M.I. Jordan and I.S. Mian, "Statistical modeling of biomedical corpora: mining the Caenorhabditis Genetic Center Bibliography for genes related to life span", *BMC Bioinformatics*, vol. 7, no. 1, pp. 250, 2006.
- [92] S.G.A. Konietzny, L. Dietz, and A. C. McHardy. "Inferring functional modules of protein families with probabilistic topic models", *BMC bioinformatics*, vol. 12, no. 1, pp. 1, 2011.
- [93] X.Y. Pan, Y.N. Zhang and H.B. Shen, "Large-Scale prediction of human protein protein interactions from amino acid sequence based on latent topic features", *Journal of Proteome Research*, vol. 9, no. 10, pp. 4992-5001, 2010.
- [94] Y. Yang, B.L. Lu and W.Y. Yang, "Classification of Protein Sequences Based on Word Segmentation Methods." In APBC, pp. 177-186. 2008.
- [95] G. Valentini and F. Masulli, "Ensembles of learning machines", *Neural Nets*, Springer Berlin Heidelberg, pp. 3-20, 2002.
- [96] T.G. Dietterich, "Ensemble methods in machine learning", In International workshop on multiple classifier systems, Springer Berlin Heidelberg, pp. 1-15. 2000.
- [97] J. Stefanowski, "Multiple Classifiers", Institute of Computing Sciences, Poznań University of Technology [Online]. Available:
<http://www.cs.put.poznan.pl/jstefanowski/aed/DMmultipleclassifiers.pdf>
- [98] P. Yang, Y. Hwa Yang, B.B. Zhou and A.Y Zomaya, "A review of ensemble methods in bioinformatics", *Current Bioinformatics*, vol. 5, no. 4, pp. 296-308, 2010.

- [99] N.C. Oza, "Ensemble data mining methods", NASA Ames Research Center, USA, 2004.
- [100] Hal Daume III , "Ensemble methods", chapter 11, A Course in Machine Learning [Online]. Available:
http://ciml.info/dl/v0_9/ciml-v0_9-ch11.pdf
- [101] L. Rokach, "Ensemble methods for classifiers", *Data Mining and Knowledge Discovery Handbook*, pp. 957-980, Springer US, 2005.
- [102] S. Whalen and G.K. Pandey, "A comparative analysis of ensemble classifiers: case studies in genomics", In IEEE 13th International Conference on Data Mining, pp. 807-816, 2013.
- [103] T.G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes", *Journal of artificial intelligence research*, pp. 263-286, 1995.
- [104] E.B. Kong and T.G. Dietterich, "Error-Correcting Output Coding Corrects Bias and Variance" In ICML, pp. 315-321, 1995.
- [105] V.I. Nazarov, M.V. Pogorelyy, E.A. Komech, I.V. Zvyagin, D.A. Bolotin, M. Shugay, D.M. Chudakov, Y.B. Lebedev and I.Z. Mamedov, "tcR: an R package for T cell receptor repertoire advanced data analysis", *BMC bioinformatics*, vol. 16, no. 1, pp.1, 2015.
- [106] K. Hornik, B. Grun, "topicmodels: An R package for fitting topic models", *Journal of Statistical Software*, vol. 40, no. 13, pp. 1-30, 2011.
- [107] J. Holland, "Genetic algorithms", 1992.
- [108] L. Kuncheva, "Genetic algorithm for feature selection for parallel classifiers", *Information Processing Letters*, vol. 46, no. 4, pp. 163-168, 1993.
- [109] L. Scrucca, "GA: A Package for Genetic Algorithms in R", *Journal of Statistical Software*, vol. 53, no. 4, pp. 1-37, 2013.

- [110] M. Scutari, "bnlearn: Bayesian network structure learning", R package, 2010.
- [111] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers", *Machine learning*, vol. 31, no. 1, pp.1-38, 2004.
- [112] A.J. Viera and J.M. Garrett, "Understanding interobserver agreement: the kappa statistic", *Fam Med*, vol. 37, no. 5, pp. 360-363, 2005.
- [113] J.L. Fleiss, "Measuring nominal scale agreement among many raters", *Psychological bulletin*, vol. 76, no. 5, pp. 378, 1971.
- [114] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves", *In Proceedings of the 23rd international conference on Machine learning*, pp. 233-240, 2006.
- [115] M Gamer, J Lemon, I Fellows and P Singh, "irr: Various Coefficients of Interrater Reliability and Agreement", R package, 2012.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Appendix A

