

ENSEMBLE METHODS AND HYBRID ALGORITHMS
FOR COMPUTATIONAL AND SYSTEMS BIOLOGY



THE UNIVERSITY OF
SYDNEY

A thesis submitted in fulfilment of the requirements for the
degree of Doctor of Philosophy in the School of Information Technologies at
The University of Sydney

Pengyi Yang
April 2012

© Copyright by Pengyi Yang 2012
All Rights Reserved

This thesis is dedicated to my parents
Mengyi Heng and Xiaofang Yang
for their unconditional and unlimited love and encouragement.

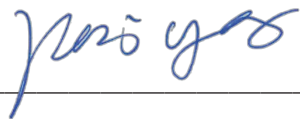
Abstract

Modern molecular biology increasingly relies on the application of high-throughput technologies for studying the function, interaction, and integration of genes, proteins, and a variety of other molecules on a large scale. The application of those high-throughput technologies has led to the exponential growth of biological data, making modern molecular biology a data-intensive science. Huge effort has been directed to the development of robust and efficient computational algorithms in order to make sense of these extremely large and complex biological data, giving rise to several interdisciplinary fields, such as computational and systems biology.

Machine learning and data mining are disciplines dealing with knowledge discovery from large data, and their application to computational and systems biology has been extremely fruitful. However, the ever-increasing size and complexity of the biological data require novel computational solutions to be developed. This thesis attempts to contribute to these inter-disciplinary fields by developing and applying different ensemble learning methods and hybrid algorithms for solving a variety of problems in computational and systems biology. Through the study of different types of data generated from a variety of biological systems using different high-throughput approaches, we demonstrate that ensemble learning methods and hybrid algorithms are general, flexible, and highly effective tools for computational and systems biology.

Statement of Originality

I hereby certify that this thesis contains no material that has been accepted for the award of any other degree in any university or other institution.



Pengyi Yang

April, 2012

Publications

Most of the research conducted during my PhD candidature has been published, or submitted for publication, in internationally refereed journals, conference proceedings, and invited book chapters. Those materials that contributed significantly to my PhD candidature are included in this thesis.

List of journal publications

1. Pengyi Yang, Sean J. Humphrey, Daniel J. Fazakerley, Matthew J. Prior, Guang Yang, David E. James, and Jean Yee-Hwa Yang, **Re-Fraction: a machine learning approach for deterministic identification of protein homologs and splice variants in large-scale MS-based proteomics**, *Journal of Proteome Research*, doi: 10.1021/pr300072j, 2012
2. Pengyi Yang, Jia Ma, Penghao Wang, Yunping Zhu, Bing B. Zhou, Yee Hwa Yang, **Improving X!Tandem on peptide identification from mass spectrometry by self-boosted Percolator**, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, accepted.
3. Penghao Wang, Pengyi Yang, Jean Yee-Hwa Yang, **OACAP: An Open Comprehensive Analysis Pipeline for iTRAQ**, *Bioinformatics*, doi: 10.1093/bioinformatics/bts150, 2012
4. Pengyi Yang¹, Joshua W.K. Ho¹, Yee Hwa Yang, Bing B. Zhou, **Gene-gene interaction filtering with ensemble of filters**, *BMC Bioinformatics*, 12(Suppl 1):S10, 2011

¹equal contribution

5. Pengyi Yang, Yee Hwa Yang, Bing B. Zhou, Albert Y. Zomaya, **A review of ensemble methods in bioinformatics**, *Current Bioinformatics*, 5(4):296-308, 2010
6. Pengyi Yang, Joshua W.K. Ho, Albert Y. Zomaya, Bing B. Zhou, **A genetic ensemble approach for gene-gene interaction identification**, *BMC Bioinformatics*, 11:524, 2010 (labeled as “highly accessed” by BMC Bioinformatics)
7. Penghao Wang, Pengyi Yang, Jonathan Arthur, Jean Yee Hwa Yang, **A dynamic wavelet-based algorithm for pre-processing mass spectrometry data**, *Bioinformatics*, 26(18):2242-2249, 2010
8. Paul D. Yoo, Yung S. Ho, Jason Ng, Michael Charleston, Nitin K. Saksena, Pengyi Yang, Albert Y. Zomaya, **Hierarchical kernel mixture models for the prediction of AIDS disease progression using HIV structural gp120 profiles**, *BMC Genomics*, 11:S4, 2010
9. Pengyi Yang, Zili Zhang, Bing B. Zhou, Albert Y. Zomaya, **A clustering based hybrid system for biomarker selection and sample classification of mass spectrometry data**, *Neurocomputing*, 73:2317-2331, 2010
10. Pengyi Yang, Bing B. Zhou, Zili Zhang, Albert Y. Zomaya, **A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data**, *BMC Bioinformatics*, 11:S5, 2010
11. Pengyi Yang, Liang Xu, Bing B. Zhou, Zili Zhang, Albert Y. Zomaya, **A particle swarm based hybrid system for imbalanced medical data sampling**, *BMC Genomics*, 10:S34, 2009
12. Pengyi Yang, Zili Zhang, **An embedded two-layer feature selection approach for microarray data analysis**, *IEEE Intelligent Informatics Bulletin*, 10:24-32, 2009
13. Zili Zhang, Pengyi Yang, Xindong Wu, Chengqi Zhang, **An agent-based hybrid system for microarray data analysis**, *IEEE Intelligent Systems*, 24(5):53-63, 2009
14. Zili Zhang, Pengyi Yang, **An ensemble of classifiers with genetic algorithm based feature selection**, *IEEE Intelligent Informatics Bulletin*, 9:18-24, 2008

List of conference publications

1. Pengyi Yang, Zili Zhang, Bing B. Zhou, Albert Y. Zomaya, **Sample subsets optimization for classifying imbalanced biological data**, In: *Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKD-D)*, LNAI 6635, 333-344, 2011
2. Li Li, Pengyi Yang, Ling Qu, Zili Zhang, Peng Cheng, **Genetic algorithm-based multi-objective optimisation for QoS-aware web services composition**, In: *Proceedings of the 4th International Conference on Knowledge Science, Engineering and Management (KSEM)*, LNAI 6291, 549-554, 2010
3. Pengyi Yang, Li Tao, Liang Xu, Zili Zhang, **Multiagent framework for bio-data mining**, In: *Proceedings of the Fourth Rough Set and Knowledge Technology (RSKT)*, LNCS 5589, 200-207, 2009
4. Pengyi Yang, Zili Zhang, **A clustering based hybrid system for mass spectrometry data analysis**, In: *Proceedings of Pattern Recognition in Bioinformatics (PRIB)*, LNBI 5265, 98-109, 2008. (This paper won Student Travel Award)
5. Pengyi Yang, Zili Zhang, **A hybrid approach to selecting susceptible single nucleotide polymorphisms for complex disease analysis**, In: *Proceedings of BioMedical and Engineering Informatics (BMEI)*, IEEE Press, 214-218, 2008
6. Pengyi Yang, Zili Zhang, **Hybrid methods to select informative gene sets in microarray data classification**, In: *Proceedings of Australian Conference on Artificial Intelligence (AI)*, LNAI 4830, 811-815, 2007

List of invited book chapters

1. Pengyi Yang, Yee Hwa Yang, Bing B. Zhou, Albert Y. Zomaya, **Stability of feature selection algorithms and ensemble feature selection methods in bioinformatics**, In: *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data*, Wiley-Blackwell, John Wiley & Sons Ltd., New Jersey, USA, forthcoming.

List of posters

1. Pengyi Yang, Penghao Wang, Bing B. Zhou, Yee Hwa Yang, **Studying false positive identifications in target-decoy search of mass spectrometry-based proteomics**, Human Proteome World Congress 2010 (HUPO), Sydney, Australia
2. Pengyi Yang, Bing B. Zhou, **A clustering based hybrid system for mass spectrometry data analysis**, EII PhD School 2009, University of Queensland, Brisbane, Australia (**Highly Commended Award**)

Acknowledgements

I thank my supervisor, A/Prof. Bing B. Zhou, for his constant support and guidance throughout my PhD study. Bing has given me much freedom to pursue my own research interests. His extremely valuable support has shaped me into a highly motivated person who is capable of carrying out research independently. My research interests are in the area of computational biology and bioinformatics; Bing has provided strong support on the algorithmic and computational aspects that are critical for developing robust and computationally effective algorithms for solving complex biological problems. In research, Bing has always been insightful and keen to understand the fundamentals. Critical thinking is a key requirement that Bing encourages in his students. In every discussion I had with Bing, the central element was always the understanding of the basics of the question. I would like to express my gratitude to Bing for his guidance and supervision that helped me to realize my full potential in my research.

Computational biology is a multi-disciplinary research field. Throughout my PhD study, I have received significant help from my associate supervisors, Dr Jean Yee-Hwa Yang, Prof. David E. James, and Prof. Albert Y. Zomaya. Jean has been the primary person to open the door of computational biology to me. Through Jean, I had the chance to collaborate with statisticians and biologists, which has helped me to develop the ability to work with those from diverse backgrounds. It is also through Jean's supervision that I became much more competent in formulating biological problems in a mathematical and analytical way, which is critical in real-world applications, and I would like to express my gratitude to Jean for making all this possible. I spent the third year of my PhD study in Prof. David James' extremely dynamic and active laboratory in the Garvan Institute of Medical Research. I thank David for providing me with this unique opportunity to work with biologists and cutting-edge technologies. In particular, I would like to thank Mr Sean Humphrey for patiently teaching me every detail of mass spectrometry-based proteomics. Through Sean's help, I had the opportunity to

learn the details of experiment preparation and cutting-edge mass spectrometry, which have greatly strengthened my appreciation and understanding of biological systems and biotechnologies. During my PhD study, I have participated in several national and international conferences and PhD schools. I owe gratitude to Prof. Albert Y. Zomaya who has always been supportive about my research and made possible my participation in conferences and PhD schools.

I had the great honour of working with many excellent collaborators and mentors who made my PhD study an exceptional learning experience. Particularly, I would like to thank Dr Joshua W.K. Ho from the Harvard Medical School; Dr Daniel J. Fazakerley and Dr Matthew Prior from the Garvan Institute of Medical Research; Dr Penghao Wang, Dr Uri Keich, Prof. Cristobal dos Remedios, and Ms Alana Mohammed from the University of Sydney; Dr Jie Ma from the Beijing Proteome Research Center; and Prof. Zili Zhang from Southwest University.

I thank Ms Katie Yang and Dr Joachim Gudmundsson for making my NICTA research a rewarding experience, and I thank Ms Evelyn Riegler, Dr Bernhard Scholz, and Prof. Sanjay Chawla for ensuring my study at the University of Sydney was as smooth as possible.

I acknowledge the scholarship support of NICTA International Postgraduate Award and NICTA Research Project Award, and a variety of financial support awarded by the University of Sydney to participate in national and international conferences.

Last but not least, I would like to express my greatest gratitude to my parents for their everlasting love and support that encouraged me throughout my entire life.

Contents

	iii
Abstract	iv
Statement of Originality	v
Publications	vi
Acknowledgements	x
List of Tables	xvii
List of Figures	xix
1 Introduction	1
1.1 Methods in computational and systems biology	2
1.1.1 Genome-wide association (GWA) studies	3
1.1.2 Gene expression microarray	4
1.1.3 Mass spectrometry (MS)-based proteomics	6
1.1.4 Ensemble methods and hybrid algorithms	7
1.2 Contributions and organization of the thesis	7
2 Ensemble & Hybrid Algorithms in Computational Biology	11
2.1 Ensemble methods	12
2.1.1 Ensemble methods for classification	12
2.1.2 Ensemble methods for feature selection	18
2.2 Hybrid algorithms	25

3	Gene-Gene Interaction Filtering	28
3.1	Gene-gene interaction in GWA studies	28
3.2	Filtering gene-gene interactions	29
3.2.1	ReliefF algorithm	30
3.2.2	Tuned ReliefF (TuRF)	31
3.2.3	Instability of ReliefF-based algorithm	32
3.3	Ensemble of filters for gene-gene interaction filtering	33
3.4	Experiment on simulation and real-world GWA data	34
3.4.1	The effect of the sample order dependency	35
3.4.2	The origin of the sample order dependency	37
3.4.3	Determination of ensemble size	38
3.4.4	Ensemble approach to improve success rate in SNP filtering	39
3.5	Summary	40
3.6	Software availability	43
4	Gene-Gene Interaction Identification	44
4.1	Combinatorial testing for gene-gene interaction identification	44
4.2	Genetic ensemble hybrid algorithm	47
4.2.1	Genetic component	48
4.2.2	Integration functions	50
4.2.3	Selecting classifiers	52
4.3	Evaluation datasets	52
4.4	Evaluation statistics	53
4.4.1	Evaluation statistics for single algorithm	53
4.4.2	Evaluation statistics for combining algorithms	54
4.5	Experiments and results	56
4.5.1	Classifier selection for ensemble construction	56
4.5.2	Simulation results	57
4.5.3	Real-world data application	65
4.6	Summary	68
4.7	Software availability	70
5	Gene Sets Selection From Microarray	71
5.1	Microarray data from a computational viewpoint	71

5.2	Hybrid approach for gene set selection	72
5.2.1	Multiple filter enhanced genetic ensemble	73
5.2.2	Multiple filtering algorithms score mapping	75
5.3	Filters and classifiers	76
5.3.1	Filter algorithms	77
5.3.2	Classification components	78
5.4	Experiment designs and results	79
5.4.1	Datasets and data pre-processing	79
5.4.2	Implementation	80
5.4.3	Results	82
5.5	Summary	89
6	Post-processing MS-based Proteomics Data	90
6.1	Peptide-spectrum match post-processing	90
6.2	Experiment settings and implementations	93
6.2.1	Evaluation datasets	93
6.2.2	Database searching	93
6.2.3	Percolator for X!Tandem search results	94
6.2.4	Semi-supervised learning on creating training dataset	96
6.2.5	Self-boosted Percolator	97
6.2.6	Performance comparison on PSM post-processing	98
6.3	Results and discussion	99
6.3.1	Percolator is sensitive to PSM ranking	99
6.3.2	Determining the number of boost runs	100
6.3.3	PSM post-processing	102
6.3.4	Protein identification	103
6.4	Summary	103
6.5	Software availability	105
7	Extracting Complementary Proteomics Biomarkers	106
7.1	Biomarker discovery from MS-based proteomics data	106
7.2	Feature correlation and complementary feature selection	107
7.3	A clustering-based hybrid approach	109
7.3.1	Filter-based prefiltering	112

7.3.2	<i>k</i> -means clustering	113
7.3.3	Cluster feature extraction and representative selection	113
7.3.4	Using genetic ensemble for m/z biomarker identification	114
7.4	Evaluation datasets and experiment designs	114
7.4.1	Datasets	114
7.4.2	Data pre-processing	117
7.4.3	Results evaluation	117
7.5	Experimental results	118
7.5.1	Evaluating <i>k</i> value of <i>k</i> -means clustering	118
7.5.2	Sample classification	119
7.5.3	Correlation reduction	127
7.6	Discussion and summary	129
8	Conclusions and Future Work	131
8.1	Conclusions of the thesis	131
8.2	Future directions	133
	Bibliography	135

List of Tables

3.1	Simulated SNP datasets for filtering experiment	34
3.2	Average cumulative success rate in retaining a functional SNP pair	40
4.1	Genetic algorithm parameter settings.	49
4.2	Simulated SNP datasets for identification experiment	53
4.3	Functional SNP pair identification for balanced data	58
4.4	Functional SNP pair identification for imbalanced data	58
4.5	Combining algorithms for SNP pair identification in balanced data	62
4.6	Combining algorithms for SNP pair identification in imbalanced data . . .	62
4.7	Two-factor interaction candidates of AMD	67
4.8	Three-factor interaction candidates of AMD	67
4.9	SNPs and environmental factors that associated with AMD	68
5.1	Microarray datasets used for algorithm evaluation.	80
5.2	Parameter setting for genetic ensemble.	81
5.3	Classification comparison using Leukemia dataset	83
5.4	Classification comparison using Colon dataset	83
5.5	Classification comparison using Liver dataset	84
5.6	Classification comparison using MLL dataset	84
5.7	Generation of convergence & subset size for MF-GE and GE	86
5.8	Top 5 most frequently selected genes	88
6.1	Summary of features used by Percolator for X!Tandem search results. . . .	95
7.1	MS datasets used in evaluation.	116
7.2	Error rate comparison on ovarian cancer-WCX2 dataset	121
7.3	Error rate comparison on ovarian cancer-WCX2-PBSII-a dataset	122
7.4	Error rate comparison on ovarian cancer-WCX2-PBSII-b dataset	123

7.5	Error rate comparison on prostate cancer-H4-PBS1 dataset	124
7.6	Significance test on error rates	126
7.7	Correlation evaluation on selected m/z features	127

List of Figures

1.1	Information flow in the cell	1
1.2	SNP chip and data structure	4
1.3	Gene expression microarray data	5
1.4	Protein identification using mass spectrometry	6
2.1	Hypothesis space partitioning with ensemble of classifiers.	14
2.2	Ensemble of classifiers using majority voting.	15
2.3	Popular ensemble methods.	16
2.4	Categorization of feature selection algorithms	20
2.5	Constructing ensemble of filters	22
2.6	Constructing ensemble of wrappers	23
3.1	Ensemble of filters	34
3.2	Stability in SNP filtering	36
3.3	Number of times a tie-breaking case happens	37
3.4	SNP filtering with tie causing samples removed	38
3.5	Determining the size of ensemble of filters	39
3.6	Success rate for retaining a functional SNP pair	41
3.7	Average cumulative success rate of single filter and its ensemble	42
4.1	Genetic ensemble system	47
4.2	Selection of base classifiers and ensemble configuration	57
4.3	TPR and FDR estimation of GE at different rank cut-off	59
4.4	TPR and FDR estimation of GE at different frequency cut-off	60
4.5	Power of GE, PIA, MDR, and combination of the three algorithms	64
5.1	Flow chart of the MF-GE hybrid system	74
5.2	Multiple filter score mapping example	76

5.3	Sample classification comparison	85
5.4	Multi-filter consensus scores	86
5.5	Gene subset size and generation of convergence	87
6.1	PSM rank effect on creating training dataset	97
6.2	Self-boosting of Percolator	100
6.3	Correlations of PSM rankings from boost runs	101
6.4	Number of accepted PSMs with respect to q -values	102
6.5	Number of accepted proteins and number of assigned PSMs	104
7.1	The overall work flow of the FCGE hybrid system.	110
7.2	k value evaluation of FCGE hybrid system	119
7.3	Average blocking accuracy according to different k values.	120
7.4	Correlation evaluation on selected m/z features	128

Chapter 1

Introduction

Central dogma is the classic framework for studying and understanding biological systems and their functions [46]. It loosely divides the information in biological systems into three levels, i.e. genes, transcripts, and proteins, in which the information flows from gene to transcript by transcription and from transcript to protein by translation (Figure 1.1). Although there are many other information flows in a variety of biological systems, the studies of genes, transcripts, and proteins and the information flows among them have been the most fundamental subjects in molecular biology research.

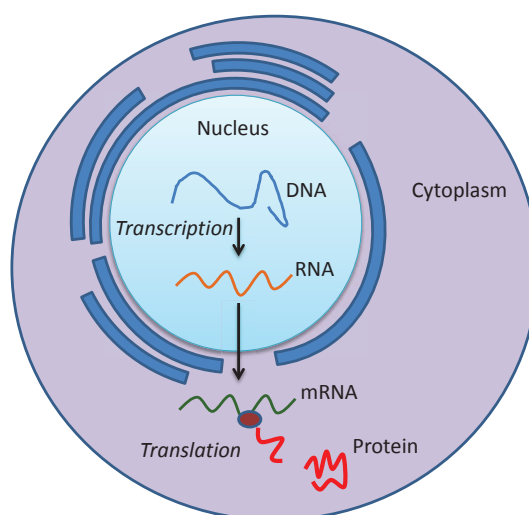


Figure 1.1: The biological system of the cell. The information flows from genes (DNA) to transcripts (RNA and mRNA) and to proteins through transcription and translation.

The collections of all the genes, transcripts, and proteins in a cell, tissue, or organism at a given time or state are commonly referred to as genome, transcriptome, and proteome [6], respectively. With the development and the application of various high-throughput technologies, we are in the era of profiling and interrogating the entire genome, transcriptome, and proteome of a cell, tissue, organism, or even multiple organisms, giving rise to new emerging research fields such as genomics [39], transcriptomics [16], and proteomics [147] among numerous other “-omics” science. The explosion of the biological data generated from -omics studies and the attempt to understand tens of thousands of genes, proteins, and other biological molecules in a systematic way transformed molecular biology into an information-based science that is best exemplified by the rise of inter-disciplinary fields such as computational biology and systems biology. The key characteristic of computational and systems biology is the application of computational techniques and statistical models for the analysis and interpretation of the huge amount of biological data. The knowledge discovered from these data and systems could have significant impact on biology and human welfare.

Machine learning and data mining are intelligent computational approaches used to extract information from large datasets and discover relationships. Their application to computational and systems biology have been extremely fruitful [111]. Ensemble learning and hybrid algorithms are intensive studies techniques in machine learning and data mining. The goal of this thesis is to contribute to the fast-growing field of computational and systems biology by designing ensemble learning methods and hybrid algorithms and applying them to solve biological and computational challenges in genomics, transcriptomics, and proteomics.

1.1 Methods in computational and systems biology

Systems biology aims to study and understand biological systems in its full scale and complexity. It is characterized by using high-throughput technologies to identify and profile biological systems in high speed and large scales. It relies on computational methods for effective data analysis and interpretation. Here we provide a brief introduction on some of the key high-throughput technologies utilized for studying genomics, transcriptomics, and proteomics and the main questions that associated with each of them. Specifically, at the genomic level, we introduce genome-wide association (GWA)

studies, at the transcriptomic level, we focus on microarray-based gene expression profiling, and at the proteomic level, we describe mass spectrometry (MS)-based protein identification. These topics are the main focus of our research and are the subjects that this thesis is devoted to. They span across genomics and transcriptomics to proteomics, capturing the main aspects of systems biology.

1.1.1 Genome-wide association studies

Single nucleotide polymorphisms (SNPs) are single-base-pair variants on DNA sequences that contribute to the genotype difference among individuals. Genome-wide association (GWA) studies are designed to specifically explore SNP genotypes to understand the genetic basis of many common complex diseases [85]. The studies rely on screening common SNPs and comparing the variations between individuals who have a certain disease (case) from a control population of individuals (control) by adopting a case-control study design [88]. The rationale is that comparing the SNP genotypes of case and control samples can provide critical insight to the genetic basis and the hereditary aspects of complex diseases. One of the key technologies that enables the genome-wide screening of SNPs is known as SNP chips [72]. SNP chips interrogate alleles by hybridizing the target DNA to the allele-specific oligonucleotide probes on the chips [188]. Since a DNA sequence containing a SNP may match perfectly to a probe-producing a stable hybridization, or be a mismatch to the probe-producing an unstable hybridization, the amount of DNA that could be found in the stable hybridization is relatively much more abundant than what could be found in unstable hybridization. Based on the amount of hybridization of the target DNA to each of those probes, one can determine if an allele is homozygous or heterozygous. Figure 1.2 is a schematic illustration of SNP chips. On the SNP chip, each spot corresponds to a SNP site on the genome. The data obtained from SNP chips is a matrix with each position providing a profiling of the genotype of a SNP as homozygous or heterozygous alleles inherited from the parents [148]. Each row represents a sample that has been genotyped, and the last column is the class label for the disease status of each sample.