Exploratory Data Analysis

A.1 Initial Analysis of GO Annotations

There are 98,957 protein annotations with 17 variables, for 6381 distinct proteins from SGD database. 5530 distinct GO IDs have been used for annotation. 32,015 annotations were found to be duplicates (32.35% out of all). 232 annotations are explicitly noted as not being associated with the GO term. Under 18 different evidence codes: 25,412 annotations are with experimental evidence code; 5628 are with computational analysis evidence code; 481 are with author statement evidence code; 4838 are with curatorial statement evidence code; and 30,583 annotations are with IEA (Inferred from Electronic Annotation). There are 26,140 Biological Process GO term annotations; 19,129 Molecular Function GO term annotations; and 21,673 Cellular Component GO term annotations, in total. The annotations have been assigned by 8 parties: CACAO, GO Central, GOC, HGNC, InterPro, MGI, SGD and UniProt. Out of them, SGD stands for most of the annotations. All annotations belong to same database object type: 'gene', and the same taxon: 'taxon:559292', which indicates *S. cerevisiae*. Annotations have been made during the time period of 2000 - 2015. Most of the annotations have been made in 2015. All 30,583 IEA annotations have been made in 2015. However, the amount of curated annotations (36,359) surpasses the amount of electronically inferred annotations.

A.2 Data Visualizations

Following data visualizations were obtained using R graphic packages: *ggplot2* and *RColorBrewer*.

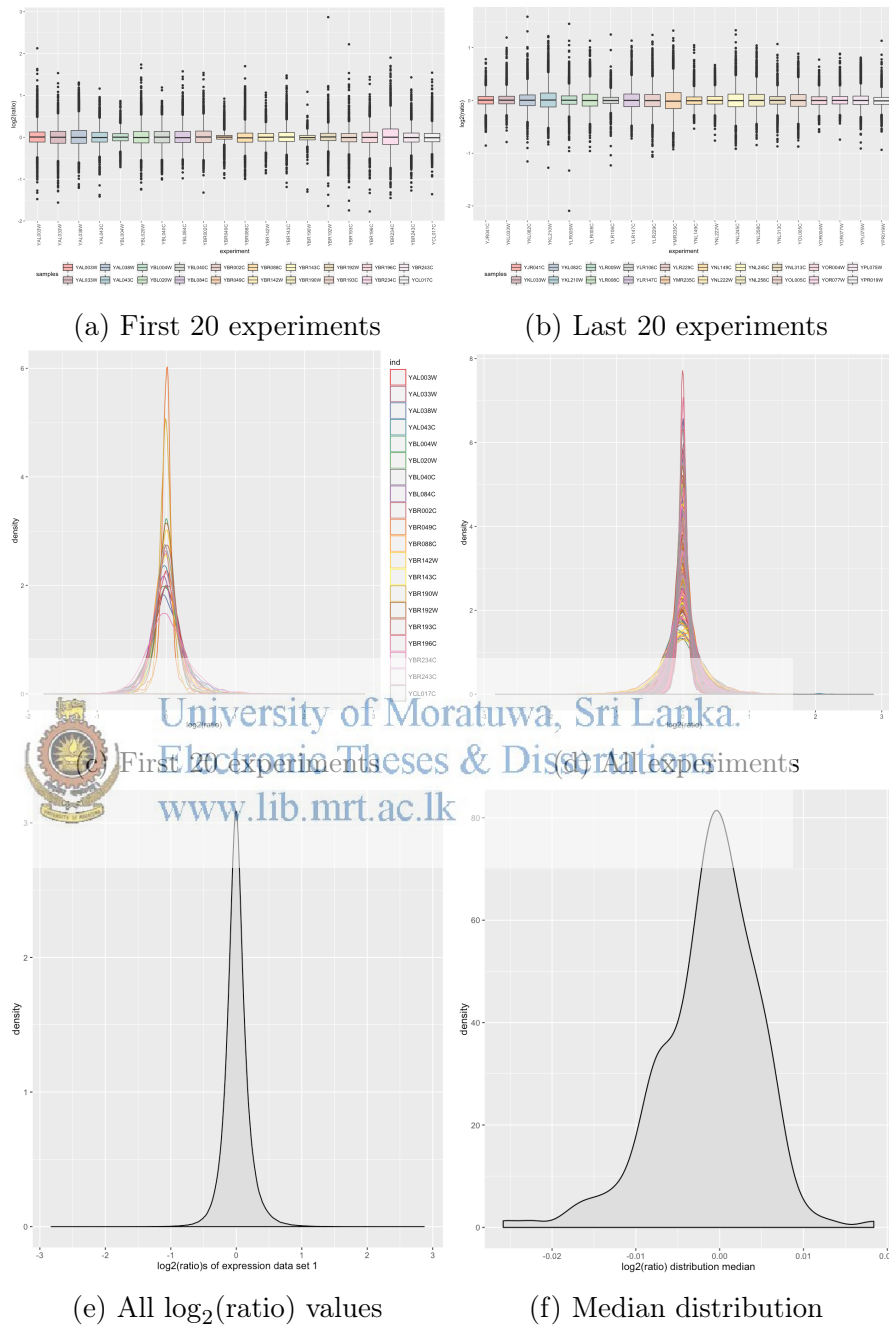
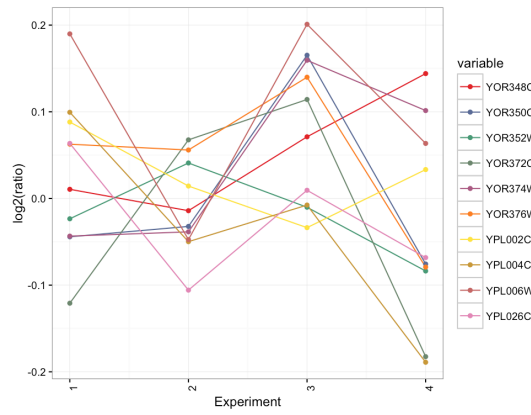
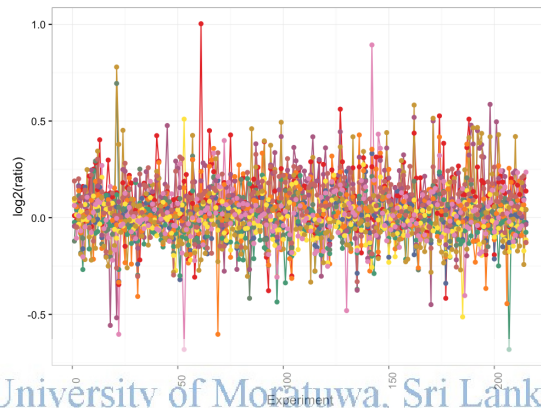