GWA studies have been proven to be extremely useful for locating disease associated genes in complex diseases. Some of the most cited studies include the identification of genes *TCF7L2* and *SLC30A98*, which contribute to the risk of developing type 2 diabetes [180], and the identification of genes *CFH* and *ARMS2* as the risk factors for

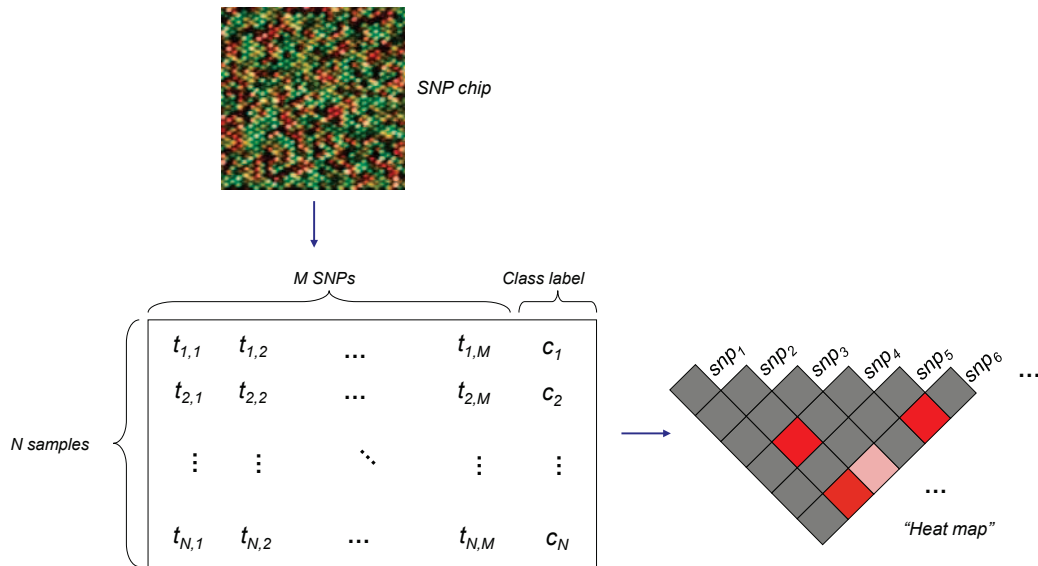


Figure 1.2: A schematic illustration of SNP chip and the data structure. A SNP chip is applied for genotyping and the data matrix obtained is a categorical data matrix with each variable taking a genotype of AA , AB , or BB corresponding to homozygous or heterozygous alleles. The SNP-disease associations and the SNP-SNP interactions can be represented as a “heat map” with brighter colours indicating stronger associations.

developing age-related macular degeneration [103]. Some of the main computational challenges in GWA data analysis include data normalization [31], SNP calling [161], disease-associated SNP identification [87, 142], and gene-gene interaction identification [42, 59]. In particular, the analysis of the huge amount of SNP data has been the bottleneck. That is, the number of SNPs considered in a typical GWA study is very large compared to the number of samples, giving an extremely high SNP-to-sample ratio. Furthermore, given the large number and the high density of SNPs in a genome, the SNP genotyping process is subject to errors [155]. Therefore, the development of computational algorithms that are robust to data noise and high data dimensionality, and can efficiently process several hundreds of thousands of SNPs is the key to successful GWA studies [122].

1.1.2 Gene expression microarray

Developed in the mid-90s, a microarray-based hybridization approach [49, 174] has served as the key high-throughput technology for quantifying the expression of genes

at the transcript level for more than a decade. Although there are a few types of microarrays, they utilize essentially the same principle for measuring gene expressions [184]. Essentially, gene expression microarray relies on hybridization to capture mRNA expressed in the cells, tissues, and organisms with the complementary probes manufactured on the glass slides. Using the intensities of fluorophores labelled on mRNAs as the surrogate of gene expression levels, we are able to compare the relative changes between cells and tissues from different treatments (Figure 1.3). Following a decade of development, microarray has become a highly effective transcriptome profiling technology for model organisms where the genomes are relatively complete. Tens of thousands of genes can be measured simultaneously, which provides a holistic measurement of biological systems under various treatments and conditions.

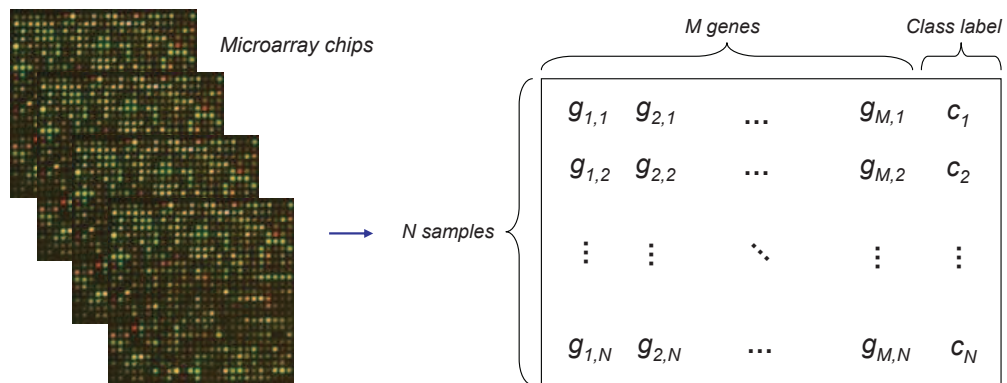


Figure 1.3: A schematic illustration of gene expression microarray data. From the computational viewpoint, microarray data can be viewed as an $N \times M$ matrix. Each row represents a sample while each column represents a gene except the last column which represents the class label of each sample. $g_{i,j}$ is a numeric value representing the gene expression level of the i^{th} gene in the j^{th} sample. c_j in the last column is the class label of the j^{th} sample

The analysis of microarray data has been an extensively studied subject. The fundamental issues include how to (1) normalize data so as to reduce data noise and enhance biological signal [160, 205], (2) group samples and genes into clusters based on their expression profiles [68, 186], (3) identify genes where the expression are up- and down-regulated (collectively known as differentially expressed (DE) genes) with respect to the treatments or disease status [57, 181], (4) identify enriched biological pathways [187], (5) computationally select key genes and gene subsets that are associated with the treatments or disease status [55, 74], and (6) classify samples based on their gene expression profiles [56, 70].

1.1.3 Mass spectrometry-based proteomics

The study of the global protein translation in the cell, tissue or organism is known as proteomics [2]. The goal of proteomic research is to identify and quantify all the proteins present in a cell, tissue or organism at a specific state or moment. Liquid chromatography-mass spectrometry (LC-MS)-based high-throughput proteomics is the key technology for such a large-scale profiling. With the tandem design (LC-MS/MS), increased sensitivity and specificity can be achieved [79].

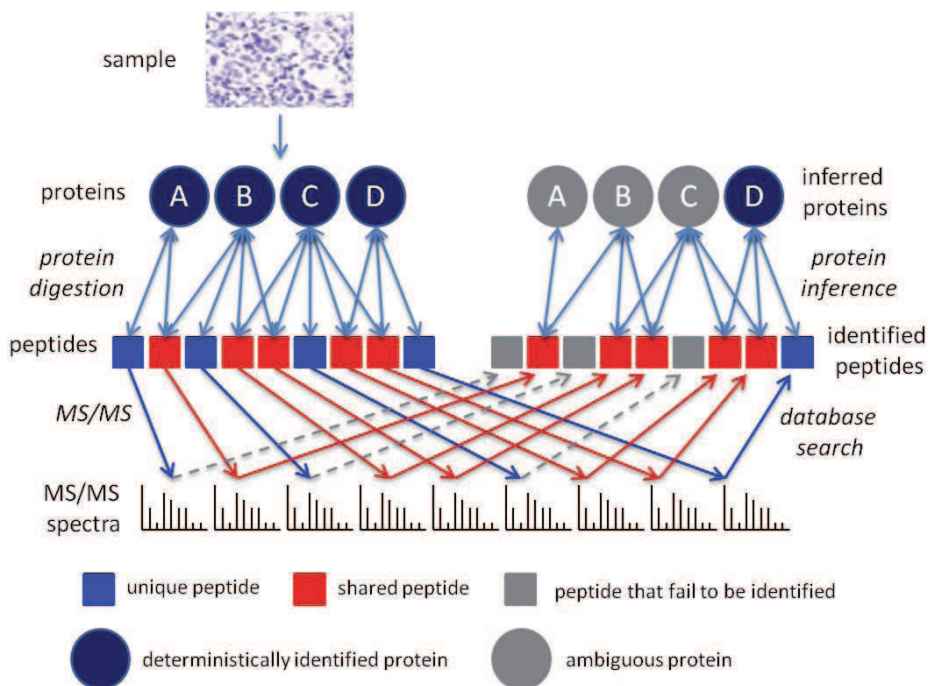


Figure 1.4: A schematic illustration of experimental procedures and computational procedures in protein identification using mass spectrometry.

In a typical MS-based experiment, cell or tissue samples are extracted and the protein mixture from the samples is purified and digested with an enzyme such as trypsin. The digested protein mixture is then injected into liquid chromatography and captured by a mass spectrometer or tandem mass spectrometer (LC-MS/MS) according to the mass/charge (m/z) of the generated peptide and peptide fragment ions. The output from the mass spectrometer is spectra, each corresponding to a peptide or peptide fragment.

LC-MS/MS-based proteomics relies highly on the computational analysis. Typically, the raw spectra files are processed by a denoising algorithm [195], and from those spectra, the peptides are identified [38]. This is commonly accomplished by comparing

the observed spectra with theoretical spectra generated *in silico* from a given protein database (database searching) [43, 62], or with an annotated spectral library (library searching) [45, 109]. The identified peptides are then further post-processed for filtering potential false positive identifications [101, 141], and the filtered peptides are then used to infer the proteins that may present in the sample [139]. Figure 1.4 summarizes the experimental procedures and computational procedures.

After determining the protein identifies and abundances in a sample, the data can be analysed in a similar fashion to microarray-based gene expression profiling. Specifically, similar questions are commonly asked, such as disease-associated protein identification [84], and sample classification based on the protein abundance [200, 215].

1.1.4 Ensemble methods and hybrid algorithms

Ensemble methods and hybrid algorithms are fast developing techniques in the field of data mining and pattern recognition. These techniques have been increasingly applied to processing the large amount of biological data generated from using aforementioned high-throughput technologies. The strength of ensemble methods mainly reside in the robustness to the data noise. This is commonly achieved through various types of model averaging techniques which are one of the most important components in ensemble methods. For hybrid algorithms, they are, by definition, comprised of multiple algorithms and therefore are highly specialized for solving complex biology problems that are often modular and require the application of a diverse set of algorithmic tools. In Chapter 2, we will briefly review some of the most popular ensemble methods and hybrid algorithms. Those techniques will serve as the key techniques from which the followup chapters build on and extend to specific biological questions and systems.

1.2 Contributions and organization of the thesis

In this thesis, we present our research on designing ensemble learning methods and hybrid algorithms for addressing some of the key biological questions in computational and systems biology. Specifically, the organization and the contributions of the thesis are as follows:

- In Chapter 2, we introduce some of the most popular ensemble methods and hybrid algorithms and review their applications in computational and systems biology. We start by describing the rationale behind ensemble methods. Then, based on the applications, we categorize the ensemble methods as those for sample classification and those for feature selection. The rest of the chapter mainly focuses on reviewing some of the most representative applications of ensemble methods and hybrid algorithms in dealing with some of the key questions in computational and systems biology. These literature reviews will serve as the motivation and the building blocks for the subsequent chapters of this thesis.
- Chapter 3 describes using the ensemble feature selection approach for filtering gene-gene interactions in complex diseases. In this chapter, we propose a novel ensemble of filters using the ReliefF algorithm and its variants. By permutating the samples in the GWA dataset, we can create multiple filters, each built on a permuted version of the original dataset. We demonstrate that this permutation and ensemble of filter approach is advantageous in that complementary information in the dataset can be extracted. We show that the original filter algorithms are unstable in terms of SNP ranking. A low reproducibility is observed with the ReliefF algorithm and its variants in SNP filtering. By using the proposed ensemble of filters, not only can we largely improve the reproducibility of SNP rankings but also we can significantly increase the success rate on ranking functional SNPs and interaction pairs. This is critical for the follow up gene-gene interaction identification.
- Chapter 4 is about gene-gene interaction and gene-environmental interaction identification. It takes the SNP filtering results from Chapter 3 and utilizes a much more computationally intensive procedure to jointly evaluate multiple SNPs and environmental factors for potential gene-gene interaction and gene-environmental interaction identification in complex disease. Our contribution here is in developing an effective algorithm for gene-gene interaction identification. Specifically, we propose a novel *genetic ensemble* approach that incorporates multiple classification algorithms in a genetic algorithm. By using three integration functions in a novel way to combine the results from multiple classification algorithms, we observe a large increase of power on identifying SNP interaction pairs, significantly better than using any single classifier. Moreover, we introduce an equation

for evaluating the degree of complementarity of results generated by different gene-gene interaction identification algorithms. We show that the proposed genetic ensemble algorithm generates complementary results to other algorithms and is therefore useful even when other algorithms are successfully applied for data analysis.

- In Chapter 5, we move on to the transcriptome level by analysing gene expression data generated from microarray. In particular, we design a hybrid algorithm for gene set selection for accurate classification of disease and control samples. Given the small sample size and the large number of genes measured by microarray, traditional approaches either use computationally efficient filter algorithms to evaluate each gene separately, or evaluate a subset of prioritized genes in combinations using computationally intensive wrapper algorithms. Different from the traditional approach, we propose a score-mapping strategy to combine the advantages of filter and wrapper algorithms in that multiple filter algorithms are used to pre-evaluate each gene from microarray data in a computationally efficient way, and the pre-evaluation scores are combined and fused to a genetic ensemble-based wrapper algorithm for gene set selection. We named this hybrid algorithm “MF-GE” and demonstrate that (1) MF-GE converges faster than genetic ensemble without the multiple-filter component; (2) the size of the gene subset selected by MF-GE is smaller than the original genetic ensemble; and (3) MF-GE is superior to several other filter and wrapper feature selection algorithms in terms of identifying discriminative genes in sample classification.
- From Chapter 6, we turn to the proteome level. In this chapter, we address one of the key computational challenges, known as post-processing of peptide identifications, in processing and analysing mass spectrometry (MS)-based proteomics data. In MS-based proteomics, proteins are digested to peptides prior to the MS analysis and the proteins that are present in the sample are inferred from the identified peptides after the MS analysis. Prioritizing true peptide identifications while removing false positive identifications is a key post-processing step for eliminating false positive protein identifications. We model this post-processing step as a semi-supervised learning (SSL) procedure and propose a cascade-ensemble learning approach to improve peptide identification results. The proposed method is considered as an ensemble approach in that multiple

learning models are built in a cascade manner; each attempts to improve the result for its next model. By using the cascade-ensemble learning approach, the SSL algorithm boosts itself to a stable state, producing many more peptide identifications at a controlled level of false discovery rate.

- Chapter 7 focuses on protein set selection for normal and disease sample classification. Here we propose a novel clustering-based hybrid algorithm to extract complementary protein sets. Those protein sets are functionally distinctive units and represent potential biological pathways that are each involved in a unique biological process. By selecting proteins from those diverse functional units, the proposed hybrid algorithm can reduce the dominance of some universal biological pathways and extract much more useful information from the proteomics dataset for accurate sample classification and disease discrimination. We compare the hybrid algorithm with four other competitive algorithms on protein selection and sample classification. The proposed hybrid algorithm is able to give significantly lower error rate on sample classification across 10 different classification algorithms. Furthermore, we show that the proteins selected by the hybrid algorithm are highly complementary, providing useful extra information on potential biomarker identification.
- In the final chapter (Chapter 8), we summarize the thesis and propose potential directions for future work.

Chapter 2

Ensemble and Hybrid Algorithms in Computational Biology: Methods and Reviews

This chapter is partially based on the following publication:

Pengyi Yang, Yee Hwa Yang, Bing B. Zhou, Albert Y. Zomaya, A review of ensemble methods in bioinformatics. Current Bioinformatics, 5(4):296–308, 2010

One key component in computational and systems biology is the application of computational techniques for analysing and integrating different biological data sources and types. Various computational techniques, especially machine learning and data mining algorithms, are applied, for example, (1) to select biomarkers such as genes or proteins that are associated with the traits of interest, (2) to classify different types of samples based on genomic, transcriptomic, and proteomic profiling of biological systems, and (3) for the integration of data from multiple levels such as the integrative analysis of transcriptomic and proteomic data.

These tasks are data intensive in nature and often involve solving multiple subtasks in a modular or parallel fashion in achieving the final result. In order to analyse these complex biological systems, multiple models and multiple algorithms may be combined to solve the problem in an efficient and effective way. *Ensemble methods* refer to combining multiple models to improve performance [81]. For example, in classification,

an ensemble of decision tree models, each generated from a bootstrap of the original dataset, may perform in a superior fashion to a single decision tree model on the same dataset. In contrast, *hybrid algorithms* refer to combining multiple algorithms for solving tasks that are modular in nature [8]. In particular, the original problems are often subdivided to smaller and functionally unique subproblems, and each subproblem is solved by an algorithmic component in the hybrid algorithm.

In this chapter, we briefly introduce some of the most popular ensemble methods and hybrid algorithms that have been successfully applied to computational and systems biology. We also review some of the most representative applications in gene expression microarray, MS-based proteomics, and gene-gene interaction identification from GWA studies. They will serve as the motivation and the building blocks for the rest of the thesis.

2.1 Ensemble methods

Based on their applications, we categorize ensemble methods into (1) ensemble methods for classification, and (2) ensemble methods for feature selection. Ensemble methods for classification have been established as a useful approach for improving sample classification accuracy [145]. For classification, ensemble methods are effective in extracting limited information, which is critical for bioinformatics applications where only a small sample size is available. In contrast to classification, ensemble feature selection is a fast-developing technique where the main focus has been to improve feature selection stability [82]. Yet, several recent studies have found that, besides improving feature selection stability, many other aspects such as sample classification accuracy can also benefit from the ensemble feature selection approach [1].

2.1.1 Ensemble methods for classification

2.1.1.1 The rationale

Ensemble methods for classification have been intensively studied in machine learning and pattern recognition. They are effective ways for improving classification accuracy and model stability [53]. In bioinformatics, ensemble methods provide the advantage of alleviating the small sample size problem by averaging and incorporating over multiple

models to reduce the potential on overfitting [54]. In this regard, the training data are used in a more efficient way, which is critical to many biological applications with limited sample size. Some ensemble methods such as *random forests* [21] are particularly useful for high-dimensional datasets because increased classification accuracy can be achieved by generating multiple prediction models, each with a different *feature* subset. These properties have a major impact on many different bioinformatics applications.

For the task of classification, increased accuracy is often obtained by aggregating a group of classifiers (referred to as *base classifiers*) as an ensemble committee and making the prediction for unseen data in a consensus way. The aim of designing/using ensemble methods is to achieve more accurate classification (on training data) as well as better generalization (on unseen data). However, this is often achieved at the expense of increased model complexity (decreased model interpretability) [107]. A better generalization property of the ensemble approach is often explained by using the classic bias-variance decomposition analysis [197]. Here we provide an intuitive interpretation of the advantage of ensemble approach.

Let the best classification rule (called *hypothesis*) h_{best} of a given induction algorithm for certain kind of data be the circle in Figure 2.1. Suppose the training data is free from noise, without any missing values, and sufficiently large to represent the underneath pattern. Then, we expect the classifier trained on the dataset to capture the best classification hypothesis represented as the circle. In practice, however, the training datasets are often confounded by small sample size, high dimensionality, and high noise-to-signal ratio, etc. Therefore, obtaining the best classification hypothesis is often nontrivial because there are a large number of suboptimal hypotheses in the hypothesis space (denoted as H in Figure 2.1a) that can fit the training data but do not generalize well on unseen data.

Creating multiple classifiers by manipulating the training data in an intelligent way allows one to obtain a different hypothesis space with each classifier (H_1, H_2, \dots, H_L ; where L is the number of classifiers), which may lead to a narrowed overlap hypothesis space (H_o) as shown in Figure 2.1b. By combining the classification rules of multiple classifiers using integration methods that take advantage of the overlapped region (such as averaging and majority voting), we are approaching the best classification rule by using multiple rules as an approximation. As a result, the ensemble composed in such a manner often appears to be more accurate.

To aggregate the base classifiers in a consensus manner, strategies such as *majority*

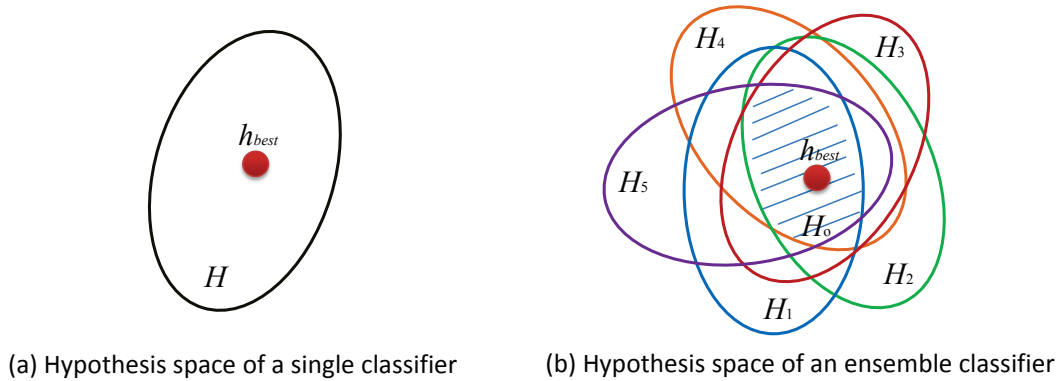


Figure 2.1: A schematic illustration of hypothesis space partitioning with ensemble of classifiers. By combining moderately accurate base classifiers, we can approximate the best classification rule h_{best} with the increase of model complexity. This can be achieved by combining base classifiers with averaging or majority voting, which takes advantage of the overlapped region.

voting or simple averaging are commonly used. Assuming the prediction outputs of the base classifiers are independent of each other (which, in practice, is partially achieved by promoting diversity among the base classifiers), the majority voting error rate ϵ_{mv} can be expressed as follows [110]:

$$\epsilon_{mv} = \sum_{i=\lfloor L/2 \rfloor + 1}^L \binom{L}{i} \epsilon^i (1 - \epsilon)^{L-i} \quad (2.1)$$

where L is the number of base classifiers in the ensemble. Given the condition that $\epsilon < \epsilon_{random}$ for ϵ_{random} being the error rate of a random guess and all base classifiers have identical error rate ϵ , the majority voting error rates ϵ_{mv} monotonically decreases and approaches 0 when $L \rightarrow \infty$.