Figure A.2.1: Expressions 1 - Before normalization/preprocessing



(a) Expression ratio profiles of first 10 genes for the first 4 experiments



University of Moratuwa, Sri Lanka.
 E-Expression Theses & Dissertations
 for all experiments
www.lib.mru.ac.lk

Figure A.2.2: Expressions 1 - After normalization/preprocessing

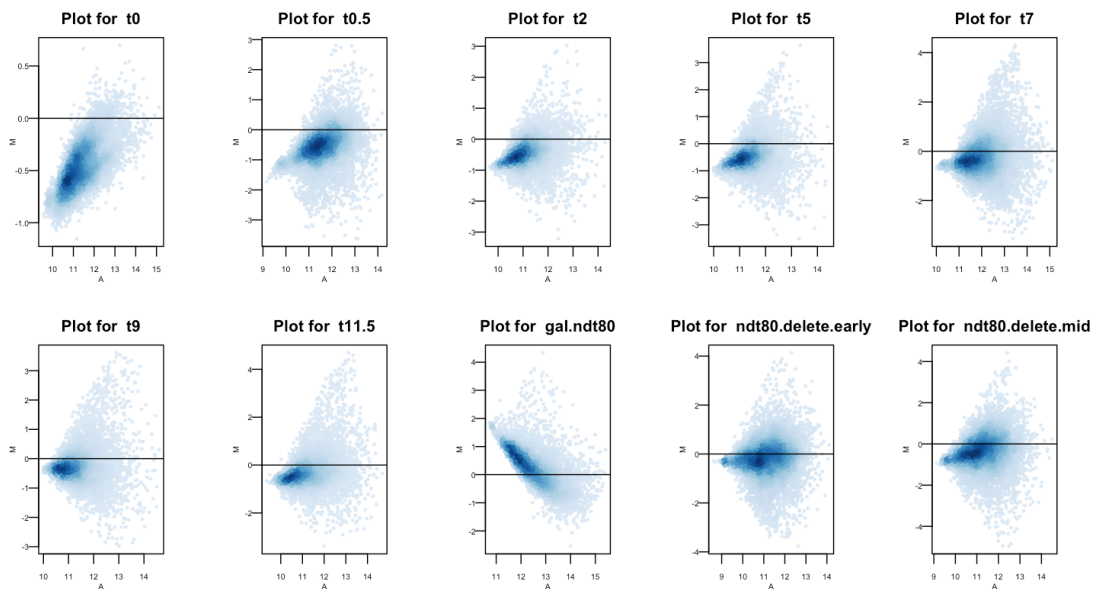


Figure A.2.3: Expressions 2 - MA plots before background correction

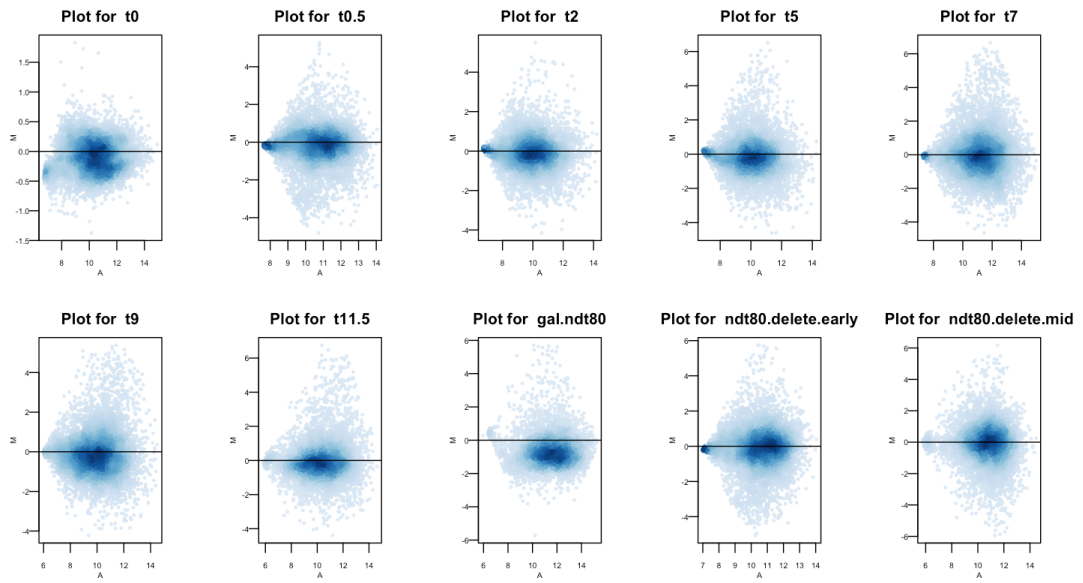


Figure A.2.4: Expressions 2 - MA plots after background correction

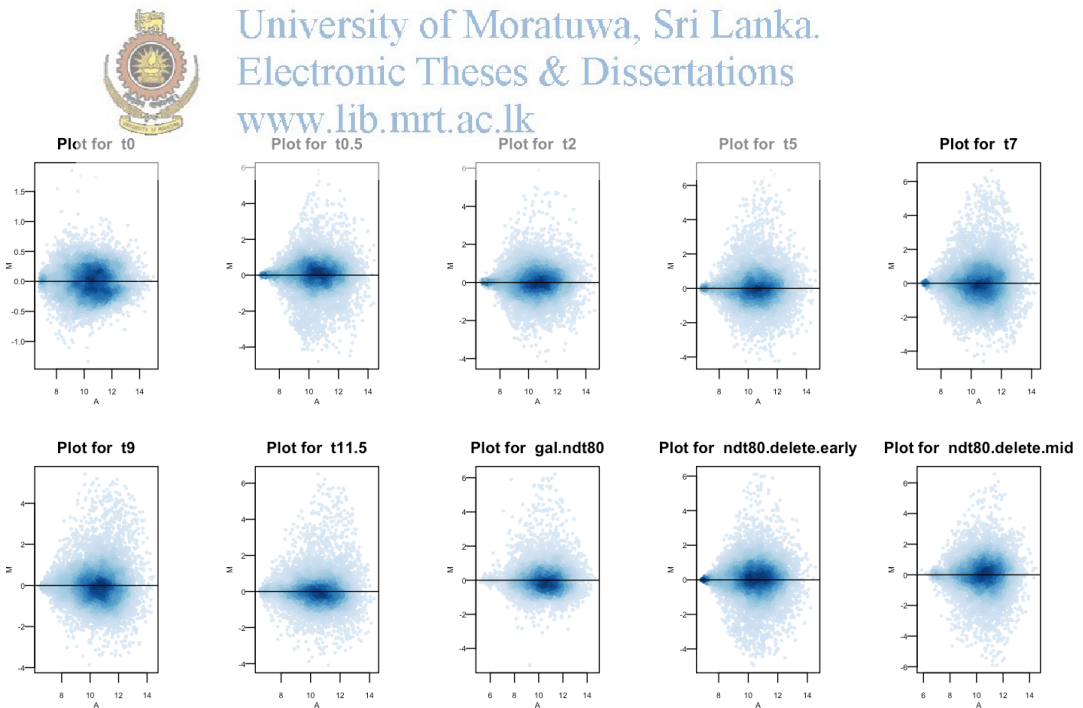
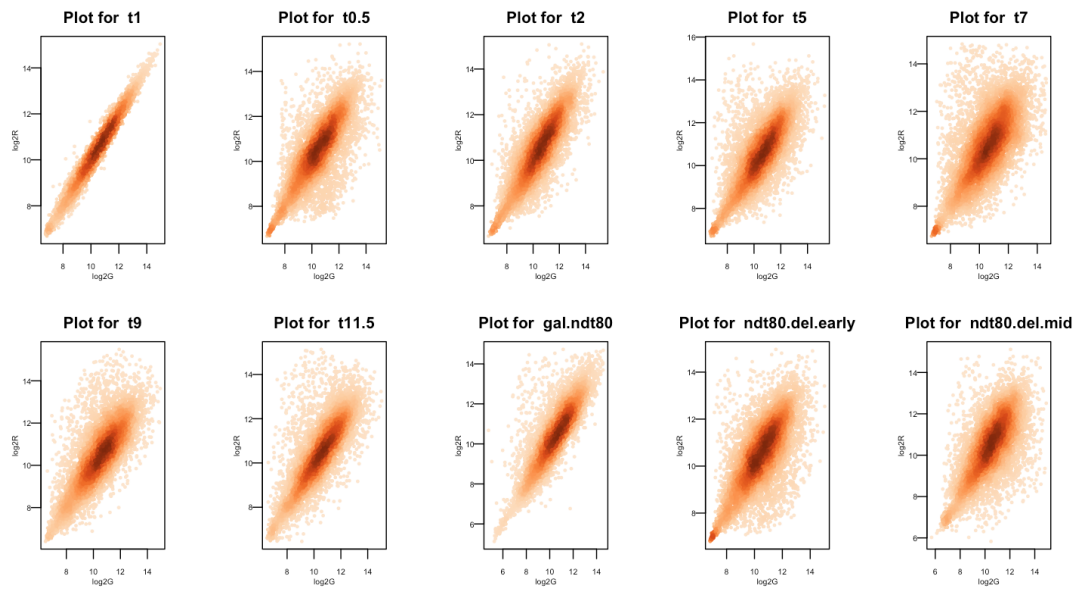