Figure 2.2 shows an ideal scenario in which the dataset has two classes each with the same number of samples, the prediction of base classifiers is independent of each other, and all base classifiers have an identical error rate. It can be seen from the figure that, when the error rate of the base classifiers is smaller than 0.5, which is a random guess for a binary dataset with equal numbers of positive and negative samples, the ensemble error rate quickly gets smaller than the error rate of the base classifiers. If we add more base classifiers, the improvement becomes more significant. In this example, we used odd numbers of base classifiers where the consensus is made by $(L + 1)/2$

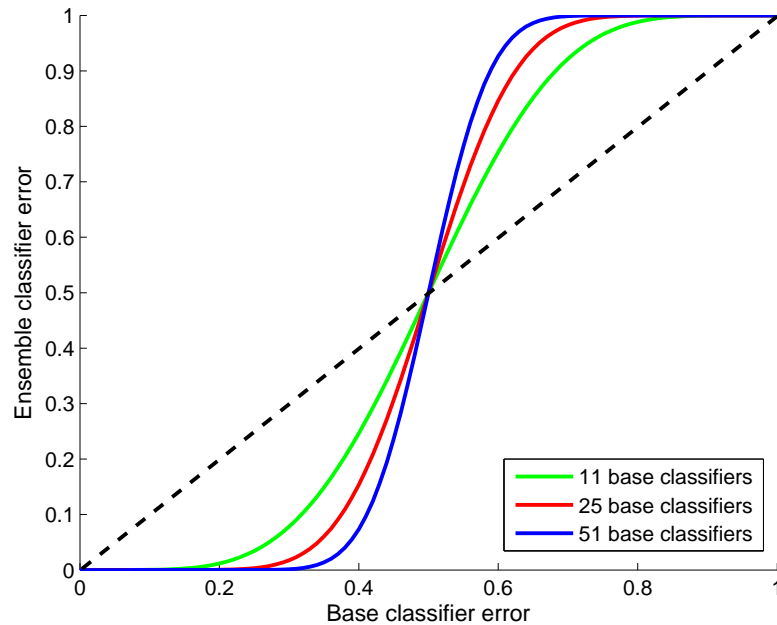


Figure 2.2: Majority Voting. The relationship of error rates of base classifiers and error rates of the ensemble classifier in majority voting. The diagonal line represents the case in which the base classifiers are identical to each other, while the three curved lines represent combining different numbers of base classifiers that are independent of each other.

classifiers. When using an even number of base classifiers, the consensus is made by $L/2 + 1$ classifiers.

From the above analysis, it is clear that in order to obtain an improvement the base classifiers need to be accurate (better than chance) and diverse from each other [193]. The need for diversity originates from the assumption that if a classifier makes a misclassification, there may be another classifier that complements it by correctly classifying the misclassified sample. Ideally, each classifier makes incorrect classifications independently. Popular ensemble methods like *bagging* [20] (Figure 2.3a) and *random subspace* [86] (Figure 2.3c) harness the diversity by using different perturbed data sets and different feature sets for training base classifiers, respectively. That is, each base classifier is trained on a subset of samples/features to obtain a slightly different classification hypothesis, and then combined to form the ensemble. The difference is that bagging relies on bootstrap sampling of the original dataset, whereas random subspace uses randomly selected samples without replacement to create multiple subsets. Random forests [21] (Figure 2.3d) is a combination of boosting on samples and random

subspace on features. As for *boosting* [173] (Figure 2.3b), diversity is obtained by increasing the weights of misclassified samples in an iterative manner. Each base classifier is trained and combined from the samples with different classification weights, and therefore, different hypotheses. By default, these three methods use *decision tree* as base classifiers because decision trees are sensitive to small changes on the training set [53], and are thus suited for the perturbation procedure applied to the training data.

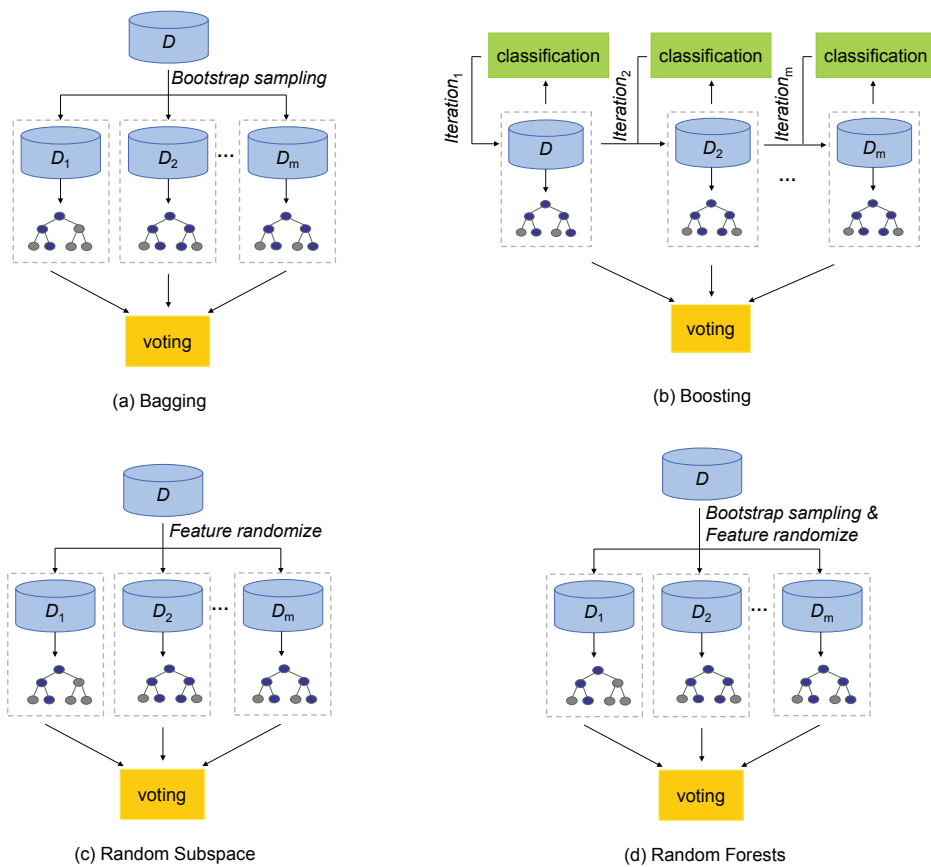


Figure 2.3: Schematic illustration of the four most popular ensemble methods. They are known as (a) bagging, (b) boosting, (c) random subspace, and (d) random forests.

2.1.1.2 Related literatures

Ben-Dor *et al.* [12] and Dudoit *et al.* [56] pioneered the application of bagging and boosting algorithms for classifying tumour and normal samples using gene expression

profiles. Both studies compared the ensemble methods with other individual classifiers such as k -nearest neighbour (k NN), clustering based classifiers, support vector machines (SVM), linear discriminant analysis (LDA), and classification trees. The conclusion was that ensemble methods of bagging and boosting performed similarly to other single classification algorithms included in the comparison.

In contrast to the results obtained by Dudoit *et al.* and Ben-Dor *et al.*, the follow up studies revealed that much better results can be achieved through minor tuning and modification. For instance, Dettling and Bühlmann [50] proposed an algorithm called LogitBoost that replaces the exponential loss function used in AdaBoost with a log-likelihood loss function. They demonstrated that LogitBoost is more accurate in classification of gene expression data compared with the original AdaBoost algorithm. Long [120] argued that the performance of AdaBoost can be enhanced by improving the base classifiers. He then proposed several customized boosting algorithms for microarray data classification. The experimental results indicate that the customized boosting algorithms performed favourably compared with SVM-based algorithms. In comparison to the single tree classifier, Tan and Gilbert [189] demonstrated that, overall, ensemble methods of bagging and boosting are more robust and accurate in microarray data classification using seven publicly available datasets.

In MS-based proteomics, Qu *et al.* [159] conducted the first study using boosting ensembles for classifying mass spectra serum profiles. A classification accuracy of 100% was estimated using the standard AdaBoost algorithm, while a simpler ensemble called “boosted decision stump feature selection” (BDSFS) showed slightly lower classification accuracy (97%) but gives more interpretable classification rules. A thorough comparison study was conducted by Wu *et al.* [200], who compared the ensemble methods of bagging, boosting, and random forests to individual classifiers of LDA, quadratic discriminant analysis, k NN, and SVM for MALDI-TOF (matrix assisted laser desorption/ionization with time-of-flight) data classification. The study found that among all methods, on average, random forests gives the lowest error rate with the smallest variance. Another recent study by Gertheiss and Tutz [67] designed a block-wise boosting algorithm to integrate feature selection and sample classification of mass spectrometry data. Based on LogitBoost, their method addresses the horizontal variability of the m/z values by dividing the m/z values into small subsets called blocks. Finally, the boosting ensemble has also been adopted as the classification and biomarker discovery component in the proteomic data analysis framework proposed by Yasui *et al.* [207].

In comparison to bagging and boosting ensemble methods, random forests holds a unique advantage because its use of multiple feature subsets is well suited for high-dimensional data such as those generated by microarray and MS-based proteomics studies. This is demonstrated by several studies such as [112] and [52]. In [112], Lee *et al.* compared the ensemble of bagging, boosting and random forests using the same experimental settings and found random forests was the most successful. In [52], the experimental results through ten microarray datasets suggest that random forests are able to preserve predictive accuracy while yielding smaller gene sets compared with diagonal linear discriminant analysis (DLDA), k NN, SVM, shrunken centroides (SC), and k NN with feature selection. Other advantages of random forests such as robustness to noise, lack of dependence upon tuning parameters, and the speed of computation have been demonstrated by Izmirlian [89] in classifying SELDI-TOF proteomic data.

Giving the good performance of random forests in high-dimensional data classification, the development of random forests variants is a very active research topic. For instance, Zhang *et al.* [213] proposed a deterministic procedure to form a forest of classification trees. Their results indicate that the performance of the proposed deterministic forest is similar to that of random forests, but with better reproducibility and interpretability. Geurts *et al.* [69] proposed a tree ensemble method called “extra-trees” which selects at each node the best among k randomly generated splits. This method is an improvement on random forests because unlike random forests, which are grown with multiple subsets, the base trees of extra-trees are grown from the complete learning set and by explicitly randomizing the cut-points.

2.1.2 Ensemble methods for feature selection

Feature selection is a key technique originating from the fields of artificial intelligence and machine learning [17,73] in which the main motivation has been to improve sample classification accuracy [48]. Since the focus is mainly on improving classification outcome, the design of feature selection algorithms seldom considers specifically which features are selected. Due to the exponential growth of biological data in recent years, many feature selection algorithms have been found to be readily applicable, or only require minor modification [172], for example, to identify potential disease-associated genes from microarray studies [201], proteins from MS-based proteomics studies [114], or SNP from GWA studies [214]. While sample classification accuracy is an important

aspect in many of those biological studies such as discriminating cancer and normal tissues, the emphasis is also on the selected features as they represent interesting genes, proteins, or SNPs. These biological features are often referred to as biomarkers and they frequently determine how further validation studies should be designed and conducted.

One unique issue arising from the application of feature selection algorithms in identifying potential disease-associated biomarkers, is that those algorithms may give unstable selection results [96]. That is, a minor perturbation in the data such as a different partition of data samples, removing a few samples, or even reordering the data samples may cause a feature selection algorithm to select a different set of features. For instance, typical microarray-based gene profiling studies produce high-dimensional datasets with several thousand genes and a few dozen samples. Commonly, a t -test may be used to rank the importance of the genes in discriminating disease and controls, tumours and normals, etc. It is possible that a small change in the dataset, such as removing a few samples, may cause the t -test to rank the genes differently. For those algorithms with stochastic components, simply rerun the algorithm with a different random seeding may give a different feature selection result. The term *stability* and its counterpart *instability* are used to describe whether a feature selection algorithm is sensitive or insensitive to small changes in the data and the settings of algorithmic parameters. The stability of a feature selection algorithm becomes an important property in many biological studies because biologists may be more confident about the feature selection results that do not change much with a small perturbation in the data or a rerun of the algorithm. While this subject has been relatively neglected in the past, we saw a fast-growing interest in recent years where different approaches to improve the stability of feature selection algorithms and different matrices for measuring them have been proposed. It has been demonstrated that ensemble methods could be used to improve feature selection stability and data classification accuracy [1]. In this chapter, we categorize different feature selection algorithms, introduce two common approaches for creating ensemble feature selection, and review recent development and applications of ensemble feature selection algorithms in computational and systems biology.

2.1.2.1 Categories of feature selection algorithms

From a computational perspective, feature selection algorithms can be broadly divided into three categories of *filter*, *wrapper*, and *embedded* approaches according to their

selection manners [73]. Figure 2.4 shows the schematic view according to the categorization.

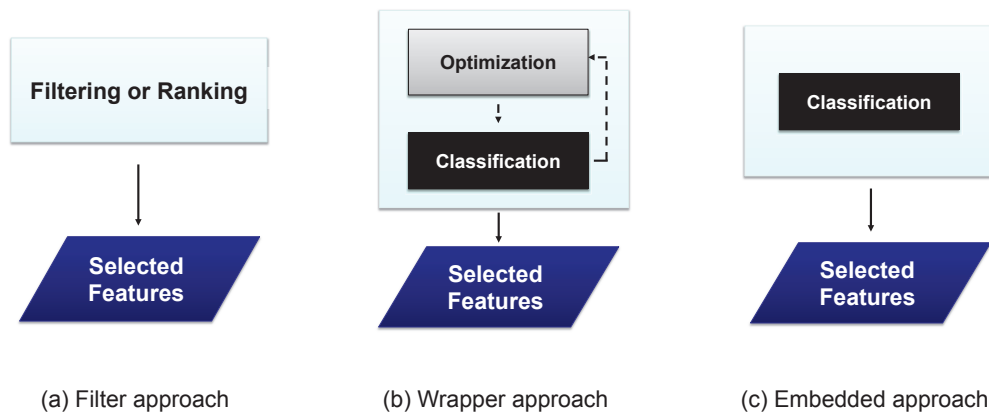


Figure 2.4: Categorization of feature selection algorithms. (a) Filter approach where feature selection is independent from the classification. (b) Wrapper approach where feature selection is explicitly performed by an inductive algorithm for sample classification in an iterative manner. (c) Embedded approach where feature selection is performed implicitly by an inductive algorithm during sample classification.

Filter algorithms commonly rank/select features by evaluating certain types of association or correlation with class label, etc. They do not optimize the classification accuracy of a given inductive algorithm directly. For this reason, filter algorithms are often computationally more efficient compared with wrapper algorithms. For numeric data analysis such as differentially expressed (DE) gene selection from microarray data or DE protein selection from mass spectrometry data, the most popular methods are probably the t -test and its variants [181]. As for categorical data types such as disease-associated SNP selection from GWA studies, the commonly used methods are χ^2 -test or odds ratio while increasingly popular methods are the ReliefF algorithm and its variants [130].

Although filtering algorithms often show good generalization and extend well on unseen data, they suffer from several problems. Firstly, filtering algorithms commonly ignore the effects of the selected features on sample classification of a given inductive algorithm. Yet the performance of the inductive algorithm could be useful for accurate phenotype classification [104]. Secondly, many filter algorithms are univariate and greedy based. They assume that each feature contributes to the phenotype independently and evaluate each feature separately. The feature set is often determined by ranking the features according to certain scores calculated by filter algorithms and selecting the

top- k candidates. Those assumptions are most likely invalid in biological systems, and the selection results produced in this way are often suboptimal.

Compared with filter algorithms, wrapper algorithms have several advantages. Firstly, wrapper algorithms incorporate the performance of an inductive algorithm in feature evaluation, and are therefore likely to perform well in sample classification. Secondly, most wrapper algorithms are multivariate and treat multiple features as a unit for evaluation. This property preserves the biological interpretation of genes and proteins since they are linked by pathways and function in groups. A large number of wrapper algorithms have been applied to gene selection of microarray and protein selection of mass spectrometry. Those include evaluation approaches such as genetic algorithm (GA)-based selection [92,116,117], and greedy approaches such as incremental forward selection [168], and incremental backward elimination [156].

Despite their common advantages, wrapper approaches often suffer from problems such as overfitting, since the feature selection procedure is guided by an inductive algorithm that fitted on training data. Therefore, the features selected by a wrapper approach may generalize poorly on new datasets if overfitting is not prevented. Other than that, wrapper algorithms are often much slower compared with filter algorithms (by several orders of magnitude), due to their iterative training and evaluating procedures.

An embedded approach is somewhat between the filter approach and the wrapper approach, where an inductive algorithm implicitly selects features during sample classification. As opposed to filter and wrapper approaches, embedded approaches rely on certain types of inductive algorithms and are therefore less generic. The most popular ones that apply for gene and protein selection are support vector machine-based recursive feature elimination (SVM-RFE) [74] and random forest-based feature evaluation [52].

2.1.2.2 Ensemble feature selection algorithms

Ensemble feature selection algorithms are composed for many reasons. Generally, the goals are to improve feature selection stability, or sample classification accuracy, or both simultaneously, as demonstrated in numerous studies [1, 93, 118]. In many cases, other aspects such as identifying important features or extracting feature interaction relationships could also be achieved with higher accuracy using ensemble feature selection algorithms as compared with the single approaches.

Depending on the type of feature selection algorithm, there may be many different ways to create an ensemble feature selection algorithm. Here we describe two most commonly used approaches for creating ensemble filters and ensemble wrappers, respectively.

Ensemble based on data perturbation. The first class of methods is based on data perturbation. This approach has been extensively utilized and studied as can be viewed in the literature [1, 19, 203]. The idea is built on the successful experience in ensemble classification [53] and it has been found to be able to stabilize the feature selection result. For example, a bootstrap sampling procedure can be used for creating an ensemble of filter algorithms, each of which may give a different ranking of genes. The consensus is then obtained through combining those ranking lists. Naturally, besides bootstrap sampling many other data perturbation methods (such as random spacing, etc.) can also be used to create multiple versions of original datasets in the same framework. A schematic illustration of this class of methods is shown in Figure 2.5.

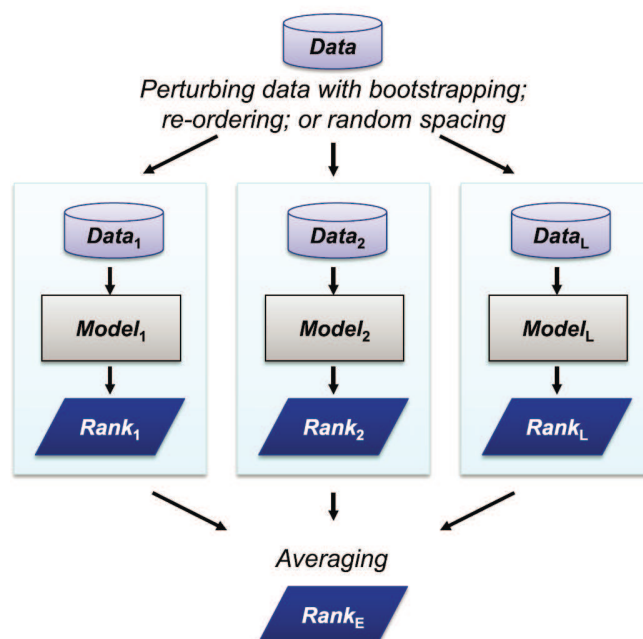


Figure 2.5: Schematic illustration of an ensemble of filters using data perturbation approach.

Ensemble based on different data partitioning. The second approach is based on partitioning the training and testing data differently, which is specifically for wrapper-

based feature selection algorithms. That is, data that are used for building the classification model and data that are used for feature evaluation are partitioned using multiple cross validations (or any other random partitioning procedures). The final feature subset is determined by calculating the frequency of each gene selected from each partitioning. If a gene is selected more than a given threshold, it is then included into the final feature set.

A schematic illustration of this method is shown in Figure 2.6. This method is firstly described in [58] where a forward feature selection (FFS) wrapper and a backward feature elimination (BFE) wrapper are shown to benefit from this ensemble approach.

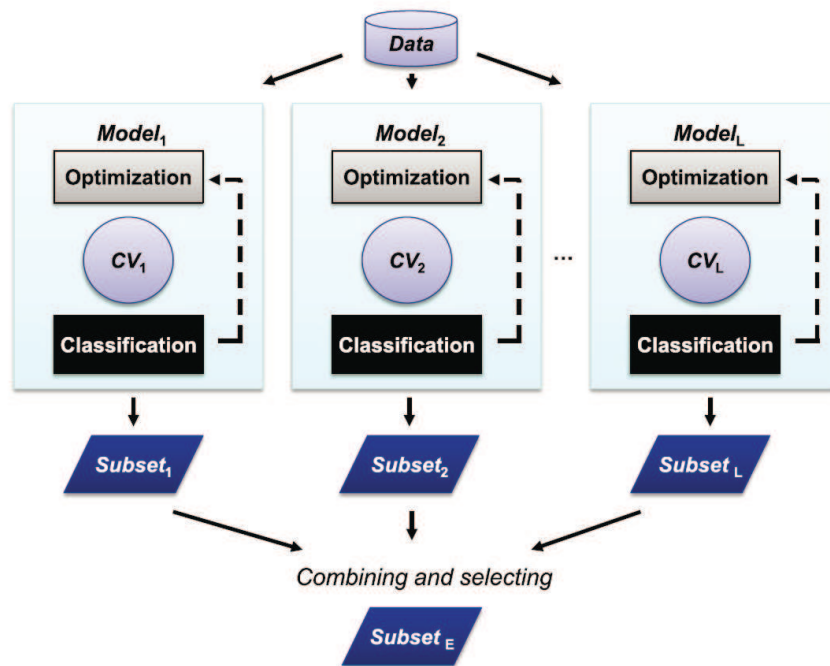


Figure 2.6: Schematic illustration of an ensemble of wrappers using different partitions of an internal cross validation for feature evaluation.

Besides using a different data partitioning, for stochastic optimization algorithms such as GA or particle swarm optimization (PSO), ensemble could also be achieved by using different initializations or different parameter settings. For wrappers such as FFS or BFE, a different starting point in the feature space could result in a different selection result. Generally, bootstrap sampling or other random spacing approaches can also be applied to wrapper algorithms for creating ensembles.

2.1.2.3 Related literatures

In computational and systems biology, ensemble feature selection originated from the use of multiple filters for evaluating genes and proteins in microarray and MS-based proteomics data [172]. This is due to the fact that no single feature selection algorithm can perform optimally on all datasets or under all criteria [206] and the potential existence of multiple subsets of features that have similar discriminant power [112].

The most straightforward approach for creating an ensemble of filters is to borrow the idea of bagging by generating multiple bootstrap samples; each is then used for building a filter. This approach is first adopted by Yu and Chen for m/z feature selection from MS-based proteomics data [210] and then extended by Saeys *et al.* for gene selection from microarray data [1, 171]. Particularly, Saeys *et al.* also considered the stability of the feature selection algorithms and found that an ensemble approach based on bootstrap sampling can significantly improve the stability of the feature selection algorithm and therefore reproducible feature selection results. For the wrapper feature selection algorithm, Li *et al.* proposed a genetic algorithm (GA) based wrapper approach, called GA/kNN, for gene selection from microarray and combining the result through averaging multiple runs with different initializations [115]. The power and the parameters in GA/kNN were further optimized [117] and the algorithm was extended for m/z feature selection from MS-based proteomics data [116] in their subsequent studies.

Besides these data sampling-based approaches, a Bayesian model averaging approach has been applied for ensemble gene selection from microarray data [113, 209], and a distance synthesis scheme for combining the gene selection results from multiple statistics has been introduced by Yang *et al.* for gene selection [206].

Among different ensemble feature selection methods proposed for identifying gene-gene interaction [208, 217], random forests enjoyed the most popularity [42]. This is largely due to its intrinsic ability to take multiple SNPs jointly into consideration in a nonlinear fashion [124]. In addition, random forests can be used easily as an embedded feature evaluation algorithm [26], which is very useful for disease-associated SNP selection.

The initial work of Bureau *et al.* [26] shows the advantage of the random forests regression method in linkage data mapping. Several quantitative trait loci have been successfully identified. The same group [25] then applied the random forests algorithm in the context of the case-control association study. A similar method was also used by

Lunetta *et al.* [121] for complex interaction identification. However, these early studies limited the SNPs under analysis to a relatively small number (30 - 40 SNPs).

Recent studies focus on developing customized random forests algorithms and applying them for gene-gene interaction identification to a much higher data dimension, containing several hundred thousands of candidate SNPs. Specifically, Cheng *et al.* [34] investigated the statistical power of random forests in SNP interaction pair identification. Their algorithm was then applied to analyse the SNP data from the complex disease of age-related macular degeneration (AMD) [103] by using a haplotype-based method for dimension reduction. Meng *et al.* [128] modified random forests to take into account the linkage disequilibrium (LD) information when measuring the importance of SNPs. Jiang *et al.* [91] developed a sequential forward feature selection procedure to improve random forests in gene-gene interaction identification. The random forests algorithm was first used to compute the *Gini index* for a total of 116,204 SNPs from the AMD dataset [103] and then used as a classifier to minimize the classification error by selecting a subset of SNPs in a forward sequential manner with a predefined window size.

2.2 Hybrid algorithms

In artificial intelligence (AI), hybrid algorithms often refer to the effective combination of multiple learning algorithms for solving complex problems [40]. Hybrid algorithms are flexible tools that could be very useful in many bioinformatics applications where the solution involves solving multiple subtasks. Hybrid algorithms could be categorized into (1) tightly coupled in that both algorithms executes in an intertwined way, (2) less tightly coupled in that only the objective function links the two, or (3) loosely coupled in that the algorithms do not have any direct interaction with each other but rather they execute in relative isolation [99]. However, since there are no hard rules dictating which and how algorithms can be combined, one of the difficulties is the discovery of the most appropriate combinations of algorithms for a specific biological problem. One approach is to select different combinations of hybrid algorithms using an agent-based framework [216]. Utilizing domain knowledge has also been demonstrated to be an effective approach for designing specialized and highly tailored systems for answering specific biological questions [137].

Evolutionary-based algorithms [60], such as genetic algorithm (GA), genetic programming, and particle swarm optimization (PSO) to name a few, are popular building blocks for creating hybrid algorithms. Classification algorithms such as support vector machines (SVM) [27] and k -nearest neighbour (k NN) [3] are also commonly used as algorithmic building blocks that when combined with evaluation algorithms form one of the most popular hybrid approach which can be used for feature selection and sample classification. The computation principle of this approach has been validated by Yang and Honavar [202] and it has been subsequently applied in various forms to numerous biological studies. For instance, Li *et al.*'s study in combining GA with k NN (called GA/ k NN) has been very successful in simultaneously performing gene set selection and sample classification for microarray data [117]. This hybrid algorithm has then been extended for protein marker selection and sample classification of mass spectrometry (MS)-based proteomic data [116]. Based on the same framework, many similar hybrid algorithms have been proposed such as (1) the combination of GA with SVM [149] for gene selection and sample classification of microarray data, (2) the combination of PSO with SVM (PSO/SVM) [178] for gene selection and sample classification of microarray data, and (3) the combination of ant colony optimization (ACO) with SVM (ACO/SVM) for m/z feature selection and sample classification of MS-based proteomic data [162].

Another commonly utilized hybrid component is neural networks [75] which is one of the key foundation algorithm in machine learning and data mining. For example, in gene-gene interaction identification from GWA study, a combination of genetic programming with neural networks has been demonstrated to identify disease associated interactions among multiple genes [165]. In gene networks construction, the combination of a neural-genetic hybrid has been successfully applied for reverse engineering from microarray data the gene networks relationship [98]. Several other neural network-based hybrid approaches were also compared by Motsinger-Reif *et al.* [135] for identifying gene-gene interactions.

The optimization of feature space is a key component in disease associated biomarker selection. Several researchers propose a hybrid approach to improve optimization performance and efficiency. For example, Shen *et al.* proposed a hybrid algorithm that combined PSO and tabu search to overcome local optimum in gene selection from microarray [177]. Chuang *et al.* embedded in a GA in PSO for gene selection so as to perform local optimization in each PSO iteration [36].

In contrast to ensemble algorithms, which typically focus on improving the performance of a specific task (e.g. improving classification accuracy of a single classifier), hybrid algorithms can be composed in such a way that multiple subtasks are solved in a modular and parallel manner, and are thus multitasking. Nevertheless, hybrid algorithms can also be designed to improve the performance of a single task. The flexibility and the numerous ways to integrate multiple algorithms have been the key characteristics of hybrid algorithms and their successful applications in computational and systems biology.

Chapter 3

Gene-Gene Interaction Filtering Using Genotype Data

This chapter is based on the following publication:

Pengyi Yang, Joshua W.K. Ho, Jean Yee-Hwa Yang, Bing B. Zhou, Gene-gene interaction filtering with ensemble of filters. BMC Bioinformatics, 12:S10, 2011

3.1 Gene-gene interaction in GWA studies

High-throughput genome-wide association (GWA) studies have become the main approach in exploring the genetic basis of many common complex diseases [190]. Under the assumption that common diseases are associated with common variants, the goal of GWA studies has been to identify a set of single nucleotide polymorphisms (SNPs) that are associated with the complex disease of interest. Typically, this is achieved by adopting a case-control study design that prospectively identifies SNPs that distinguish individuals who have a certain disease (case) from a control population of individuals (control) [88]. However, there are several practical issues when achieving this goal in terms of data analysis. First, to identify true disease associated SNPs from a massive set of candidate SNPs, an accurate SNP selection strategy is of critical importance. However, the accurate identification of disease associated SNPs is hindered by the *curse-of-dimensionality* and the *curse-of-sparsity* [182]. More importantly, it has

become increasingly clear that gene-gene interactions and gene-environment interactions are ubiquitous and fundamental mechanisms for the development of complex diseases [42]. That is, complex diseases such as type 2 diabetes or Alzheimer are unlikely to be explained by any single SNP variant. In contrast, the characterization of gene-gene interactions and gene-environment interactions may be the key to understanding the underlying pathogenesis of these complex diseases [42, 154, 191]. The explanations from the biological perspective are as follows: (1) a SNP in a coding region may cause amino acid substitution, leading to the functional alteration of the protein; (2) a SNP in a promoter region can affect transcriptional regulation, causing the change of the protein expression abundance; and (3) a SNP in an intron region can affect splicing and expression of the gene [192]. All these effects contribute quantitatively and qualitatively to the ubiquity of molecular interactions in biological systems.

For this reason, several methods have been developed to jointly evaluate SNP and environmental factors with the aim of identifying gene-gene and gene-environment interactions that have major implications for complex diseases [136]. These methods analyse genetic factors in a combinatorial manner when applied to the SNP dataset with case and control samples. Therefore, we shall refer to them as *combinatorial methods*. Combinatorial methods will be described in Chapter 4.

The problem of applying combinatorial methods to GWA datasets is that they are commonly computationally intensive and the computation time increases exponentially with the number of SNPs considered. Therefore, it is commonly necessary to perform a filtering step prior to the combinatorial evaluation to remove as many irrelevant SNPs as possible [125]. This is commonly known as the two-step analysis approach as described in [191]. As discussed in a number of recent reviews [42, 131, 191], a good filtering algorithm is of critical importance since, if functional SNPs are removed by the filter, the subsequent combinatorial analysis will be in vain.

3.2 Filtering gene-gene interactions

For categorical data such as genotypes of SNPs, univariate filtering algorithms including χ^2 -test and *odds ratio* are commonly used. However, these methods consider the association between each SNP and the class label independently of other SNPs in the dataset [87]. Therefore they may filter out SNP pairs that have strong interaction effects

but display weak individual association with the phenotype [42]. Recently, new multivariate approaches known as “ReliefF-based” filtering algorithms [123, 131] captured much attention. This family of methods, including ReliefF [166], tuned ReliefF (TuRF) [130], and Spatially Uniform ReliefF (SURF) [71] takes into account dependencies between attributes [166]. This is critical for preserving and prioritizing potential gene-gene interactions in SNP filtering [133].

Although ReliefF-based filtering algorithms have gained much attention and have been applied to several association studies (*e.g.*, [7]; and [158]), we found that filtering results produced by ReliefF and TuRF are sensitive to the order of samples presented in the dataset and may produce unstable SNP ranking results when the order of samples in the dataset is changed.

In this section, we first introduce the ReliefF algorithm and its variant TuRF algorithm. Then we explain why ReliefF-based algorithms are sensitive to the sample order in the dataset and may generate inconsistent SNP ranking when the order of samples is changed. Before we start, let us consider a GWA study consisting of N SNPs and M samples. We denote each SNP in the study as g_j and each sample as s_i where $j = 1 \dots N$ and $i = 1 \dots M$. The aim of the filtering procedure is to produce a ranking score defined as $W(g_j)$, commonly referred to as weight. This score represents the ability of each SNP g_j to separate samples between the case and control groups, and the filtering is done by removing those with low ranking scores according to a pre-defined threshold.

3.2.1 ReliefF algorithm

In the ReliefF algorithm, the weight score of each SNP, $W(g_j)$, is updated at each iteration as follows [123]:

$$W(g_j) = W(g_j) - D(g_j, s_i, h_k)/M + D(g_j, s_i, m_k)/M \quad (3.1)$$

where s_i is the i^{th} sample from the dataset and h_k is the k^{th} nearest neighbour of s_i with the same class label (called “hit”) while m_k is the k^{th} nearest neighbour to s_i with a different class label (called “miss”). This weight updating process is repeated for M samples selected randomly or exhaustively. Therefore, dividing by M keeps the value of $W(g_j)$ in the interval $[-1, 1]$. $D(\cdot)$ is the difference function that calculates the difference between any two samples s_a and s_b for a given gene g :

$$D(g, s_a, s_b) = \begin{cases} 0 & : \text{ if } G(g, s_a) = G(g, s_b) \\ 1 & : \text{ otherwise} \end{cases} \quad (3.2)$$

where $G(\cdot)$ denotes the genotype of SNP g for sample s , which can take the value of aa (homozygotes of recessive alleles), Aa (heterozygotes), or AA (homozygotes of dominant alleles). The nearest neighbours to a sample are determined by the distance function, $MD(\cdot)$, between the pairs of samples (denoted as s_a and s_b) which is also based on the difference function (Equation 3.2):

$$MD(s_a, s_b) = \sum_{j=1}^N D(g_j, s_a, s_b) \quad (3.3)$$

Using pseudocode, we can outline the ReliefF algorithm in **Algorithm 1**.

Algorithm 1 ReliefF

```

1: for  $j=1$  to  $N$  do
2:   initiate( $W(g_j)$ );
3: end for
4: for  $i=1$  to  $M$  do
5:    $s_i = \text{randomSelect}(\text{sampleSize})$ ;
6:    $\mathcal{H} = \text{findHitNeighbours}(s_i, K)$ ; ( $h_1 \dots h_K \in \mathcal{H}$ )
7:    $\mathcal{M} = \text{findMissNeighbours}(s_i, K)$ ; ( $m_1 \dots m_K \in \mathcal{M}$ )
8:   for  $j=1$  to  $N$  do
9:     for  $k=1$  to  $K$  do
10:       $W(g_j) = W(g_j) - D(g_j, s_i, h_k)/M + D(g_j, s_i, m_k)/M$ 
11:     end for
12:   end for
13: end for

```

The ReliefF algorithm calculates the distance between different samples using the genotype information of all SNPs. However, such a procedure is sensitive to noise in the dataset.

3.2.2 Tuned ReliefF (TuRF)

Tuned ReliefF (TuRF) [130] aims to improve the performance of the ReliefF algorithm in SNP filtering by adding an iterative component. The signal-to-noise ratio is enhanced significantly by recursively removing the low-ranked SNPs in each iteration. Specifically, if the number of iterations of this algorithm is set to R , it removes the N/R lowest

ranking (*i.e.*, least discriminative) SNPs in each iteration, where N is the total number of SNPs. The pseudocode for TuRF is shown in **Algorithm 2**.

Algorithm 2 TuRF

```

1: for  $i = 1$  to  $R$  do
2:   apply ReliefF( $M, K$ );
3:   sortSNP();
4:   removeLowSNP( $N/R$ );
5: end for
6: return last ReliefF estimate for each SNP

```

3.2.3 Instability of ReliefF-based algorithm

We found that the ReliefF algorithm is sensitive to the order of samples used to calculate the SNP ranking score (Eq. 3.1). That is, running these algorithms on the same dataset with the order of the samples permuted (while maintaining the sample-class label association), leads to different SNP ranking results.

A close investigation of the ReliefF algorithm found that such a sample order dependency is related to an intrinsic tie-breaking procedure inherited in the k -nearest neighbours (k NN) routine. It causes a partial utilization of neighbour information, leading ReliefF and TuRF to generate unstable results. Specifically, such a sample order dependency is related to the assignment of “hit” and “miss” nearest neighbours of each sample (lines 6 and 7 of **Algorithm 1**). Since K nearest neighbours are calculated by comparing the distance between each sample in the dataset (using all the SNP attributes) and the target sample (s_i in **Algorithm 1**), a tie occurs when more than K samples have a distance equal or less than the K^{th} nearest neighbour of s_i . We can show that the sample order dependency can be caused by using any tie breaking procedure that forces exactly K samples out of all possible candidates to be the nearest neighbours of s_i , which causes a different assignment of “hit” and “miss” of nearest neighbours when the sample order is permuted.

3.3 Ensemble of filters for gene-gene interaction filtering

As described in Section 2.1.2, the ensemble feature selection approach has been successfully used to reduce instability. Here we perturb the original dataset by randomly permuting the sample orders. The aim is to take advantage of the different SNP ranking results generated from the perturbed version of the original dataset by aggregating multiple SNP rankings.

From our analysis of the aforementioned tie-breaking problem, it is clear that a different set of samples may be assigned to be a sample's nearest neighbours. Therefore, the result of a single run of ReliefF utilizes only partial information embedded in the full set of the nearest neighbours. In other words, the results from multiple runs of ReliefF using the dataset with permuted sample order should contain complementary information about how well each set of SNPs can discriminate between the two classes (case vs. control). In this sense, we can potentially harness the “diversity” of ranking results from multiple executions with permuted sample order using an ensemble-based method to produce more stable and accurate SNP ranking results.

Formally, our ensemble of ReliefF (called ReliefF-E) produces L copies of the input SNP dataset by randomly permuting the order of the samples, and invoking ReliefF to calculate a ranking score for each SNP g_j in each of these permuted datasets, called $W_l(g_j)$ for iteration l , ($l = 1, \dots, L$). An ensemble ranking score of each gene $W_{ensemble}(g_j)$ is defined to be the mean of the individual ranking score of each SNP:

$$W_{ensemble}(g_j) = \frac{\sum_{l=1}^L W_l(g_j)}{L} \quad (3.4)$$

Similarly, the ensemble of TuRF (called TuRF-E) performs multiple runs of TuRF, and aggregates the ranking scores of each SNP produced in each iteration of TuRF using Equation 3.4. Schematically, the ensemble of filters can be illustrated as in Figure 3.1, where the original dataset D is randomly re-ordered L times to create multiple copies of perturbed datasets. Then, each perturbed dataset is used for filtering (F_i , ($i = 1 \dots L$)) and a corresponding ranking is obtained R_i . The final ranking is obtained by combining each individual ranking, and re-ranking the SNPs using Equation 3.4.

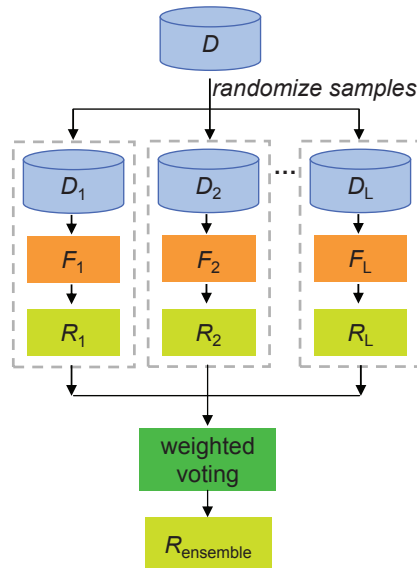


Figure 3.1: A schematic illustration of ensemble of filters using random sample re-ordering.

3.4 Experiment on simulation and real-world GWA data

To illustrate this effect, we used both a set of simulation datasets generated by [132] and a real world GWA dataset for our demonstration. These simulation datasets were generated using different genetic models (different heritability and sample size) and each model randomly simulated the genotype of 1000 SNPs across all the samples except for one functional SNP-SNP interaction pair denoted as “X0” and “X1” in the dataset. These datasets are summarized in Table 3.1.

Table 3.1: Summary of simulation datasets. Each model contains 100 datasets.

Model	SNP size	Sample size	Heritability
Epistatic_400_0.05	1000	case: 200; control: 200	0.05
Epistatic_400_0.1	1000	case: 200; control: 200	0.1
Epistatic_400_0.2	1000	case: 200; control: 200	0.2
Epistatic_400_0.3	1000	case: 200; control: 200	0.3
Epistatic_800_0.05	1000	case: 400; control: 400	0.05
Epistatic_800_0.1	1000	case: 400; control: 400	0.1
Epistatic_800_0.2	1000	case: 400; control: 400	0.2
Epistatic_800_0.3	1000	case: 400; control: 400	0.3

A GWA dataset generated from case-control design of age-related macular degeneration (AMD) samples [103] is also used to illustrate the sample order dependency of ReliefF and TuRF when applied to real SNP datasets. The AMD dataset contains 96 cases and 50 controls, with the genotype of 116,212 SNPs for each sample.

3.4.1 The effect of the sample order dependency

Figure 3.2a shows the Pearson correlation of the ranking of the SNPs in two separate runs of ReliefF and TuRF using a dataset containing 1000 SNPs and 400 samples (200 controls and 200 cases), respectively. Figure 3.2b is the result of the same analysis applied to a simulation dataset containing 800 samples. It is clear that both ReliefF and TuRF algorithms are sensitive to the order of samples presented in datasets, causing the rank of each SNP to be inconsistent between the original dataset and the randomly re-ordered dataset. While such an inconsistency is relatively small for the ReliefF algorithm, the problem is much more severe in TuRF. The Pearson correlation coefficient of two runs of TuRF is $r = 0.43$ for the dataset with 400 samples and $r = 0.36$ for the dataset with 800 samples.

By using the aggregation procedure (by aggregating ranking scores from 50 runs of the algorithms; see Section 3.4.3 for details), we are able to stabilize the ranking results of both ReliefF and TuRF. Especially, TuRF-E can significantly increase the stability of the SNP ranking results of TuRF, with $r = 0.97$ for the dataset with 400 samples and $r = 0.95$ for the dataset with 800 samples.

Similar results were obtained when the AMD dataset was analysed (Figure 3.2c). The results illustrate that the sample order instability is indeed a problem in analysing real biological datasets with ReliefF and TuRF. The use of ensemble of filters increases stability and this is evident from the increase of the ranking correlation to $r = 0.99$ for ReliefF and $r = 0.98$ for TuRF.

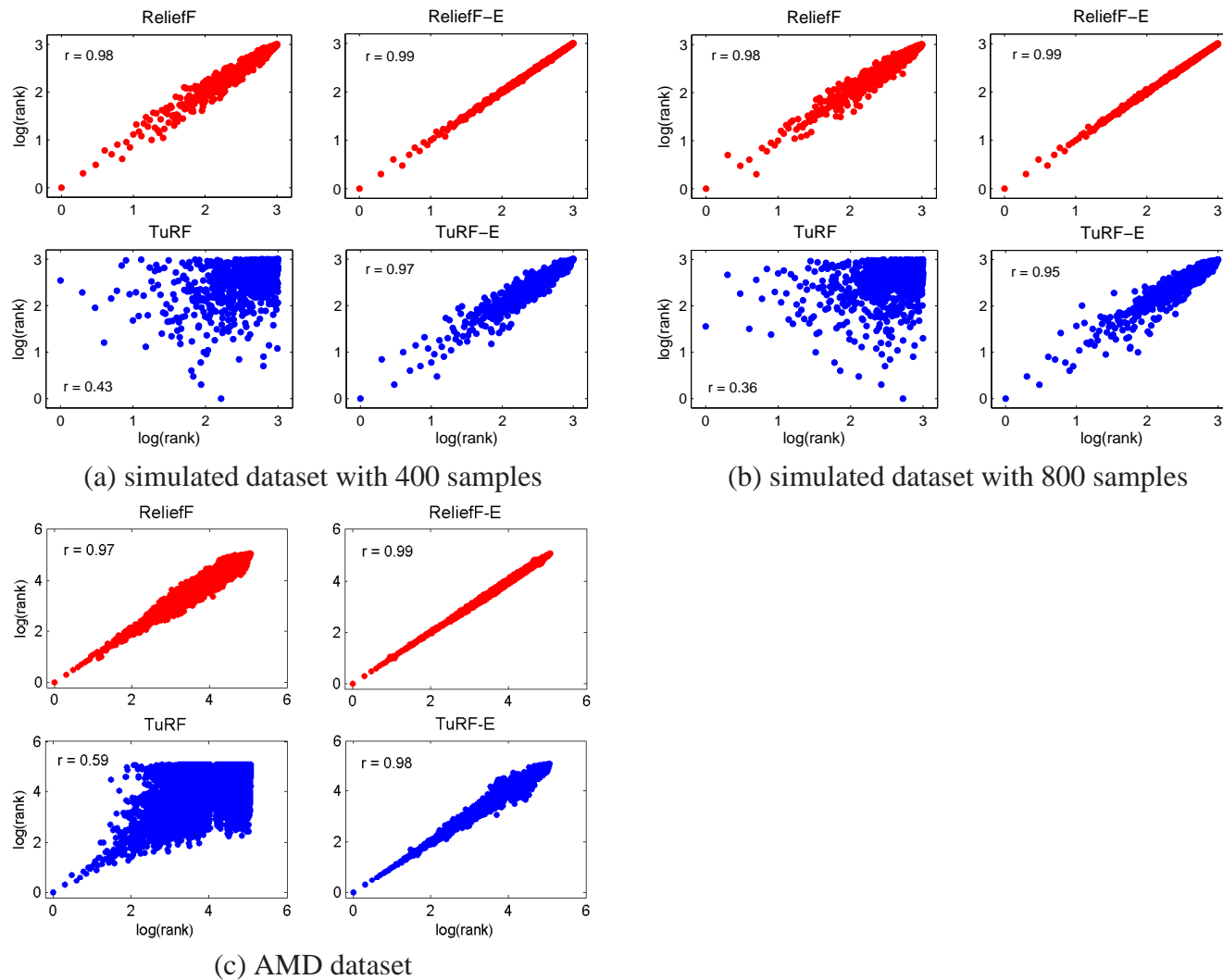


Figure 3.2: The correlation between SNP ranking (\log_{10} transformed) generated by two runs of ReliefF, TuRF, ReliefF-E, and TuRF-E using simulation datasets (400 and 800 samples) and the AMD dataset in which each run used a different sample order.

3.4.2 The origin of the sample order dependency

To verify whether the sample order dependency is indeed caused by tie-breaking, we modified and recompiled the source code of `mdr-2.0_beta_6.zip` (downloaded from <http://sourceforge.net/projects/mdr/>) to report when a tie-breaking happens. Figure 3.3 shows how many times a tie-breaking case happens when using ReliefF and TuRF for filtering SNPs in the AMD dataset, respectively. It is evident that when using TuRF for SNP filtering, many more tie-breaking cases happen. This explains why the SNP ranking results from re-ordered datasets using TuRF is far more unstable compared to those using ReliefF.

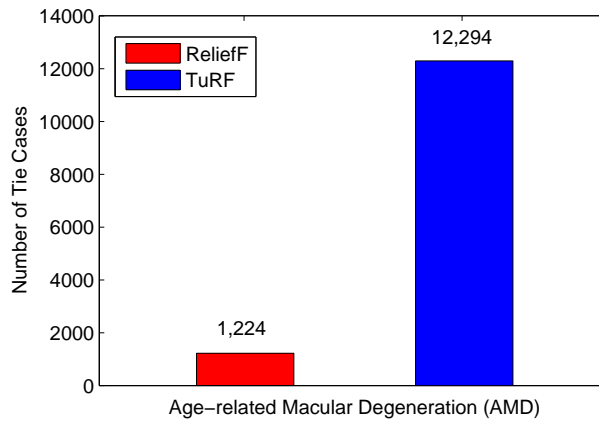


Figure 3.3: The number of times a tie breaking case happens when using ReliefF and TuRF for filtering SNPs in the AMD dataset.