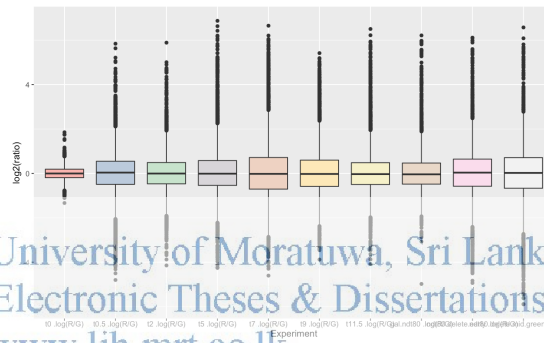
Figure A.2.5: Expressions 2 - MA plots after within/between array normalization



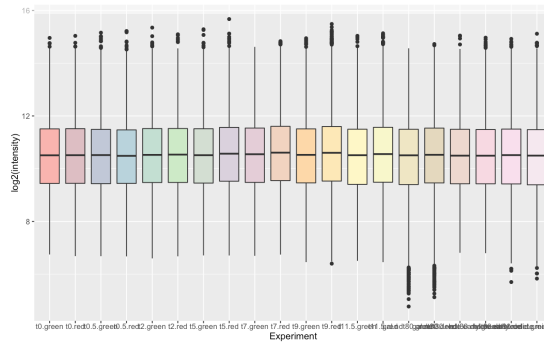
(a) Expressions 2 - $\log_2(G)$ vs $\log_2(R)$



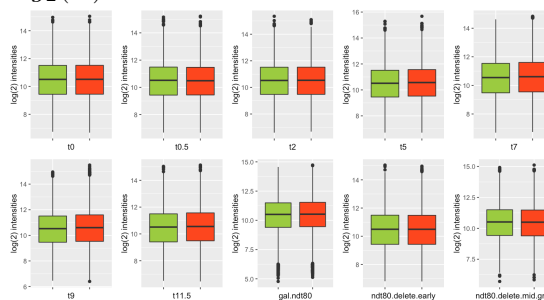
University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk



(b) Side-by-side boxplots of log ratios

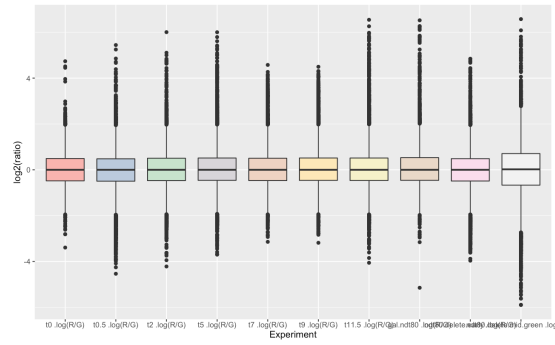


(c) Side-by-side boxplots of $\log_2(R)$ and $\log_2(G)$

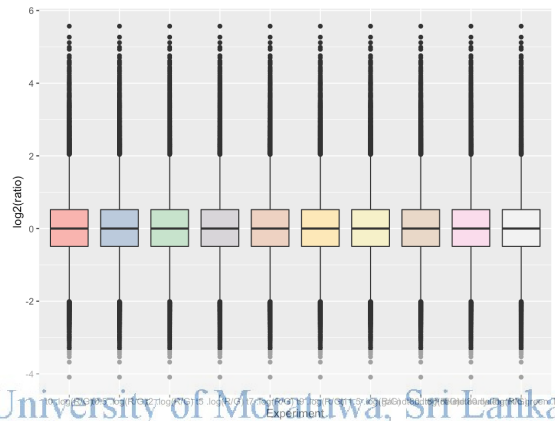


(d) Side-by-side pair boxplots of $\log_2(R)$ and $\log_2(G)$

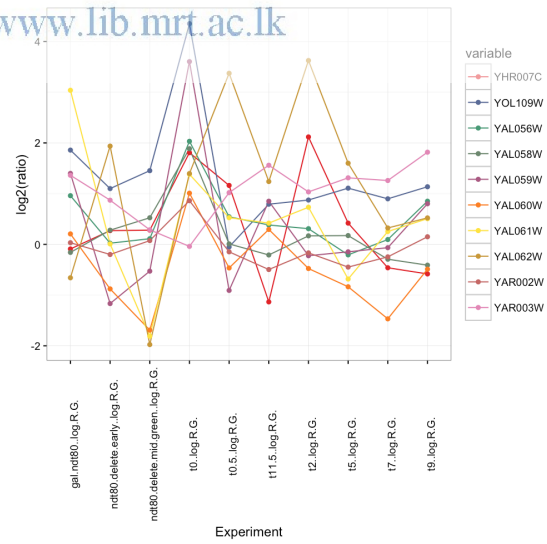
Figure A.2.6: Expressions 2 - After normalization/preprocessing



(a) After median centering and scale normalization

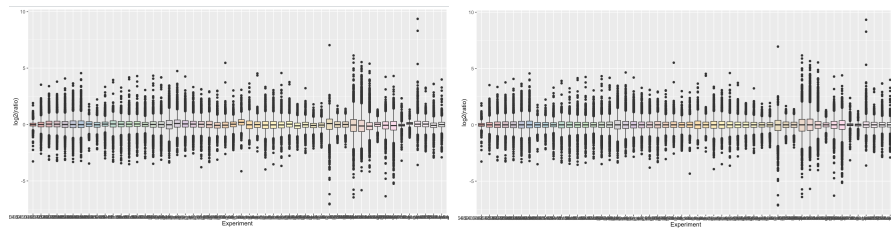


University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk



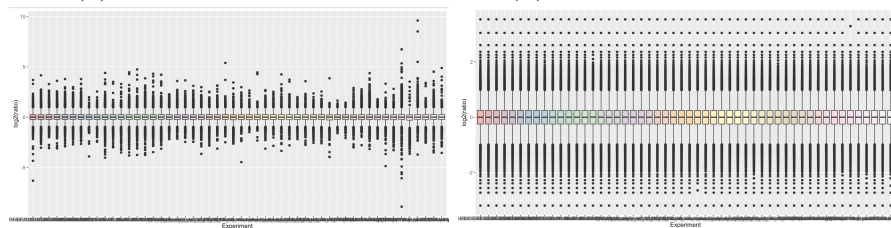
(c) Final Gene expression ratio profiles of first 10 genes