We also modified the source code of `mdr-2.0_beta_6.zip` to report the tie-causing samples and remove them from the dataset. After removing all tie-causing samples, we were able to obtain completely reproducible ranking results (*i.e.*, $r = 1$) with both ReliefF and TuRF (Figure 3.4). Hence, we pinpoint the origin of sample order dependency in ReliefF and TuRF algorithms. However, resolving sample order dependency using this approach requires aggressive removal of a large number of samples, which inevitably reduces the algorithms' power to filter functional SNP pairs.

One tempting way to solve such a sample order dependency is to use a randomize procedure to select a sample randomly when a tie occurs. However, our experiments indicate that such a procedure does not increase the correlation (data not shown). In fact, any tie-breaking procedure that chooses one sample out of all valid candidate samples will necessarily produce instability in its resulting ranking score.

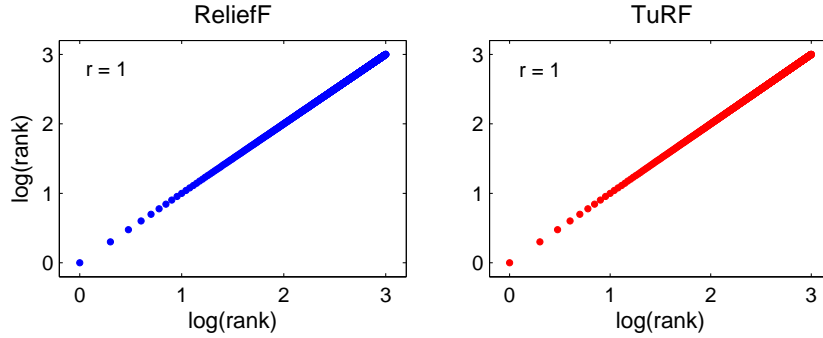


Figure 3.4: The correlation between the SNP rankings (\log_{10} transformed) of two separate runs using datasets with tie-causing samples removed.

Another way to solve such a sample order dependency can be achieved by defining nearest neighbours to a sample as the ones that are within a certain distance threshold of the target sample. A recently developed variant algorithm of ReliefF called SURF (Spatially Uniform ReliefF; [71]) employed this idea. However, by doing so, the algorithm will rely directly on a predefined threshold for nearest neighbours selection, which may negatively affect the result given the sample sparsity in high-dimensional space. Therefore, such an approach lacks the robustness of the rank based k NN criteria. Our study (Section 3.4.4) confirmed that SURF does not fully recover the SNP filtering capacity. As discussed later in this paper, our aggregation approach, which relies on sample ranking instead of direct thresholding, gives consistently better results.

3.4.3 Determination of ensemble size

An important parameter in any aggregation method is the aggregation size. This is the number of times an algorithm is repeatedly applied on a dataset with reordered samples. It is important to estimate the minimum aggregation size that is sufficient to reduce sample order dependency. We estimate this value via repeating the correlation analysis on TuRF-E with an aggregation size of 10, 20, 30, 40, and 50 using the simulated datasets with 400 samples and 800 samples (Figure 3.5). It is apparent that the increase of the correlation in two separate runs using the original and the randomly re-ordered datasets plateaus at around an aggregation size of 40 for both datasets, and there is only minor improvement when employing more than 50 runs. Therefore, the aggregation size of 50 is used in all our subsequent experiments.

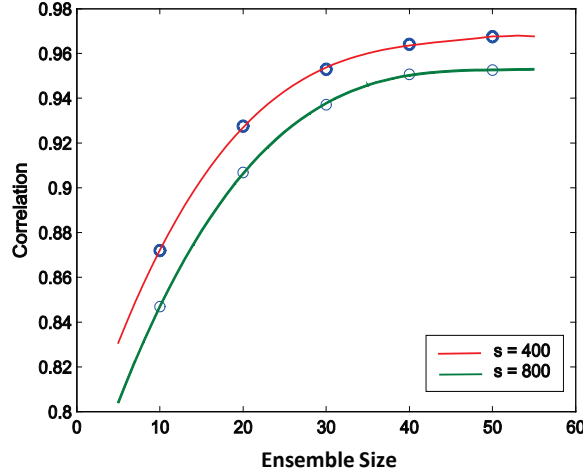


Figure 3.5: The correlation between the SNP rankings with respect to different aggregation size of TuRF, using simulation datasets with 400 samples ($s=400$) and 800 samples ($s=800$).

3.4.4 Ensemble approach to improve success rate in SNP filtering

One motivation for using the proposed aggregation approach is to gain a more informative SNP scoring. Therefore, we investigated whether our aggregation scheme can improve the ability of ReliefF and TuRF to retain functional SNP pairs in SNP filtering. Figure 3.6 shows the trend of the success rate of each filtering algorithm across percentile 1 to 50 (*i.e.*, 10-500 top ranking SNPs) using simulated datasets with 400 samples and 800 samples respectively. Table 3.2 shows the average cumulative success rate of these algorithms on the same set of simulated datasets. We found that TuRF-E performs the best in all cases examined in our experiments regardless of sample size and heritability of the simulated datasets. ReliefF-E and ReliefF have similar performance in terms of success rate, while traditional univariate filters such as χ^2 -test and odds ratio give the lowest success rates. The superiority of TuRF-E is particularly noticeable in datasets simulated with low heritability or a small number of samples. This implies that TuRF-E is applicable in even these “challenging” cases where other ReliefF-based algorithms fail to achieve high enough success rates.

It is found that ReliefF-E does not exhibit much improvement on ReliefF whereas TuRF-E achieves significant improvement on TuRF. This is probably due to the fact that the TuRF algorithm executes ReliefF multiple times while removing low ranking

Table 3.2: Average cumulative success rate from percentile 1 to 50 using the simulated datasets (400 and 800 samples). The best algorithm with the highest average cumulative success rate in each dataset is shown in **bold**.

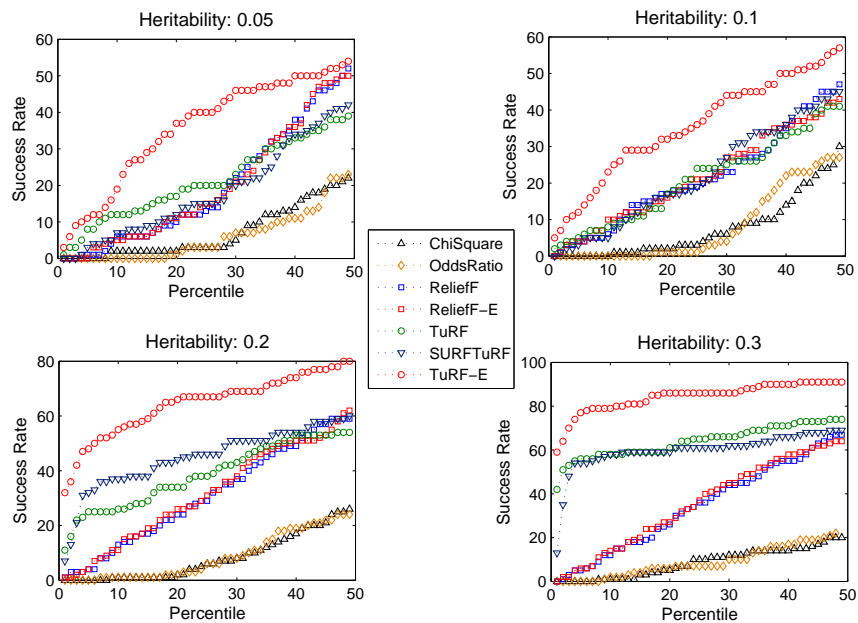
Methods	Heritability = 0.05	Heritability = 0.1	Heritability = 0.2	Heritability = 0.3
Simulated dataset with 400 samples				
χ^2 -test	6.92	7.20	8.06	8.51
Odds Ratio	5.86	7.84	8.43	8.58
ReliefF	18.96±0.38	20.93±0.47	30.35±0.28	33.98±0.31
ReliefF-E	19.27±0.17	21.22±0.14	30.92±0.24	34.76±0.26
TuRF	22.11±1.34	24.59±2.53	42.27±3.41	61.37±1.58
SURFTuRF	18.12	21.88	44.92	59.88
TuRF-E	35.23±0.37	35.85±0.82	63.55±0.93	84.71±0.25
Simulated dataset with 800 samples				
χ^2 -test	7.73	8.53	9.61	7.84
Odds Ratio	8.53	9.86	9.92	6.61
ReliefF	24.37±0.52	25.11±0.80	44.23±0.86	54.40±0.75
ReliefF-E	25.59±0.63	25.85±0.28	44.81±0.36	56.91±0.46
TuRF	33.20±2.11	39.99±2.04	78.64±3.14	91.93±1.13
SURFTuRF	41.20	50.82	96.27	99.86
TuRF-E	61.59±0.58	65.75±1.09	96.69±0.26	99.96±0.21

SNPs in each iteration. Therefore, an aggregation approach could gain more information in each iteration. It is also observed that SURFTuRF does not improve on TuRF in analysing datasets of 400 samples. This is consistent with our hypothesis that a predefined distance threshold may be sensitive to a high SNP-to-sample ratio (thus, high-dimensionality).

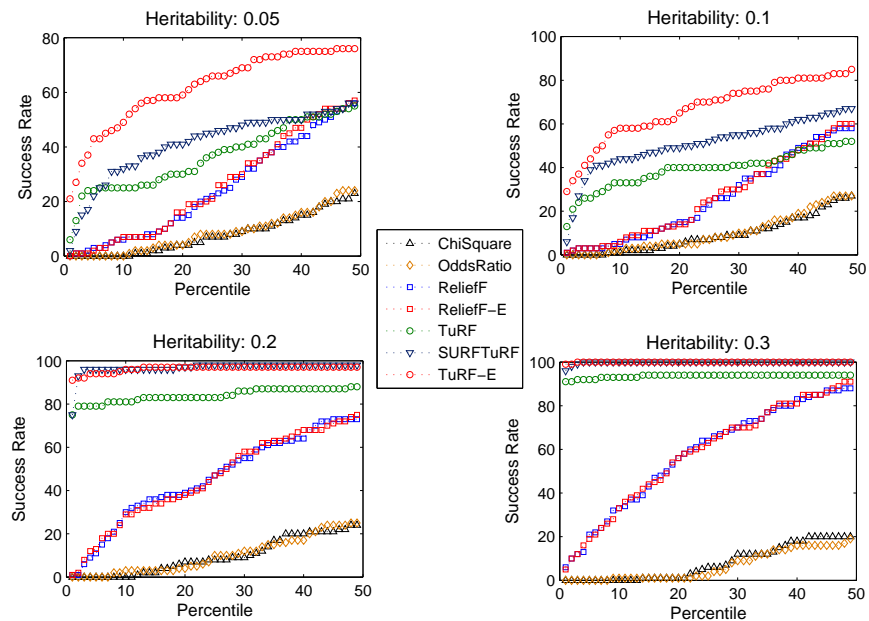
We further investigated whether TuRF-E is simply “averaging” out the detection ability in different runs of TuRF. Figure 3.7 shows the average cumulative success rates of 50 runs of TuRF on a simulated dataset (sample size = 400, heritability = 0.05) where a different sample order is used in each run, and the corresponding average cumulative success rate of their aggregate version (TuRF-E). It is clear that the aggregate SNP ranking result is significantly better than any single run of TuRF. This implies that our aggregation algorithm is indeed able to make use of the information embedded in multiple runs of TuRF to improve its detection ability, verifying our motivation for using an aggregation approach.

3.5 Summary

The field of gene-gene and gene-environment interaction identification from GWA studies is still young and rapidly developing. One of the main challenges in identification of



(a) Simulated data with 400 samples



(b) Simulated data with 800 samples

Figure 3.6: Success rate for retaining a functional SNP pair in simulated datasets with (a) 400 samples and (b) 800 samples.

such interaction relationships is computational efficiency since in the worst case an exponentially large number of SNP combinations need to be evaluated. As discussed by a

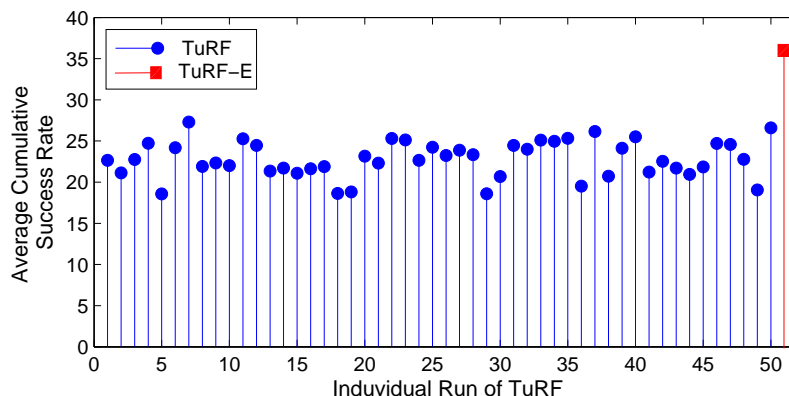


Figure 3.7: Comparison of average cumulative success rate of 50 individual runs of TuRF (shown in a blue circle) and their aggregate results (TuRF-E; shown in a red square) using a simulated dataset with 400 samples (heritability = 0.05).

number of authors [42, 131, 191], effective SNP filtering can greatly reduce the computational burden of the subsequent combinatorial evaluation by removing a large portion of noise. The main advantage of using ReliefF-based algorithms for SNP filtering is that they can detect conditional dependencies between attributes [166]. Furthermore, they are computationally efficient. A good implementation of TuRF can analyse a GWAS dataset with up to a few hundred samples in the order of minutes. Such computational efficiency, coupled with its intrinsic ability in detecting SNP dependencies, has led to its increasing wide-spread applications.

Through analysing the ReliefF-based algorithms, we discovered a previously unknown anomaly in both ReliefF and TuRF. We show these two popular filtering algorithms are sensitive to sample ordering, and therefore, give unstable and suboptimal SNP ranking in different runs when the sample order is permuted. Using a simple ensemble procedure based on the general theory of ensemble learning, we can vastly improve the stability and reliability of the SNP ranking generated by these algorithms. It is indeed quite remarkable that such a simple modification, which is guided by the theory of ensemble learning, can yield such a vast improvement in the final result. The fact that TuRF-E is better than the state-of-the-art SURFTuRF algorithm indicates that preserving the k NN rank-based routine is indeed a good idea.

ReliefF-based algorithms are also used to perform feature selection tasks for a range of machine learning problems including gene selection in microarray analysis. This implies our findings are not limited to the field of gene-gene interaction identification in GWA studies, and may have relevance to the broader machine learning community.

Although we recognize that the sample order sensitivity problem is of less relevance to continuous datasets since tie-breaking is less likely to occur, the potential problem caused by tie-breaking in a k NN procedure is still noteworthy in the development of new algorithms.

Our work indicates that new algorithms should be validated against a range of criteria. Many bioinformatics algorithms have been developed to perform such filtering tasks. These algorithms are mostly assessed and compared based on their objective, in our situation, how well a filtering algorithm can retain functional SNP pairs. However, much less focus has been placed on analysing whether the results generated by a SNP filtering algorithm satisfy a set of desirable properties. The sample order dependency property in this paper is one such example, as it is not natural to expect the SNP ranking to change due to reordering the samples in a dataset. In fact, the importance of validating a bioinformatics algorithm and its software implementation is increasingly being recognized [32], and we believe that systematically validating an algorithm against a range of desirable properties of its behaviour is becoming more important as biological interpretations are increasingly drawn from results produced by bioinformatics programs.

3.6 Software availability

The TuRF-E package is freely available from:

<http://code.google.com/p/ensemble-of-filters>

Chapter 4

Gene-Gene Interaction Identification Using Genotype Data

This chapter is based on the following publication:

Pengyi Yang, Joshua W.K. Ho, Albert Y. Zomaya, Bing B. Zhou, A genetic ensemble approach for gene-gene interaction identification. BMC Bioinformatics, 11:524, 2010

4.1 Combinatorial testing for gene-gene interaction identification from genome-wide association studies

As mentioned in Section 3.1, current opinion is that the development of complex diseases is inherently multifactorial governed by multiple genetic and environmental factors and the interactions among them. The fast development of the genotyping technologies has empowered us to study genetic and environmental interactions on a genome-wide scale. However, data analysis is swamped by the large amount of data and high-dimensionality. Methods for gene-gene interaction filtering that we described in Chapter 3 are key computational techniques to reduce the variables to a manageable amount for combinatorial testing.

A number of *combinatorial methods* have been developed recently. These include logistic regression-based approaches [146] random forests-based algorithms [25, 34], and nonparametric methods like Polymorphism Interaction Analysis (PIA) [127], Multifactor Dimensionality Reduction (MDR) [76], and Combinatorial Partitioning Method

(CPM) [138]. However, there is no one-size-fits-all method for the detection and characterization of gene-gene interaction relationships in GWA studies. Several comparison and evaluation studies suggested that applying a combination of multiple complementary algorithms, each having its own strength, could be the most effective strategy to increase the chance of a successful analysis [22, 83, 136].

Here we attempt to address the problem from an alternative perspective by converting the issue into a combinatorial feature selection problem. From the data mining perspective, a sample from a SNP dataset of an association study is described as a SNP feature set of the form $\mathbf{f}_i = \{g_1, g_2, \dots, g_n\}$, ($i = 1, \dots, m$) where each SNP, g_i , is a categorical variable that can take the value of 0, 1, and 2 for genotypes of aa , Aa , or AA at this locus, and m is the number of samples in the dataset. The dataset can, therefore, be described as an $m \times n$ matrix $\mathbf{D}_{mn} = \{(\mathbf{f}_1, y_1), (\mathbf{f}_2, y_2), \dots, (\mathbf{f}_m, y_m)\}$, where y_i is the class label of the i^{th} sample. The assumption is that a gene-gene interaction exists if it helps in discriminating the disease status. To evaluate the discrimination power of a set of SNPs jointly, we apply the following two steps. (1) Generating a reduced SNP feature set $\mathbf{f}'_i = \{g_1, g_2, \dots, g_d\}$, ($\mathbf{f}'_i \subset \mathbf{f}_i$) in a combinatorial manner which restrains the dataset matrix into $\mathbf{D}_{md} = \{(\mathbf{f}'_1, y_1), (\mathbf{f}'_2, y_2), \dots, (\mathbf{f}'_m, y_m)\}$. A key observation is that feature selection algorithms that evaluate SNPs individually are not appropriate since they cannot capture the associations among multiple SNPs. (2) Creating classification hypothesis h using an inductive algorithm, and evaluating the quality of the trained model using criteria such as accuracy, sensitivity, and/or specificity with an independent test set.

Without loss of generality, we simplify the notation as f to denote applying a SNP subset to restrain the SNP dataset \mathbf{D}_{mn} . If a SNP combination f yields a lower misclassification rate than others, we shall consider that it possibly contains SNPs with main effects or SNP-SNP interactions with major implications. We now have two challenging problems for the SNP interaction identification. The first challenge is to generate SNP combinations efficiently since the number of SNP combinations grows exponentially with the number of SNPs, and it is not feasible to evaluate all possible combinations exhaustively. The second challenge is to determine which inductive algorithm should be applied for the goodness test of SNP combinations. To tackle the first problem, we shall apply genetic algorithm (GA) since it has been demonstrated to be one of the most successful wrapper algorithms in feature selection from high-dimensional data [105, 106]. Furthermore, its intrinsic ability in capturing nonlinear relationships [193] is valuable for modelling various nonadditive interactions. With regard to the second

problem, there is no guiding principle on which inductive algorithms are preferable for identification of multiple loci interaction relationships. However, a promising solution is to employ multiple classifiers and then to integrate/balance the evaluation results from these classifiers [34]. The key issue in applying this method is that the individual classifiers used for integration should be able to capture multiple SNP interactions that commonly have nonlinear relationships. This may be achieved by using appropriate nonlinear classifiers.

As mentioned in Section 2.1.1, the rationale of using multiple classifiers is that, suppose a given classifier i generates a hypothesis space \mathcal{H}_i for sample classification, if the number of training samples m is large enough to characterize the real hypothesis f (in this context, f is the set of disease-associated SNPs and SNP combinations) and the data are noise-free, the hypothesis space generated by i should be able to converge to f through training. However, since the number of training samples is often far too small compared to the size of the hypothesis space, which increases exponentially with the size of the features (SNPs), the number of hypotheses a classifier can fit to the available data is often very large. One effective way to constrain the hypothesis space is to apply multiple classifiers, each with a different hypothesis-generating mechanism. If each classifier fulfils the criteria of being accurate and diverse [24], it can be shown that one is able to reduce the hypothesis space to better capture the real hypothesis f by combining them with an appropriate integration strategy [53]. By combining GA with multiple classifiers, we obtain a hybrid algorithm (called *genetic ensemble* or GE) for gene-gene interaction identification that is able to identify different sizes of interactions in parallel.

One other motivation for developing alternative methods for SNP-SNP interaction identification is in hope that different algorithms may complement each other to increase the overall chance of identifying true interaction relationships. Therefore, it is important to evaluate the degree of complementarity of multiple algorithms for SNP-SNP interaction identification. Specifically, based on the notion of *double fault* [170], we propose a formula for calculating the co-occurrence of mis-identification that gives an indication of the degree of complementarity between two different algorithms. Accordingly, the joint identification of using multiple algorithms is derived.

4.2 Gene-gene interaction identification using genetic ensemble hybrid algorithm

As illustrated in Figure 4.1, the GE approach is applied to SNP selection repeatedly. In each run, randomly generated SNP subsets are fed into a committee of multiple classifiers for goodness evaluation. Two classifier integration strategies, namely *blocking* and *voting*, and a diversity-promoting method called *double fault* statistic are employed to guide the optimization process.

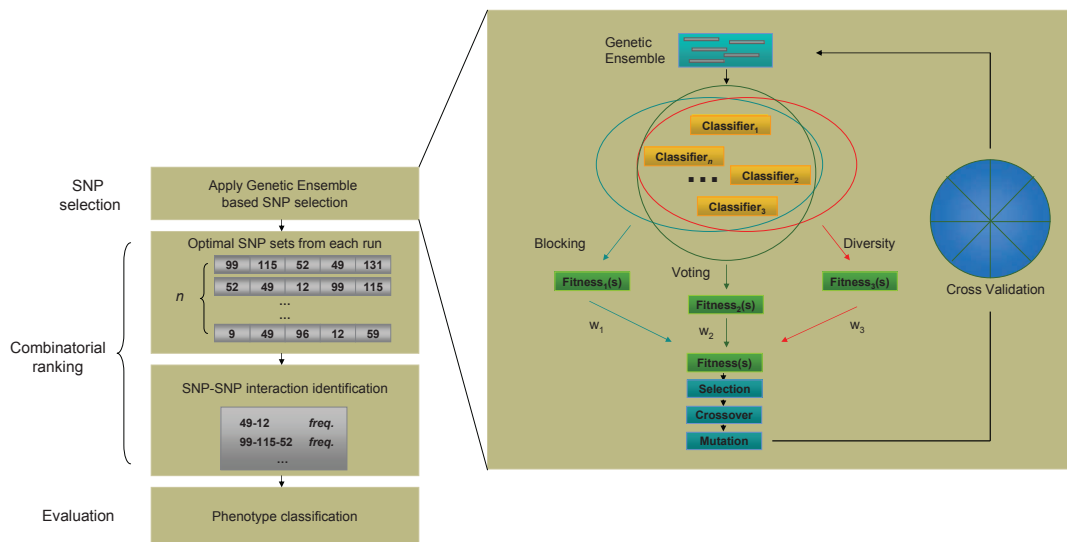


Figure 4.1: Genetic ensemble system. Multiple classifiers are integrated for gene-gene and gene-environment interaction identification. Genetic algorithm is employed to select SNP subsets that have been identified to have potential gene-gene and gene-environment interaction information.

When the evaluation of a SNP subset is done, the evaluation feedbacks of this SNP subset are combined through a given set of “weight” values and sent back to GA as the overall fitness of this SNP subset. After the whole population of GA is evaluated, selection, crossover and mutation are conducted and the next generation begins. A near optimal SNP subset is produced and collected when a set of termination conditions are met. The entire GA procedure is repeated (with different seeds for random initialization) n times ($n = 30$ in our experiments) to generate n best SNP subsets. These SNP subsets are then analysed to identify frequently occurring SNP-pairs, SNP-triplets, and higher-order SNP combinations.

For SNP interaction identification, a combinatorial ranking is applied to the n s-elected SNP subsets. Each possible SNP combination is then given an identification frequency score (the number of times it appears divided by the total number of iterations n). For example, if the SNP combination $\{snp_1, snp_2\}$ appears in 25 out of 30 iterations, then its identification frequency score is $25/30=0.833$. Two alternative criteria can be used to decide whether a SNP combination should be called or not. The first criterion is to set a frequency score cut-off, say 0.8, and call all SNP combinations with a frequency score higher than this cut-off as functional SNP combinations. The second criterion is to set a cut-off rank, and call all SNP combinations equal to or higher than that rank as functional SNP combinations. As will be demonstrated in subsequent sections, the choice between these two criteria is likely to be a balance between detection power and false discovery rate.

4.2.1 Genetic component

The number of SNPs considered by the genetic ensemble algorithm for potential interaction detection, ranges from the lower bound of 2 to the upper bound of d , where d is the “chromosome” size of GA. The size of the GA chromosome has two implications. Firstly, it controls the number of factors we can identify. For example, if the size of $d = 15$ is used, we can identify from 2-factor up to 15-factor interactions in parallel. Secondly, d also influences the size of the combinatorial space to be explored. It is a trade-off between the computational time and the combinatorial space to be searched. Therefore, for different SNP sizes (that is, the number of SNPs in the dataset), we shall use different sizes of d accordingly. Similar to the size of GA chromosome, the population size p and the generation of GA g are also specified according to the SNP size in the dataset. In our implementation of the GE algorithm, the parameters d , p , and g can be specified by users. The default values of these parameters are chosen empirically such that they work well in a range of datasets.

For the GA selection operation, we employ the tournament selection method as it allows control of convergence speed. Specifically, the tournament selection size, denoted as t , is dependent on the size of the population, varying from 3 to 7. The measure for determining the winner is as follows:

$$Winner = \arg \max_{s \in p} fitness(R_i(p)) \quad (i = 1, 2, \dots, t) \quad (4.1)$$

where $R_i(\cdot)$ is the random selection function which randomly selects gene subset s from the GA population p , t is the tournament size, and $fitness(\cdot)$ determines the overall fitness of the randomly selected gene subset. Single point crossover is adopted with the probability of 0.7. In order to allow pair mutations, we implemented a multi-mutation strategy; that is, when a single mutation occurs (configured with the probability of 0.1) on a chromosome, another single point mutation may occur on the same chromosome with a probability of 0.25 and so on. The chromosome coding scheme is to assign an *id* to each SNP in the dataset, and to represent the chromosome as a string of SNP *ids* that specify a selected SNP subset. For each position on a chromosome, it could be a SNP *id* or a “0”, which specifies an empty position. Therefore, different sizes of SNP combinations are explored in a single GA population in parallel. Table 4.1 summarizes the parameter settings.

Table 4.1: Genetic algorithm parameter settings.

Parameter	Value
Chromosome size	15-25
Population size	40-340
Termination generation	8-20
Selector	Tournament selection (3-7)
Crossover	Single point (0.7)
Mutation	Multiple points (0.1 & 0.25)

The fitness of GA is defined as follows:

$$fitness(s) = w_1 \times fitness_B(s) + w_2 \times fitness_V(s) + w_3 \times fitness_D(s) \quad (4.2)$$

where s denotes a SNP combination under evaluation. The functions $fitness_B(s)$, $fitness_V(s)$ and $fitness_D(s)$ denote the fitness of a SNP combination s as evaluated by the *blocking*, *voting* and *double fault* diversity measures, respectively. A complexity regularization procedure is implemented in the GE algorithm to favour shorter SNP combinations if two SNP combinations have the same fitness value. The computation details of each component of the fitness function are described in the next section.

4.2.2 Integration functions

4.2.2.1 Blocking

Our first integration function is *blocking*. It is a statistical strategy that creates similar conditions to compare random configurations in order to discriminate the real differences from differences caused by fluctuation and noise [18]. Suppose a total of L classification algorithms, each having a different hypothesis denoted as h_i^s , ($i = 1, \dots, L$), are used to classify the data using a SNP subset s . The fitness function determined by *blocking* integration strategy is as follows:

$$fitness_B(s) = \sum_{i=1}^L BC(p(\mathbf{t}|h_i^s, \mathbf{D}), \mathbf{y}) \quad (4.3)$$

where \mathbf{y} is the class label vector of the test dataset \mathbf{D} , function $p(\cdot)$ predicts/classifies samples in \mathbf{D} as \mathbf{t} using h_i^s , and $BC(\cdot)$ is the balanced classification accuracy devised to deal with the dataset with an imbalanced class distribution. In the binary classification, it is the area under ROC curve (AUC) [29], which can be approximated as follows:

$$BC(p(\mathbf{t}|h_i^s, \mathbf{D}), \mathbf{y}) = \frac{Se + Sp}{2} \quad (4.4)$$

$$Se = \frac{N_{TP}}{N_{case}} \times 100, \quad Sp = \frac{N_{TN}}{N_{control}} \times 100 \quad (4.5)$$

where Se is the sensitivity value calculated as the percentage of the number of true positive classifications (N_{TP}) divided by the number of cases (N_{case}). Sp is the specificity value calculated as the percentage of the number of true negative classifications (N_{TN}) divided by the number of controls ($N_{control}$). Such a balanced classification accuracy measure can accommodate the situation in which the dataset contains an imbalanced class distribution of cases and controls [194].

The idea of applying this strategy for classifier integration in SNP selection is that by using more classifiers to validate a SNP subset, we are able to constrain the hypothesis space to the overlap region \mathcal{H}_o , increasing the chance of correctly identifying functional SNPs and SNP-SNP interactions.

4.2.2.2 Majority voting

The second classifier integration strategy applied in our GE hybrid algorithm is *majority voting* [102]. Majority voting is one of the simplest strategies in combining classification results from an ensemble of classifiers. Despite its simplicity, the power of this strategy is comparable to many other more complex methods [?, 110]. With a majority voting of L classifiers, consensus is made by k classifiers where:

$$k \geq \begin{cases} L/2 + 1 & : \text{ if } L \text{ is even} \\ (L+1)/2 & : \text{ if } L \text{ is odd} \end{cases} \quad (4.6)$$

Again, suppose a total of L classifiers, each having a different hypothesis denoted as h_i^s , ($i = 1, \dots, L$), are used to classify the data using SNP subset s , the fitness function derived from majority voting is as follows:

$$fitness_V(s) = BC \left(V_k \left(\mathbf{t}' \mid \sum_{i=1}^L p(\mathbf{t} | h_i^s, \mathbf{D}) \right), \mathbf{y} \right) \quad (4.7)$$

where \mathbf{y} is the class label vector of the test dataset \mathbf{D} , $V_k(\cdot)$ is the decision function of majority voting, and \mathbf{t}' is the voting prediction. Here the balanced classification accuracy $BC(\cdot)$ is calculated with voting results.

The reason for using the majority voting integration is to improve sample classification accuracy while also implicitly promoting diversity among individual classifiers [169].

4.2.2.3 Double fault diversity

The third objective function is an explicit diversity promoting strategy called *double fault* statistic. This statistic is commonly used to measure the diversity of ensemble classifiers [170].

Let $c_a, c_b \in \{F, S\}$ in which F denotes the sample being misclassified by a classifier while S denotes the sample being correctly classified. We define $N^{c_a c_b}$ as the number of samples that are (in)correctly classified by two classifiers in which the correctness of the two classifiers is denoted by c_a and c_b respectively. Using this notation, we can obtain the term:

$$D(p(\mathbf{t} | h_{c_a}^s, \mathbf{D}), p(\mathbf{t} | h_{c_b}^s, \mathbf{D})) = \frac{N^{FF}}{N} \quad (4.8)$$

which is the estimation statistic of coincident errors of a pair of classification models $h_{c_a}^s$ and $h_{c_b}^s$ (hence the name “double fault”) in classification of a total of N samples in test dataset \mathbf{D} , using SNP subset s .

The fitness with regard to the diversity measurement of L classifiers over subset s (denoted as $fitness_D(s)$) derived from the double fault statistic is defined as follows:

$$fitness_D(s) = 1 - \frac{2}{L(L-1)} \sum_{a=1}^L \sum_{b=a+1}^L D(p(\mathbf{t}|h_{c_a}^s, \mathbf{D}), p(\mathbf{t}|h_{c_b}^s, \mathbf{D})) \quad (4.9)$$

The value of this fitness function varies from 0 to 1. The value equals 0 when all classifiers misclassified every sample. It equals 1 when no sample is misclassified or there is a systematic diversity, leading to no sample being misclassified by any pair of classifiers.

4.2.3 Selecting classifiers

The motivation for applying nonlinear classifiers is based on the assumption that nonlinear and nonadditive relationships are commonly presented in gene-gene interaction [134]. This is particularly relevant in detecting complex epistatic interaction that involves both additive and dominant effects. Therefore, in ensemble construction, we focus on evaluating nonlinear classifiers. Moreover, we prefer classifiers that are relatively computationally efficient since the identification of gene-gene interaction is carried out in a wrapper manner. Thus, our attention has been focused on decision tree-based classifiers and instance-based classifiers, as well as their hybrids because they are fast among many alternatives, while also being able to perform nonlinear classification. However, we note that any combination of linear and nonlinear classifiers can be used in our framework. With the above considerations, an initial set of experiments is conducted to select candidate classifiers for ensemble construction. Those results will be presented in Section 4.5.1.

4.3 Evaluation datasets

We used the simulation datasets generated from the same model [132] as those described in Section 3.4 for evaluation. The dataset from the GWA study of AMD is also used as a case study of a real-world dataset [103].

For the simulation study, we used both balanced and imbalanced simulation datasets. For datasets with balanced class distribution [194], the class ratio is 1:1 with 100 case samples and 100 control samples. The datasets are simulated under three different genetic heritability models (heritability of 0.2, 0.1, and 0.05), and two SNP sizes (SNP size of 20 and 100). This gives six sets of datasets and every set contains 100 replicates, each generated with a different random seed [132]. The property of the imbalanced datasets used for evaluation is the same as the balanced datasets, except that the class ratio is approximately 1:2 with 67 case samples and 133 control samples. For imbalanced data, we restrict the evaluation to SNP size of 20, and therefore, we obtain three sets of datasets with each set containing 100 replicates. Table 4.2 summarizes the characteristics of the simulated datasets.

Table 4.2: Summary of simulation datasets used for SNP pair identification.

Dataset	Sample size	Ratio	Heritability	SNP size	No. replicates
balanced_200_0.2_20	200	1:1	0.2	20	100
balanced_200_0.1_20	200	1:1	0.1	20	100
balanced_200_0.05_20	200	1:1	0.05	20	100
balanced_200_0.2_100	200	1:1	0.2	100	100
balanced_200_0.1_100	200	1:1	0.1	100	100
balanced_200_0.05_100	200	1:1	0.05	100	100
imbalanced_200_0.2_20	200	1:2	0.2	20	100
imbalanced_200_0.1_20	200	1:2	0.1	20	100
imbalanced_200_0.05_20	200	1:2	0.05	20	100

4.4 Evaluation statistics

4.4.1 Evaluation statistics for single algorithm

We compare the detection power of the proposed GE algorithm with PIA (version: PIA-2.0) and MDR (version: mdr-2.0_beta_6). In the previous studies of MDR [164] and PIA [127], the power of an algorithm to identify gene-gene interactions is estimated as the percent of times the algorithm “successfully identifies” the true functional SNP pair from 100 replicates of simulated datasets. This is repeated for every heritability model to quantify how well each algorithm performs when dealing with datasets of varying difficulty (lower heritability being more difficult). An algorithm is

said to have successfully identified a functional SNP pair in a dataset if the true SNP-pair is reported as the top rank. For comparison with MDR and PIA, we follow this approach and estimate the power of GE, MDR, and PIA using the following statistics:

$$Power = \frac{N^S}{N} \quad (4.10)$$

where N is the number of datasets tested ($N=100$ in our case), and N^S is the number of successful identification.

For GE in particular, we are also interested in estimating the distribution of false discovery rate (FDR) and true positive rate (TPR) since, in the worst case, if there is no SNP-SNP interaction in the dataset, a top-ranked interaction list only contains false positive identifications. Formally, we estimate FDR as:

$$FDR(c) = \frac{N_{FP}(c)}{N(c)} \quad (4.11)$$

where $FDR(c)$ is the FDR at the cut-off of c , $N_{FP}(c)$ is the number of accepted false positive identifications at the cut-off of c , and $N(c)$ is the number of accepted identifications at the cut-off of c . Similarly, TPR is calculated as:

$$TPR(c) = \frac{N_{TP}(c)}{N_{TP}(c) + N_{FN}(c)} \quad (4.12)$$

where $TPR(c)$ is the TPR at the cut-off of c . $N_{TP}(c)$ and $N_{FN}(c)$ are the number of accepted true positives and the number of false negatives at the cut-off of c .

Both the rank and the identification frequency score of each SNP combination can be used as the cut-off to calculate FDR and TPR at different confidence levels. We consider both approaches, and using the 100 replicate datasets of each heritability model, we obtain the average FDR and TPR at different cut-offs for each heritability model.

4.4.2 Evaluation statistics for combining algorithms

One major motivation for developing a genetic ensemble algorithm for gene-gene interaction identification is to harness the complementary strength of different classifiers such that a more robust and predictive SNP subset can be obtained. To extend this idea further, we propose to combine the inferred SNP-SNP interaction from different

algorithms (such as MDR and PIA), in the hope that more robust results can be obtained. However, such benefits may come only when the results yielded by different SNP-SNP interaction identification algorithms are complementary to each other, which is analogous to the idea of the ensemble diversity.

By modifying the equation of double fault, we design the following terms to quantify the degree of complementarity (CD) of a pair of algorithms in SNP-SNP interaction identification:

$$SF(X, Y) = N^{FS} + N^{SF}, \quad DF(X, Y) = N^{FF} \quad (4.13)$$

$$CD(X, Y) = \frac{SF(X, Y)}{DF(X, Y) + SF(X, Y)} \quad (4.14)$$

where N^{XY} is the number of datasets with certain identification status using algorithms X and Y , and $X, Y \in \{F, S\}$ in which F denotes that an algorithm fails to identify the functional SNP pair while S denotes it succeeds in identifying the functional SNP pair. $SF(X, Y)$ (single fault) is the number of times algorithms X and Y give inconsistent identification results, which is the situation when one algorithm succeeds while the other one fails. $DF(X, Y)$ (double fault) is the number of times both X and Y fail. The pairwise degree of complementarity of the algorithms X and Y is determined by $CD(X, Y)$.

Excluding the case in which both X and Y achieve 100% successful identification (which gives $\frac{0}{0}$), the value of $CD(X, Y)$ varies between 0 and 1. When the results produced by X and Y are completely complementary to each other, the value of $DF(X, Y)$ decreases to 0, and the value of $CD(X, Y)$ reaches 1. On the contrary, the value of $CD(X, Y)$ decreases with the decrease of the degree of complementarity between algorithms X and Y , and reaches 0 when no degree of complementarity is found.

Our premise is that combining algorithms with a higher degree of complementarity will yield higher identification power. In this study, we estimate the joint power of two or three algorithms as:

$$Power_J(X, Y) = N - DF(X, Y) \quad (4.15)$$

$$Power_J(X, Y, Z) = N - TF(X, Y, Z); \quad TF(X, Y, Z) = N^{FFF} \quad (4.16)$$

where $TF(X,Y,Z)$ is the “triple fault” which gives the coincident errors of three identification algorithms, and $Power_J(X,Y)$ and $Power_J(X,Y,Z)$ are the joint power of combining two and three identification algorithms respectively.

4.5 Experiments and results

4.5.1 Classifier selection for ensemble construction

One of the most important steps in forming an ensemble of classifiers is base classifier selection. As described above, characteristics such as nonlinear separation capability, computational efficiency, high accuracy and diversity should be taken into account. With these considerations, a classifier selection and ensemble construction experiment was carried out. Specifically, we tested the merits of each candidate classifier using datasets with model numbers of 10, 11, 12, 13 and 14 from Moore *et al.* [132], all of which have a minor allele frequency of 0.2, heritability of 0.1, and sample size of 200 (100 case and 100 control). These are considered “difficult” datasets since they are simulated to have low minor allele frequency, low heritability, and small sample size [127]. Twenty replicates from each model were used for evaluation, and the power of each classifier in identifying the functional SNP pair was calculated. Figure 4.2a shows the 12 candidate classifiers we evaluated in this study. They are *REPTree* (REPT), *random tree* (RT), *alternating decision tree* (ADT) [65], *random forests* (RT) [21], *1-nearest neighbour* (1NN), *3-nearest neighbor* (3NN), *5-nearest neighbour* (5NN), *decision tree* (J48), *1-nearest neighbour with cover tree* (CT1NN), *3-nearest neighbour with cover tree* (CT3NN) [15], *entropy-based nearest neighbour* (KStar) [37], and *5-nearest neighbour with cover tree* (CT5NN).

The identification power of each classifier was estimated using the simulated datasets. Among the twelve classifiers, six of them successfully identified the functional SNP pair more than 50% of the time. Five of them were selected to form the ensemble (coloured in red in Figure 4.2a). They are J48, KStar, and three decision tree and k -nearest neighbour hybrids – CT1NN, CT3NN, and CT5NN.

The configuration of parameters such as GA chromosome mutation rate and integration weights of diversity measure, blocking, and voting were tested using the same sets of data as above. Specifically, the mutation rates tested were 0.05, 0.1 and 0.15. The integration weights of diversity tested were also 0.05, 0.1 and 0.15, while the integration

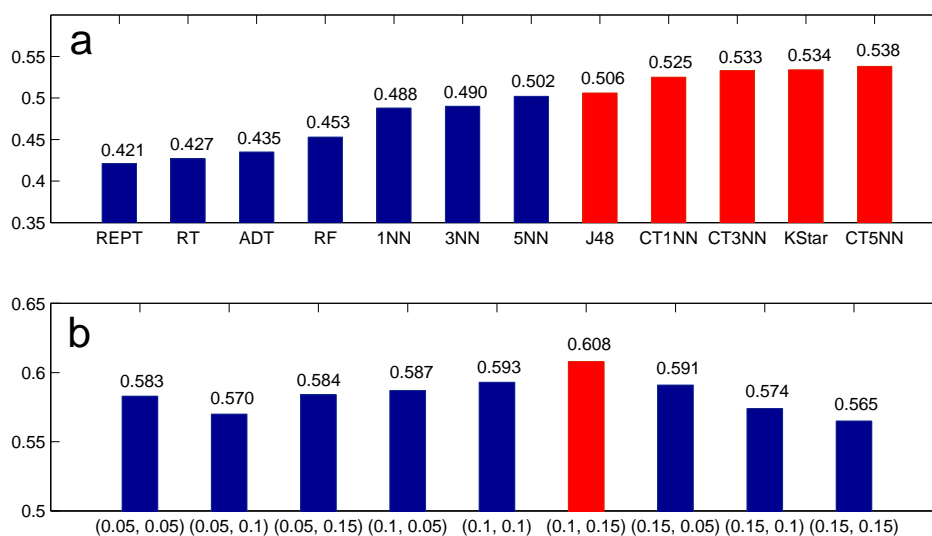


Figure 4.2: Selection of base classifiers and ensemble configuration. (a) Classifier selection. The value on the top of each bar denotes the estimated power in functional SNP pair identification using each classifier. (b) Ensemble configuration. The value on the top of each bar denotes the power in functional SNP pair identification using ensemble of classifiers with different values of GA chromosome mutation rate and diversity integration weight, respectively (denoted as a duplex in the x -axis).

weights for blocking and voting were kept equal, and the three weights add up to 1. This gives 9 possible configurations for the ensemble of classifiers. The identification powers of the ensemble of classifiers using these 9 configurations are shown in Figure 4.2b. It is observed that all the ensembles achieved better results than the best single classifier which has an identification power of 53.8%. Among them, the best parameter setting is (0.1, 0.15) which specifies the use of a mutation rate of 0.1 and integration weights of 0.15, 0.425, and 0.425 for diversity, blocking, and voting, respectively. This configuration gives an identification power of 60.8%, which is a significant improvement on 53.8%. This setting was then fixed in our GE in the followup experiments.

4.5.2 Simulation results

4.5.2.1 Gene-gene interaction identification

In the simulation experiment, we applied GE, PIA, and MDR for detecting the functional SNP pairs from 20 candidate SNPs and 100 candidate SNPs, respectively. Table

4.3 shows the evaluation results. By fixing the candidate SNP size to 20 and testing datasets generated with three heritability values (0.2, 0.1, and 0.05), we observed a decrease in the average identification power of the three algorithms (taking the average of the three identification algorithms) from 98.33 ± 0.94 to 78.67 ± 2.62 and to 43.67 ± 0.94 . By fixing the candidate SNP size at 100 and testing datasets generated with three heritability values (0.2, 0.1 and 0.05), the average identification power drops to 93.67 ± 0.94 , 48.33 ± 2.49 , and 19.00 ± 1.63 , respectively. It is clear that both heritability and SNP size are important factors to SNP-SNP interaction identification. When comparing the power of each algorithm, we found no significant differences. The standard deviation is generally small, ranging from 0.94 to 2.62, indicating that the three algorithms have similar performance.

Table 4.3: Functional SNP pair identification in balanced datasets using GE, PIA, and MDR.

Dataset	GE Power (%)	PIA Power (%)	MDR Power (%)
balanced_200_0.2_20	99	97	99
balanced_200_0.1_20	80	75	81
balanced_200_0.05_20	45	43	43
balanced_200_0.2_100	95	93	93
balanced_200_0.1_100	45	49	51
balanced_200_0.05_100	17	19	21

Table 4.4: Functional SNP pair identification in imbalanced datasets using GE, PIA, and MDR.