Figure A.2.7: Expressions 2 - After further normalization



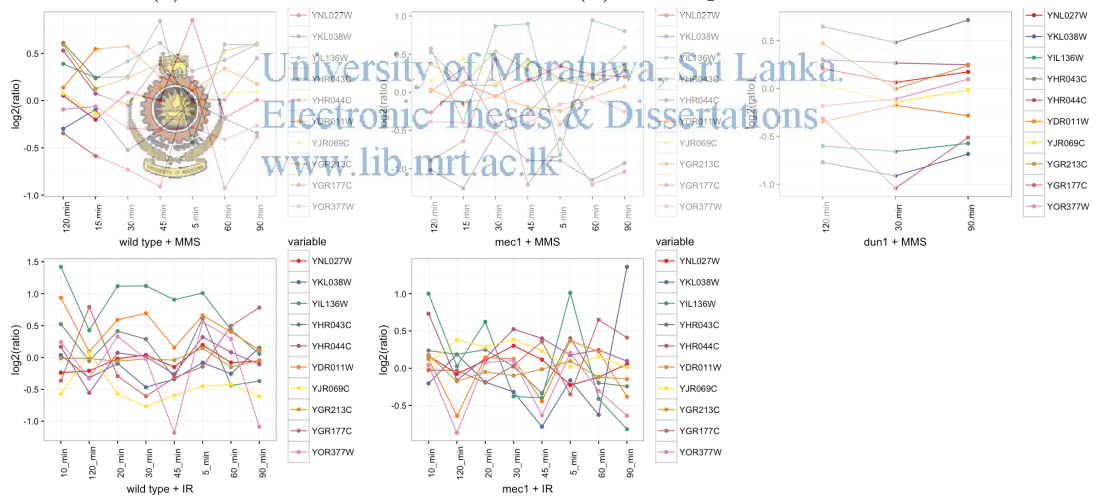
(a) Before normalization

(b) After median centering



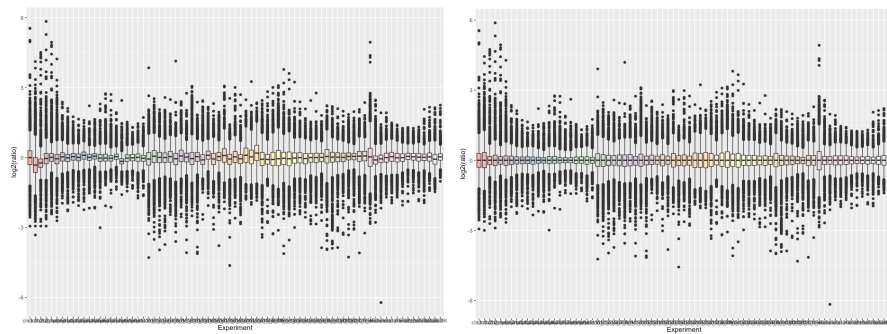
(c) After scale normalization

(d) After quantile normalization



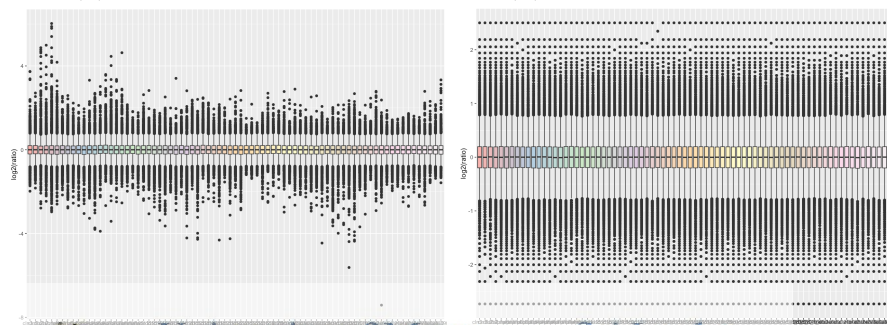
(e) After normalization, expression ratio profiles of first 10 genes for the time series over each major experiment

Figure A.2.8: Expressions 3 - before & after normalization/preprocessing



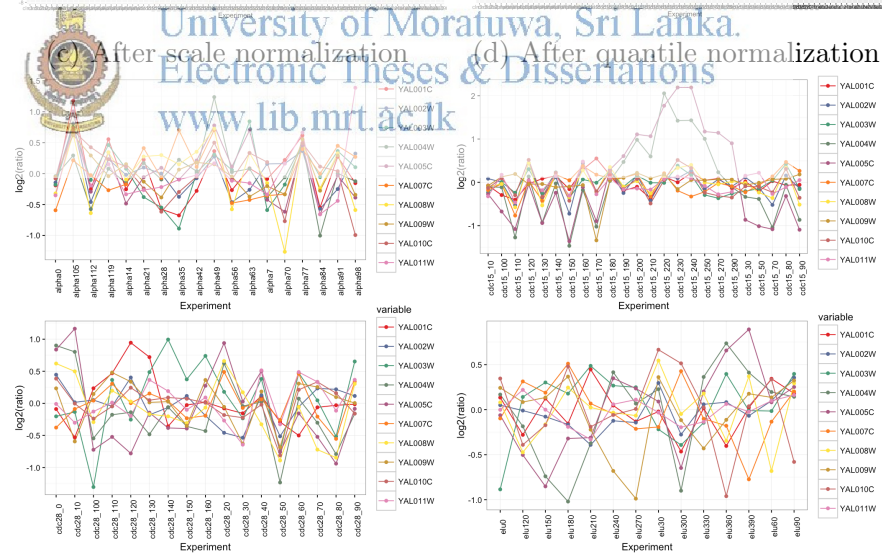
(a) Before normalization

(b) After median centering



(c) After scale normalization

(d) After quantile normalization



(e) After normalization, expression ratio profiles of first 10 genes for the time series over each major experiment

Figure A.2.9: Expressions 4 - Before & after normalization/preprocessing

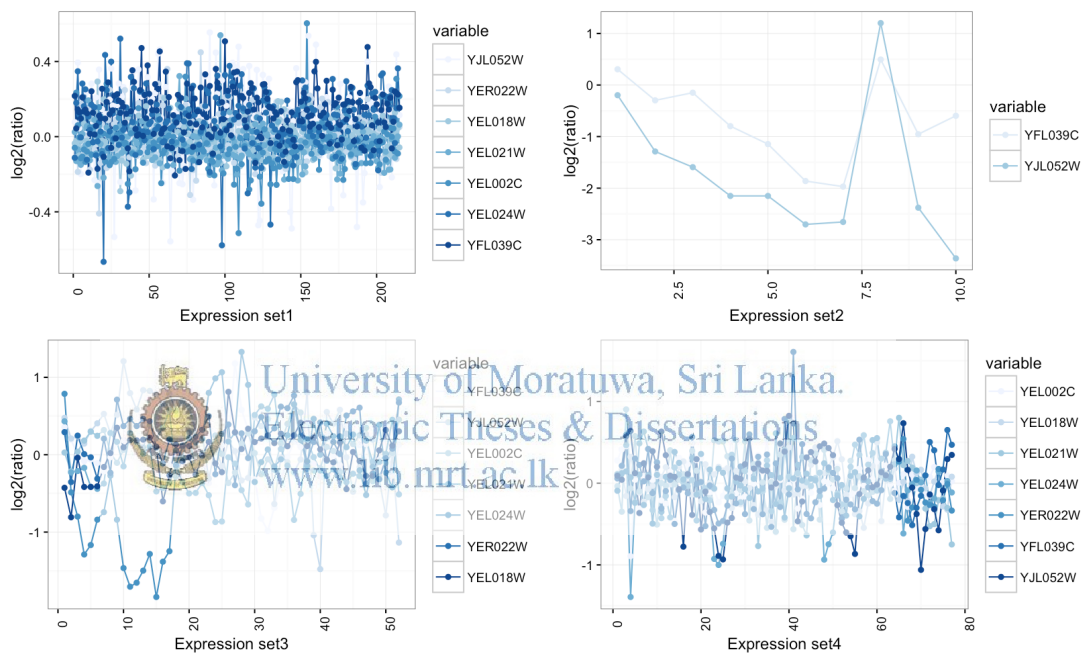


Figure A.2.10: Expression profiles of housekeeping genes

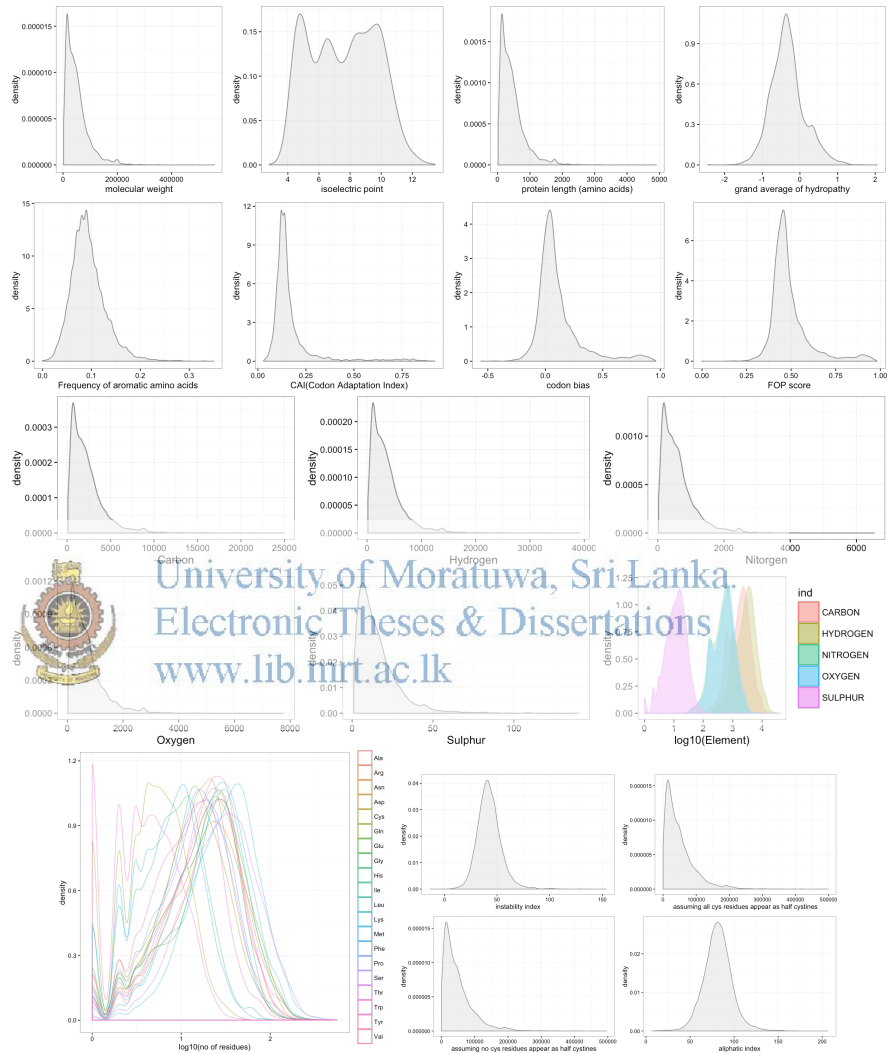


Figure A.2.11: Properties Data