Dataset	GE Power (%)	PIA Power (%)	MDR Power (%)
imbalanced_200_0.2_20	92	90	95
imbalanced_200_0.1_20	59	45	62
imbalanced_200_0.05_20	32	24	27

To investigate whether an imbalanced class distribution affects identification power, we applied GE, PIA, and MDR to imbalanced datasets with a case-control ratio of 1:2 and a candidate SNP size of 20. From Table 4.4, we found that the power of the three identification algorithms decreased in comparison to those of the balanced datasets (Table 4.3). Such a decline of power is especially significant when the heritability of the

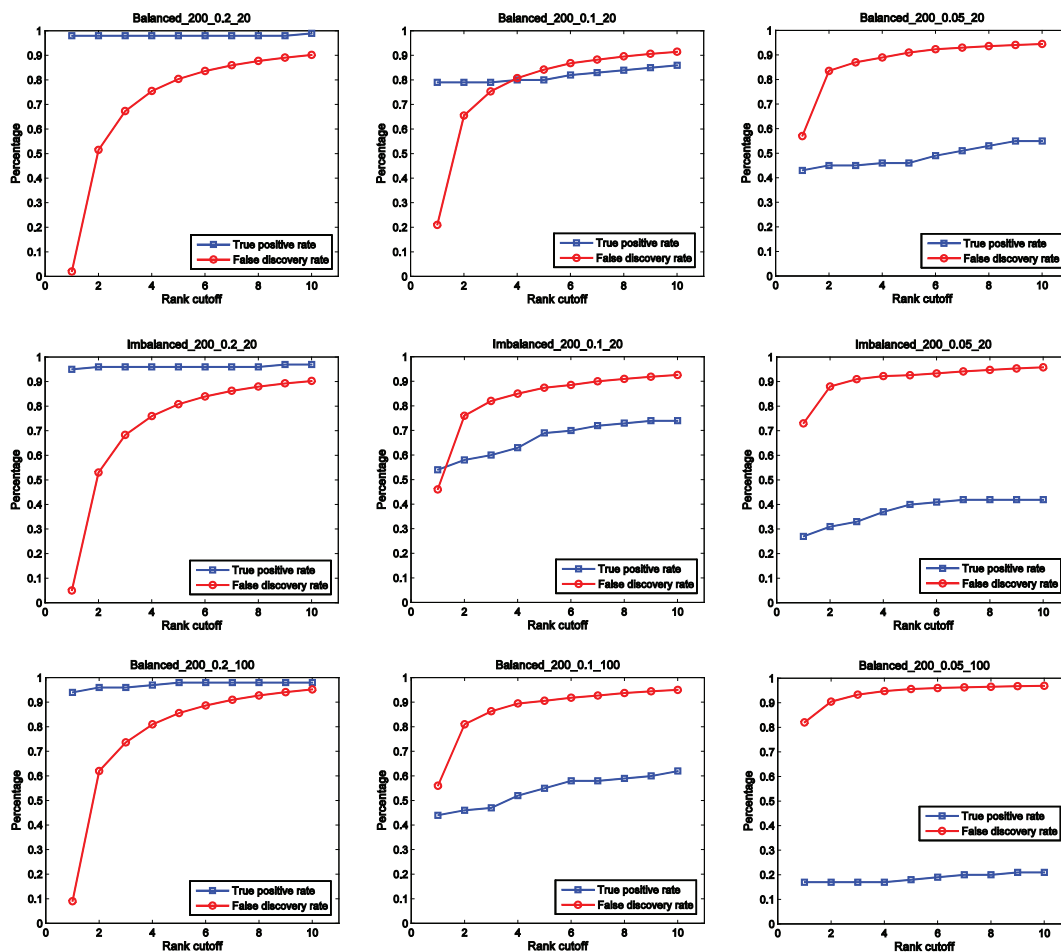


Figure 4.3: True positive rate and false discovery rate estimation of GE at different rank cut-offs. Simulated datasets with different heritability models, number of SNPs, and class distribution, are used to evaluate the true positive rate and false discovery rate of GE at different identification cut-offs using different rank-values [1-10].

dataset is small. This finding is essentially consistent with [194] in that datasets of larger heritability values are more robust to imbalanced class distribution. Since the sample size and other dataset characteristics in the balanced and the imbalanced datasets are the same, the observed decline of power could be attributed to the imbalanced class distribution. It is also noticed that the identification power of PIA is relatively lower compared to GE and MDR. This indicates that PIA may be more sensitive to the presence of the imbalanced class distribution than GE and MDR.

For the GE algorithm, two approaches were used to study the distribution of the TPR and FDR. For the first approach, we calculated the TPR and FDR by varying the rank

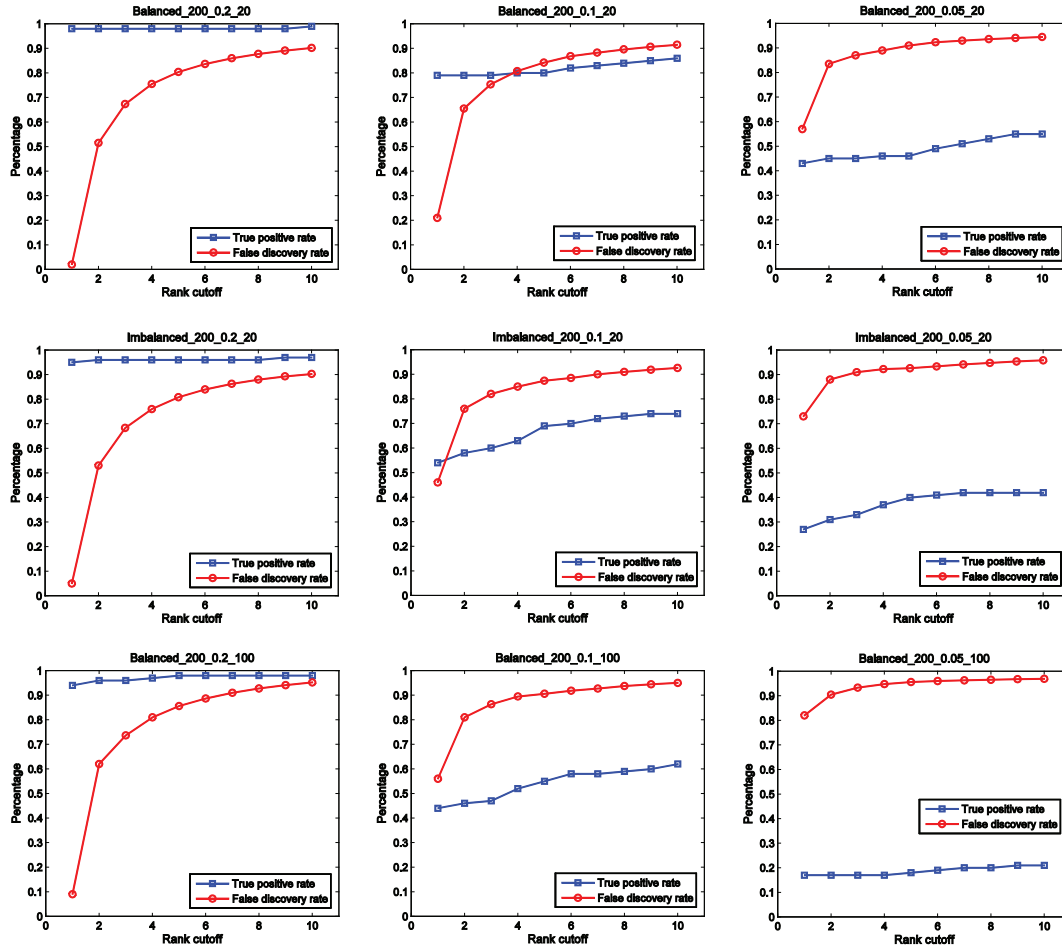


Figure 4.4: True positive rate and false discovery rate estimation of GE at different frequency score cut-offs. Simulated datasets with different heritability models, number of SNPs, and class distribution, are used to evaluate the true positive rate and false discovery rate of GE at different identification cut-offs using different frequency scores [1-0].

cut-off of the reported SNP pairs. Figure 4.3 shows the distribution by using a rank cut-off of 1 to 10 (the lower the number, the higher the rank). Note that the rank cut-off of 1 gives the results equal to the power defined in Equation 4.10. For the second approach, we calculated the TPR and FDR by varying the identification frequency cut-off of the reported SNP pairs. Figure 4.4 shows the distribution by decreasing the frequency cut-off from 1 to 0. By comparing the results, we found that the decrease of the heritability (from 0.2, to 0.1 and to 0.05) has the greatest impact on TPR of GE. Sample size appears to be the second factor (from 20 SNPs to 100 SNPs), and the imbalanced class distribution is the third factor (from a balanced ratio of 1:1 to an imbalanced ratio of

1:2).

Generally, by decreasing the cut-off stringency (either rank cut-off or identification frequency cutoff), the TPR increases, and therefore, more functional SNP pairs can be successfully identified. However, this is achieved by accepting increasingly more false identifications (higher FDR). The simulation results indicate that FDR calculated by using the identification frequency cut-off is very steady, regardless of the change of heritability, SNP size, or class ratio. In most cases, an FDR close to 0 is achieved with a cut-off greater than 0.78.

4.5.2.2 The degree of complementarity among GE, MDR, and PIA

As illustrated in Table 4.3 and Table 4.4, large candidate SNP size, low heritability value, and the presence of imbalanced class distribution together give the worst scenario for detecting SNP-SNP interaction. One solution to increase the chance of successful identification in such a scenario is to combine different identification results produced by different algorithms, which extends the idea of the ensemble method further. However, similar to the notion of diversity in ensemble classifier, the improvement can only come if the combined results are complementary to each other. Hence, the evaluation of the degree of complementarity among each pair of algorithms becomes indispensable.

We carried out a pairwise evaluation using Equation 4.13 and 4.14. Tables 4.5 and 4.6 give the results for balanced and imbalanced situations, respectively. We observed that higher degree of complementarity is generally associated with higher identification power. For the balanced datasets, the degree of complementarity of PIA and MDR is relatively low compared to those generated by GE and PIA, or GE and MDR. The results indicate that the GE algorithm, which tackles the problem from a different perspective, is useful in complementing methods like PIA and MDR in gene-gene interaction identification. As for the imbalanced datasets, the difference of the complementarity degree between each pair of algorithms is reduced. This suggests that more methods need to be combined for imbalanced datasets in order to improve identification power.

Table 4.5: Functional SNP pair identification in balanced datasets by combining multiple algorithms.

Dataset	(GE + PIA)		(GE + MDR)		(PIA + MDR)		(GE + PIA + MDR)
	CD	Power _J (%)	CD	Power _J (%)	CD	Power _J (%)	Power _J (%)
balanced_200_0.2_20	1.000	100	1.000	100	0.667	99	100
balanced_200_0.1_20	0.448	84	0.556	88	0.240	81	88
balanced_200_0.05_20	0.303	54	0.303	54	0.068	45	55
balanced_200_0.2_100	1.000	100	0.923	99	0.444	95	100
balanced_200_0.1_100	0.441	62	0.400	61	0.148	54	63
balanced_200_0.05_100	0.093	22	0.116	24	0.025	21	24

Table 4.6: Functional SNP pair identification in imbalanced datasets by combining multiple algorithms.

Dataset	(GE + PIA)		(GE + MDR)		(PIA + MDR)		(GE + PIA + MDR)
	CD	Power _J (%)	CD	Power _J (%)	CD	Power _J (%)	Power _J (%)
imbalanced_200_0.2_20	0.714	96	0.818	98	0.750	97	99
imbalanced_200_0.1_20	0.567	71	0.481	73	0.475	68	76
imbalanced_200_0.05_20	0.286	40	0.301	42	0.287	38	47

The last columns of Tables 4.5 and 4.6 show the joint identification power of the three algorithms in analysing balanced and imbalanced data. These results indicate a significant recovery of detection ability in functional SNP pair identification by applying three algorithms collaboratively. This is especially true when analysing imbalanced datasets and the heritability of the underlying genetic model is low. For example, the average identification power of three algorithms for imbalanced datasets with heritability of 0.1 and 0.05 are 55.33% and 27.67%, respectively (Table 4.4). By combining the results of the three algorithms, we are able to increase the power to 76% and 47%, respectively, improving by around 20% (Figure 4.5).

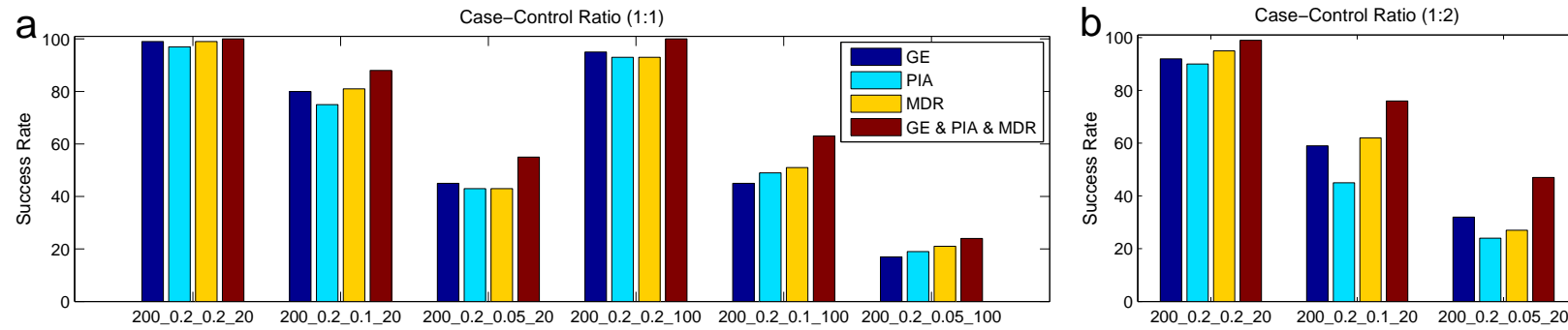


Figure 4.5: A comparison of identification power of GE, PIA, MDR, and combination of the three algorithms. The name of each dataset denotes sample size, heritability, and the number of SNPs (SNP size). (a) Identification power of each algorithm and their joint power using datasets with balanced class distribution. (b) Identification power of each algorithm and their joint power using datasets with imbalanced class distribution.

4.5.3 Real-world data application

As an example of a real-world data application, we applied the GE algorithm, PIA and MDR, to analyze the complex disease of AMD. To reduce the combinatorial search space, we followed the two-step analysis approach [191] and used a SNP filtering procedure that is similar to the method described in [34], which can be summarized as follows:

- S1: Excluding SNPs that have more than 20% missing genotype values of total samples.
- S2: Calculating allelic χ^2 -statistics of each remaining SNP and keeping SNPs which have a p -value smaller than 0.05 while discarding others. A total of 3583 SNPs passed filtering.
- S3: Utilizing RTREE program [212] to select top splitting SNPs in AMD classification. Two SNPs with *id* of rs380390 and rs10272438 are selected.
- S4: Utilizing Haploview program [11] to construct the Linkage Disequilibrium (LD) blocks around the above two SNPs.

After the above processing steps, we obtained 17 SNPs from the two LD blocks. They are rs2019727, rs10489456, rs3753396, rs380390, rs2284664, and rs1329428 from the first block, and rs4723261, rs764127, rs10486519, rs964707, rs10254116, rs10486521, rs10272438, rs10486523, rs10486524, rs10486525, and rs1420150 from the second block. Based on the previous investigation of AMD [63, 77, 175], we added another six SNPs to avoid analysis bias. They are rs800292, rs1061170, rs1065489, rs1049024, rs2736911, and rs10490924. Moreover, environmental factors of Smoking status and Sex are also included for potential environment interaction detection. Altogether, we formed a dataset with 25 factors for AMD association screening and gene-gene interaction identification.

Tables 4.7 and 4.8 illustrate the top 5 most frequently identified 2-factor and 3-factor interactions, respectively. At first glance, we see that the identification results given by different methods are quite different from one another. Considering the results of 2-factor and 3-factor interaction together, however, we find that two gene-gene interactions and a gene-environment interaction are identified by all three methods. Specifically, the first gene-gene interaction is characterized by the SNP-SNP interaction pair

of rs10272438×rs380390. The first SNP is an A/G variant located in intron 5 of *BBS9* located in 7p14, while the second SNP is a C/G variant located in intron 15 of *CFH* located in 1q32. The second frequently identified gene-gene interaction is characterized by the SNP-SNP interaction pair of rs10490924×rs10272438. The first SNP in this interaction pair is a nonsynonymous coding SNP of Ser69Ala alteration located in exon 1 of *ARMS2* located in 10q26, and the second SNP is again the A/G variant located in intron 5 of *BBS9* located in 7p14. As to the gene-environment interaction pair, it is characterized by rs10272438×Sex. This pair indicates that the SNP factor of the A/G variant located in intron 5 of *BBS9* located in 7p14 is likely to associate with the disease differently in males and females.

We also test the association of the Age factor with AMD by using Gaussian discretization to partition the age value of each sample into three categories as follows:

$$age(x) = \begin{cases} \text{“young”} & x \leq \mu - \sigma/2 \\ \text{“medium”} & \mu - \sigma/2 < x < \mu + \sigma/2 \\ \text{“elderly”} & x \geq \mu + \sigma/2 \end{cases} \quad (4.17)$$

where μ is the average age value and σ is the standard deviation of age values.

After including the Age factor in the dataset, all three algorithms identified the gene-environment interaction of rs1420150×Age as the interaction with major implication, indicating that Age factor is, expectedly, strongly associated with the development of AMD. The SNP that interacted with the Age factor is a C/G variant located in intron 9 of *BBS9* located in 7p14.

Table 4.7: Two-factor interaction candidates of the AMD dataset using GE, PIA, and MDR, respectively.

GE	CV Acc %	PIA	CV Acc %	MDR	CV Acc %
rs10272438×rs4723261	68.5	rs10272438×rs380390	64.2	rs10490924×rs1420150	65.5
rs10272438×rs2736911	66.9	rs10490924×rs10272438	68.2	rs10272438×rs1065489	68.4
rs10272438×rs964707	68.5	Y402H×rs10272438	65.5	rs10272438×rs2284664	66.7
rs10272438×Sex	67.5	rs10254116×Smoking	67.1	rs10272438×Sex	67.5
rs10272438×rs2284664	66.7	rs10490924×rs10254116	67.7	rs10254116×rs2736911	67.7

Table 4.8: Three-factor interaction candidates of the AMD dataset using GE, PIA, and MDR, respectively.

GE	CV Acc %	PIA	CV Acc %	MDR	CV Acc %
rs10272438×rs4723261 ×rs964707	68.5	rs10272438×rs380390 ×rs10486524	59.8	rs10272438×rs380390 ×rs10486524	59.8
rs10272438×rs4723261 ×rs2736911	67.1	rs10272438×rs380390 × Sex	61.2	rs10272438×rs380390 ×rs964707	63.4
rs10272438×rs380390 ×rs964707	63.4	rs10272438×Sex ×rs1065489	68.1	Y402H×rs10272438 ×rs964707	60.7
rs10490924×rs10272438 ×rs4723261	65.0	rs10272438×rs380390 ×rs10254116	66.6	rs10490924×rs10272438 × Sex	63.4
rs10272438×Sex ×rs4723261	63.5	rs10272438×rs380390 ×rs1420150	59.4	rs10272438×Sex ×rs2736911	65.7

Table 4.9 summarizes the factors involved in potential interactions identified by all three different algorithms. Overall, the experimental results suggest that genes of *BBS9* (Bardet-Biedl syndrome 9), *CFH* (complement factor H), and *ARMS2* (age-related maculopathy susceptibility 2) with the external factors of Age and Sex, and the interactions among them are strongly associated with the development of AMD. This is essentially consistent with current knowledge of AMD development in the literature [63, 77, 103, 175].

Table 4.9: SNPs and environmental factors that statistically associated with AMD.

Factor	Chrom.	Gene	Location	Effect	Main effect p -value
rs10272438	7p14	<i>BBS9</i>	intron 5	A/G	1.4×10^{-6}
rs1420150	7p14	<i>BBS9</i>	intron 9	C/G	2.1×10^{-2}
rs380390	1q32	<i>CFH</i>	intron 15	C/G	4.1×10^{-8}
rs10490924	10q26	<i>ARMS2</i>	exon 1	Ser69Ala	1.8×10^{-3}
Sex	–	–	–	–	1.4×10^{-2}
Age	–	–	–	–	1.1×10^{-3}

4.6 Summary

The advance of high-throughput genotyping technologies provides the opportunity to elucidate the mechanism of gene-gene and gene-environment interaction via SNP markers. However, current algorithms have limited power in terms of identifying true SNP-SNP interactions. Moreover, the simulation results indicate that factors such as heritability, candidate SNP size, and the presence of imbalanced class distribution all have profound impact on a given algorithm's power in identifying functional SNP interactions. One practical way to improve the chance of identifying SNP-SNP interactions is to combine different methods where each addresses the same problem from a different perspective. The rationale is that the consensus may increase the confidence of identifications and complementary results may improve the power of identification.

Due to these considerations, we proposed a hybrid algorithm using a genetic ensemble approach. Using this approach, the problem of SNP-SNP interaction is converted to a combinatorial feature selection problem. Our simulation study indicates that the proposed GE algorithm is comparable to PIA and MDR in terms of identifying gene-gene interaction for complex disease analysis. Furthermore, the experimental results

demonstrate that the proposed algorithm has a high degree of complementarity to PIA and MDR, suggesting the combination of GE with PIA and MDR is likely to lead to higher identification power.

For practical application of the GE algorithm, the experimental results from the simulation datasets suggest that taking the top-ranked result generally gives a higher sensitivity of identifying SNP-SNP interactions than using a frequency score cut-off. However, if the detectability of the SNP-SNP interaction is low or no such interaction is present in the dataset, the top-ranked result is likely to be a false positive identification. A more conservative approach is to use an identification frequency cut-off of 0.75–0.8 which in our simulation study gives identification results with an FDR close to 0. For any identified SNP pair with an identification frequency higher than 0.8, the confidence is very high.

As a down-stream analysis, we can fit the identified SNP pairs using a logistic model with interaction terms and calculate the p -values of their coefficients in order to quantify the strength of the interaction. In particular, to test additive and dominant effects, we can fit the reported SNP combinations using the model described by Cordell [41] and analyse the coefficients associated with additive and dominant effects of each SNP.

Current GWA studies commonly produce several hundreds of thousands of SNPs, yet the gene-gene interaction identification algorithms like MDR, PIA and the proposed GE algorithm can only cope with a relatively small number of SNPs in a combinatorial manner. Therefore, a filtering procedure is required to reduce the number of SNPs to a “workable” amount before those combinatorial methods can be applied to datasets generated by GWA studies [71, 130]. More efforts are required to seamlessly connect these two components to maximize the chance of detecting complex interactions among multiple genes and environmental factors [191].

In conclusion, we proposed a GE algorithm for gene-gene and gene-environment interaction identification. It is comparable to two other state-of-the-art algorithms (PIA and MDR) in terms of SNP-SNP interaction identification. The experimental results also demonstrated the effectiveness and the necessity of applying multiple methods each with different strengths to the gene-gene and gene-environment interaction identification for complex disease analysis.

4.7 Software availability

The genetic ensemble package for gene-gene interaction identification is freely available from:

<http://code.google.com/p/genetic-ensemble-snpX>

Chapter 5

Gene Sets Selection From Microarray

This chapter is based on the following publication:

Pengyi Yang, Bing B. Zhou, Zili Zhang, Albert Y. Zomaya, A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data. BMC Bioinformatics, 11:S5, 2010

5.1 Microarray data from a computational viewpoint

In previous two chapters, we concentrated on processing genotype data generated from genomic levels. In this section, we focus on processing transcriptomic data with ensemble methods and hybrid algorithms.

One of the key technologies that has been predominately applied for high-throughput transcriptome profiling since its development in the mid 90s is gene expression microarray [49, 174]. Microarray technologies parallelize the finding of the disease/trait causing genes by simultaneously measuring tens of thousands of genes. For example, for the studies that are designed to find genes associated with certain cancers, tissue samples from cancer patients and normal individuals can be collected and profiled using microarrays.

Common steps in microarray data analysis include data normalization, disease/trait-associated gene identification, sample classification, and gene enrichment analysis [4]. Following these analysis procedures, downstream validation may be performed in a wetlab. It is clear that a successful downstream validation relies heavily on the initial data analysis, yet the data analysis has been found to be nontrivial. For example,

the measured gene expressions from microarray experiments are unavoidably affected by random variations and systematic variations that occur in different samples and experimental effects. Therefore, a proper data normalization procedure is critical to ensure gene expressions are comparable within and between each sample and experiment batch [205]. Similar to the SNP interaction filtering and identification, the identification of disease/trait-associated genes is also hampered by the problems such as *curse-of-dimensionality* and the *curse-of-sparsity* because the number of genes measured by microarray is commonly several orders higher than the number of samples used for profiling [182]. Therefore, an efficient and accurate gene selection approach that is capable of identifying key genes and gene sets that are differentially expressed between different treatments or diseases from a huge candidate set is crucial to ensure accurate sample classification and followup biological validation.

In this chapter, we explore using hybrid approaches for disease associated gene set selection and sample classification. Wrapper and filter algorithms are commonly treated as different approaches for differentially expressed gene selection. The uniqueness of the proposed approach is that filter and wrapper algorithms are combined as a hybrid algorithm and the strengths of each approach are harnessed in an integrative way. We apply our hybrid approach to several benchmark microarray datasets and compare results with those obtained from using either filter or wrapper feature selection approaches.

5.2 Hybrid approach for gene set selection and sample classification of microarray data

Feature selection is a key technique for identifying disease/trait-associated genes from high-dimensional microarray data. We categorized feature selection algorithms into filter, wrapper, and embedded approaches in Section 2.1.2. As mentioned earlier, a filter approach separates feature selection from the sample classification component, thus, they are generally computationally efficient. However, the effects of the selected genes in sample classification is useful information that may be used to improve classification accuracy [104]. Therefore, the wrapper approach, which incorporates the classification information for feature selection, may provide higher sample classification accuracy. If the goal of the study is to accurately distinguish disease samples and controls, one may prefer wrapper algorithms to filter algorithms. Yet, the computational complexity

of wrapper algorithms is generally much higher than filter algorithms since one needs to iteratively classify samples, often in a cross-validation manner, so as to objectively extract classification information for feature selection.

We argue that a good trade-off between filter and wrapper approaches can be achieved by combining the two techniques in that the filter algorithm is used for a fast initial screening and the wrapper algorithm is then applied to the reduced gene subset to accurately identify the most important gene set in a computationally efficient manner. Therefore we propose following procedure:

1. Split each dataset into external training sets and external test sets with an external N -fold stratified cross validation.
2. Filter the external training sets by using a filter algorithm.
3. Split the filtered external training sets into internal training sets and internal test sets with an internal N -fold stratified cross validation.
4. Identify gene set with a wrapper algorithm using internal training sets and internal test sets.
5. Evaluate the selected gene set on sample classification using the external test set.

The above procedure embedded feature selection in an internal cross validation and therefore provides an objective evaluation of the algorithm.

5.2.1 Multiple filter enhanced genetic ensemble

For the wrapper algorithm, we apply a similar genetic ensemble (GE) system as those used for gene-gene interaction identification in Section 4.2 because this model is able to evaluate genes as subsets, as opposed to individual genes, and could potentially identify functional units. This is important because genes are commonly connected by pathways and function as groups. Therefore, evaluating individual genes may miss important biopathway information.

To increase the speed of convergence and to further improve the generalization property of the selected genes and gene subsets on unseen data classification, we incorporate multiple filtering algorithms into the GE system. This hybrid system is named the *multi-filter enhanced genetic ensemble* system, or MF-GE for short. The flow chart of

the hybrid system is illustrated in Figure 5.1. A novel mapping strategy for multiple filtering information fusion is developed to fuse the evaluation scores from multiple filters, and this information is incorporated into the GE system for gene selection and classification. Thus, the system encompasses two components, i.e., “filtering process” and “wrapper process”. In the filtering process, multiple filtering algorithms are applied to score each candidate gene in the microarray dataset. The scores of each gene are then integrated to the wrapper process. In the wrapper process, the GE system is used to select discriminative genes using the information provided by the filtering process. The algorithm executes iteratively, collecting multiple gene subsets. The final collections are ranked and the top genes are used for sample classification.

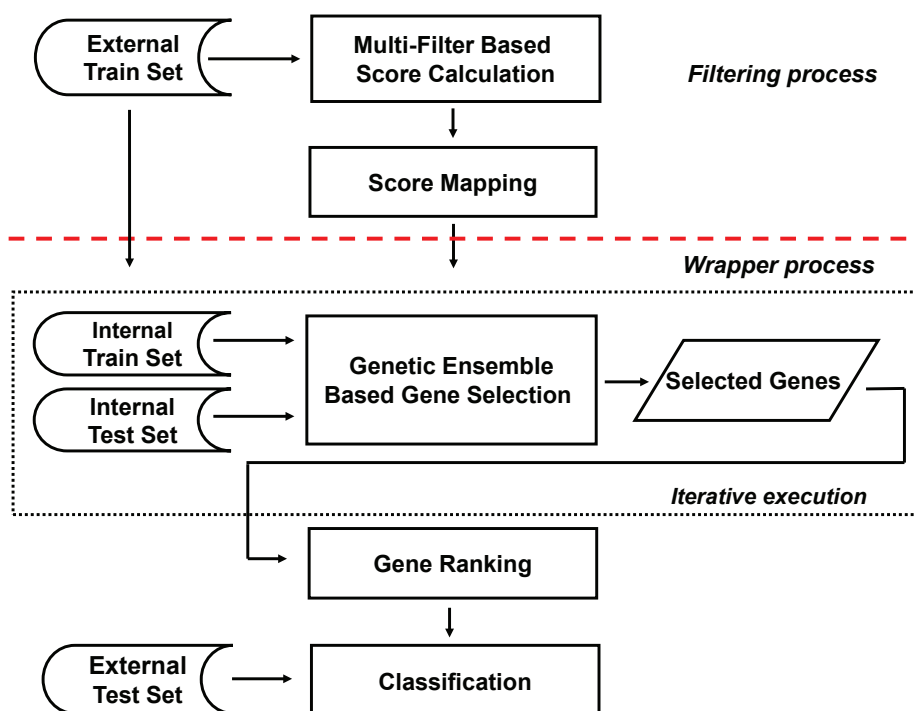


Figure 5.1: Flow chart of the MF-GE hybrid system for gene selection and classification of microarrays.

5.2.2 Score mapping for information fusion of multiple filtering algorithms

Traditionally, filtering algorithms select differential genes independently for the classification process. However, such information could be beneficial if appropriately integrated into the wrapper procedure. As shown in Figure 5.1, the intermediate step called “score mapping” serves as the synergy between the filtering process and the wrapper process.

The score mapping process starts by calculating scores for each candidate gene with different filtering algorithms. One issue in integrating those scores is that different filtering algorithms often provide evaluation scores with different scales. In order to combine the evaluation results of multiple filters, we must transform the evaluation scores into a common scale. Therefore, softmax scaling is adopted to normalize the gene evaluation results of each filtering algorithm into the range of [0, 1]. The calculation is as follows:

$$\hat{x}_{ik} = \frac{1}{1 + \exp(-y)}$$

in which

$$y = \frac{x_{ik} - \bar{x}_k}{r\sigma_k}$$

where \bar{x}_k is the average expression value of the k th gene among all samples, σ_k is the standard deviation of the k th gene among all samples, and \hat{x}_{ik} is the transformed value of x_{ik} which denotes the expression value of the k th gene in sample i .

After softmax scaling, the evaluation scores from different filtering algorithms are summed up to a set of total scores that indicates the overall score of each gene under the evaluation of multiple filtering algorithms. The total scores are then multiplied by 10 and rounded to an integer. Those with scores smaller than 1 are set to 1 to make sure all candidate genes are included in the wrapper selection process. The scores are then converted into frequency. The genetic operations such as “chromosome” initialization and mutation of the original GE system are conducted based on this “gene frequency map”. Figure 5.2 gives an example of creating a gene frequency map using two filters.

It is readily noticed that genes with higher overall evaluation scores will appear in the gene frequency map more frequently, and thus, will have a better chance to be

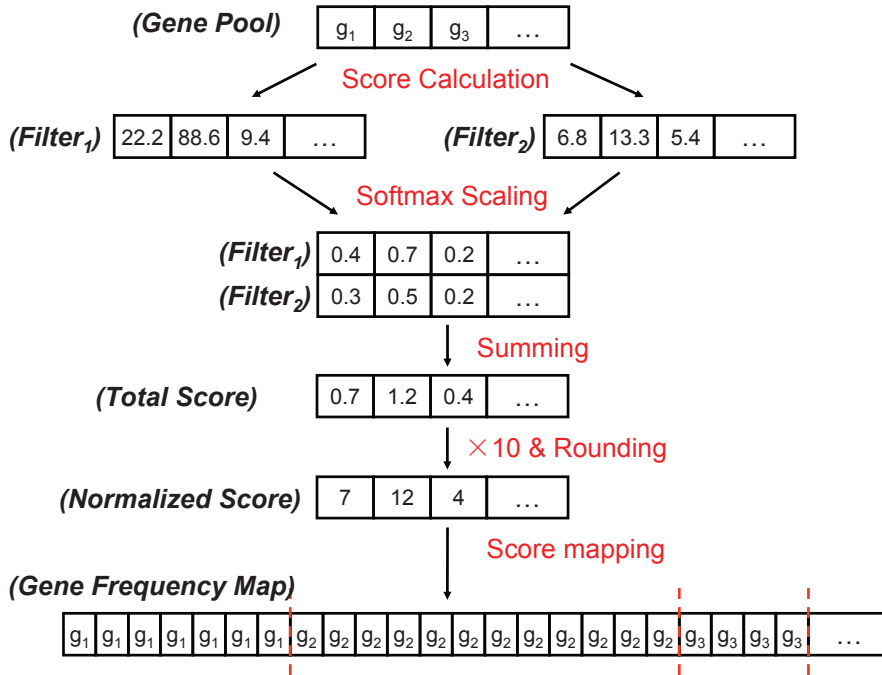


Figure 5.2: An example of multiple filter score mapping strategy for information fusion.

chosen in the initialization step and the mutation step. In this way, multiple filter information is fused into the gene selection process.

5.3 Filters and classifiers

The MF-GE system incorporated the evaluation scores of five filtering algorithms, namely χ^2 -test, ReliefF, Symmetrical Uncertainty, Information Gain, and Gain Ratio. Furthermore, we extend the GE system, introduced in Section 4.2, for multiple classes datasets. We evaluate multiple classifier combinations using a multiagent framework [216] and find that the combination of five classifiers, namely, *decision tree*, *random forests*, *3-nearest neighbour*, *7-nearest neighbour*, and *naive bayes* is the best in terms of sample classification and feature selection stability.

In this section, we start by introducing the filtering algorithms incorporated in the MF-GE hybrid system. The ReliefF algorithm is introduced in Section 3.2.1, and therefore, is excluded from here. Then, we describe the extension of the GE system for multiple classes datasets.

5.3.1 Filter algorithms

5.3.1.1 χ^2 -test

For gene selection, χ^2 -test can be considered to calculate the occurrence of a particular value of a gene and the occurrence of a class associated with this value. Formally, the merit of a gene is quantified as follows:

$$\chi^2(g) = \sum_{v \in V} \sum_{i=1}^m \frac{(N(g = v, c_i) - E(g = v, c_i))^2}{E(g = v, c_i)}$$

where c_i , ($i = 1, \dots, m$) denotes the possible classes of the samples from a dataset, while g is the gene that has a set of possible values denoted as V . $N(g = v, c_i)$ and $E(g = v, c_i)$ are the observed and the expected co-occurrence of $g = v$ with the class c_i , respectively.

5.3.1.2 Symmetrical uncertainty

Symmetrical uncertainty evaluates the worth of a gene by measuring the symmetrical uncertainty with respect to the sample class [198]. Each gene is evaluated as follows:

$$\text{Symm}U(g) = \frac{2 \times ((H(\text{class})) - H(\text{class}|g))}{H(\text{class}) + H(g)}$$

where $H(\cdot)$ is the information entropy function. $H(\text{class})$ and $H(g)$ give the entropy values of the class and a given gene, while $H(\text{class}|g)$ gives the entropy value of a gene with respect to the class.

5.3.1.3 Information gain

Information gain is commonly used in nodes selection for decision tree construction. It measures the number of bits of information provided in class prediction by knowing the value of features [196]. Let c_i belong to a set of discrete classes (1, ..., m). Let V be the set of possible values for a given gene g . The information gain of a gene g is defined as follows:

$$\text{InfoGain}(g) = - \sum_{i=1}^m P(c_i) \log P(c_i) + \sum_{v \in V} \sum_{i=1}^m P(g = v) P(c_i | g = v) \log P(c_i | g = v)$$

5.3.1.4 Gain ratio

Gain ratio incorporates “split information” of features into *information gain* statistics. The “split information” of a gene is obtained by measuring how broadly and uniformly it splits the data [129]. Let us consider again that a microarray dataset has a set of classes denoted as c_i , ($i = 1, \dots, m$), and each gene g has a set of possible values denoted as V . The discriminative power of a gene g is given as:

$$GainRatio(g) = \frac{InfoGain(g)}{Split(g)}$$

in which:

$$Split(g) = - \sum_{v \in V} \sum_{i=1}^m \frac{|S_v|}{|S|} \log \frac{|S_v|}{|S|}$$

where S_v is the subset of S of which gene g has value v .

Each algorithm evaluates the worth of a candidate gene in a different way. The hope is that genes of real biological relevance will show high scores in multiple criteria, as opposed to the artifacts that may by chance show high scores in one criterion but perform much worse according to the others.

5.3.2 Classification components

After fusing the filtering information from multiple filters, the aim is to apply the GE system to identify a subset of key genes that can maximize the prediction accuracy on diseases. We adopt the same architecture as described in Section 4.2, but extend the system for dealing with datasets with multiple classes. Specifically, for *blocking* and *majority voting*, we have the same equations as follows:

$$fitness_B(s) = \sum_{i=1}^L BC(p(\mathbf{t}|h_i^s, \mathbf{D}), \mathbf{y}) \quad (5.1)$$

and

$$fitness_V(s) = BC \left(V_k \left(\mathbf{t}' \mid \sum_{i=1}^L p(\mathbf{t} | h_i^s, \mathbf{D}) \right), \mathbf{y} \right) \quad (5.2)$$

where \mathbf{y} is the class label vector of the test dataset \mathbf{D} , function $p(\cdot)$ predicts/classifies samples in \mathbf{D} as \mathbf{t} using h_i^s , and $V_k(\cdot)$ is the decision function of majority voting that combines multiple predictions into a consensus prediction of \mathbf{t}' .

However, the calculation of $BC(\cdot)$ is modified as:

$$BC(p(\mathbf{t} | h_i^s, \mathbf{D}), \mathbf{y}) = \frac{\sum_{j=1}^m Se^j}{m} \quad (5.3)$$

and

$$Se^j = \frac{N_{TP}^j}{N^j} \times 100 \quad (5.4)$$

where Se^j is the sensitivity value calculated as the percentage of the number of true positive classification (N_{TP}^j) of samples in class j , N^j denotes the total number of samples in class j , and m is the total number of classes.

5.4 Experiment designs and results

In this section, we describe the dataset used for evaluation, the details of implementation and the experimental results.

5.4.1 Datasets and data pre-processing

We gathered four benchmark microarray datasets for our algorithm evaluation. These included binary class and multi-class classification problems. Table 5.1 is a summary of the datasets.

The ‘‘Leukemia’’ dataset [70] investigates the expression of two different subtypes of leukemia (47 ALL and 25 AML), and the ‘‘Colon’’ dataset [5] contains expression patterns of 22 normals (denoted as NOR) and 40 tumour (denoted as TUM) tissues. The ‘‘Liver’’ dataset [33] has 82 samples labelled as Hepatocellular carcinoma (HCC) and another 75 samples labelled as non-tumour (NON). The task for these three datasets is to identify a small group of genes that can distinguish samples from two classes. The ‘‘MLL’’ dataset [9] provides a multi-class classification problem. The task is to

Table 5.1: Microarray datasets used for algorithm evaluation.

Dataset name	Leukemia	Colon	Liver	MLL
Reference	[70]	[5]	[33]	[9]
Number of Samples	72	62	157	72
Number of Genes	7129	2000	20983	12582
Number of Classes	2	2	2	3
Class1	ALL: 47	TUM: 40	HCC: 82	ALL: 24
Class2	AML: 25	NOR: 22	NON: 75	MLL: 20
Class3				AML: 28

discriminate each class using a selected gene profile. These four datasets represent the general scenarios in gene selection and sample classification of microarray datasets.

Each dataset is pre-processed by converting the raw expression value by logarithm of 2 and normalizing the value to the range of [0, 1]. Then each dataset is split into external training sets and external test sets with a 3-fold stratified cross validation. A pre-filtering procedure is applied to select the top 200 genes by using the between-group to within-group sum of square ratio (BSS/WSS) [56]. Following that, the external training sets are split into internal training sets and internal test sets with an internal 3-fold stratified cross validation. The gene score calculation is conducted by using the internal training sets while the wrapper selection is performed using internal training sets and internal test sets collaboratively. The external test sets are reserved for the evaluation of the selected genes on unseen data classification, and are excluded from pre-filtering and the gene selection processes.

5.4.2 Implementation

The classification component in the genetic ensemble system is determined by using a multiagent approach as described in [216]. A set of initial tests is conducted to determine working parameter configurations. The best parameter settings in the initial test are chosen and fixed for the later experiments. Specifically, the iteration of the genetic ensemble procedure is set to 100. Within each iteration, the population size of GA is 100. These 100 populations are divided into two niches each of 50, and are evolved separately. After every 10 generations, the favourite chromosomes from the two niches are exchanged with each other. The probability of crossover p_c is 0.7. A novel mutation strategy is implemented to allow multiple mutations; that is, when a single

mutation happens (with the probability of 0.1) on a chromosome, another single point mutation may happen on the same chromosome with the probability of 0.25 and so on. The selection method is the tournament selection with the candidate size of 3, and the contribution weights of w_1 and w_2 are set to 0.5. Lastly, the termination condition for each iteration is either that the termination generation of 100 is reached or the similarity of the population converges to 90%. Table 5.2 summarizes the parameter settings.

Table 5.2: Parameter setting for genetic ensemble.

Parameter	Value
Fitness Function	Multi-Objective
Iteration	100
Population Size	100
Niche	2
Chromosome Size	15
Termination	Multiple Conditions
Selection	Tournament Selection (3)
Crossover	Single Point (0.7)
Mutation	Multi-Point (0.1 & 0.25)
Contribution Weight	$w_1 = 0.5, w_2=0.5$

In our parameter tuning experiments, the average gene subset size is within 2 to 10. Thus, the GA chromosome is represented as a string of size 15. In chromosome coding, each position is used to specify the *id* of a selected gene or assigned a “0” to denote no gene is selected at the current position. This gives a population of gene subsets of different sizes with a maximum of 15.

Classifiers and filters are created by using Waka API [78]. Specifically, J48 algorithm is used to create a classification tree. The random forest algorithm with size of 7 trees is applied, while k -nearest neighbour and naive bayes classifiers are adopted with default parameters. Each filtering algorithm is provoked for evaluation of each candidate gene and integrated from our main code through the class API of Waka.

The GA/KNN code was downloaded from the author’s web site (<http://www.niehs.nih.gov/research/resources/software/gaknn>). Chromosome length of 15, iteration of 1000, and majority voting with $k=3$ of the k NN were used. For each dataset, GA/KNN requires a pre-specified selection threshold of cut-off. Therefore, different thresholds were used according to their classification power on different datasets.

5.4.3 Results

The first set of experiments is focused on comparing the classification accuracy of the selected gene sets from MF-GE hybrid with GE, GA/KNN, and Gain Ratio filter algorithms. Instead of trying to achieve the highest classification accuracy, we aim to differentiate the classification performance of different gene selection algorithms. The ranking and classification of each dataset are repeated 5 times and each time the top 5, 10, 15, and 20 genes are used for sample classification. We report the average of the classification results.

The evaluation results obtained from the different microarray datasets are depicted in Tables 5.3-5.6. In each table, the classification results using each individual classifier as well as the mean and their majority voting are listed. It is easy to see that the MF-GE system has a higher average classification accuracy for all datasets. For example, 1.20%, 1.33%, 0.75%, and 1.85% improvements of mean over the original GE (which is the second best in average over all datasets) are obtained using the MF-GE system for Leukemia, Colon, Breast, and MLL, respectively. Given the fact that the GE part of these two algorithms is the same, the natural explanation of the improvement is the fusion of multiple filter information.

Table 5.3: Classification comparison of different gene ranking algorithms using Leukemia dataset.

Dataset	Classifier	Algorithm			
		Gain Ratio	GA/KNN	GE	MF-GE
Leukemia	C4.5	87.41	78.55 ± 2.96	83.04 ± 1.56	84.51 ± 2.53
	Random Forests	92.59	91.75 ± 0.99	90.82 ± 1.87	92.35 ± 0.70
	3-Nearest Neighbour	91.16	93.74 ± 1.27	94.30 ± 1.73	95.48 ± 0.95
	7-Nearest Neighbour	83.10	89.43 ± 1.10	90.45 ± 2.04	90.86 ± 1.26
	Naive Bayes	92.78	90.28 ± 1.33	96.20 ± 0.93	96.27 ± 1.65
	Mean	89.41	88.75	90.69	91.89
	Majority Voting	92.45	93.29 ± 1.29	95.33 ± 0.96	96.23 ± 1.26

Table 5.4: Classification comparison of different gene ranking algorithms using Colon dataset.

Dataset	Classifier	Algorithm			
		Gain Ratio	GA/KNN	GE	MF-GE
Colon	C4.5	71.49	62.43 ± 2.78	73.08 ± 2.77	76.64 ± 1.53
	Random Forests	63.66	73.48 ± 2.09	71.86 ± 2.02	74.35 ± 2.01
	3-Nearest Neighbour	68.02	73.83 ± 1.57	75.43 ± 0.92	77.01 ± 2.09
	7-Nearest Neighbour	65.43	67.62 ± 1.45	68.39 ± 1.76	68.78 ± 2.32
	Naive Bayes	70.61	72.12 ± 1.68	76.46 ± 2.14	75.07 ± 2.38
	Mean	68.84	69.90	73.04	74.37
	Majority Voting	70.56	73.37 ± 1.84	75.81 ± 2.00	76.98 ± 1.06

Table 5.5: Classification comparison of different gene ranking algorithms using Liver dataset.

Dataset	Classifier	Algorithm			
		Gain Ratio	GA/KNN	GE	MF-GE
Liver	C4.5	84.88	88.33 ± 0.94	87.09 ± 0.79	88.19 ± 0.56
	Random Forests	89.65	90.31 ± 1.11	91.87 ± 0.94	93.13 ± 1.18
	3-Nearest Neighbour	87.76	90.46 ± 0.65	93.57 ± 0.57	93.39 ± 0.79
	7-Nearest Neighbour	87.65	89.53 ± 0.56	91.91 ± 0.69	92.54 ± 0.57
	Naive Bayes	89.05	90.85 ± 0.51	92.70 ± 0.67	93.63 ± 0.64
	Mean	87.80	89.90	91.43	92.18
	Majority Voting	89.02	91.60 ± 0.36	93.37 ± 0.46	93.80 ± 0.47

Table 5.6: Classification comparison of different gene ranking algorithms using MLL dataset.

Dataset	Classifier	Algorithm			
		Gain Ratio	GA/KNN	GE	MF-GE
MLL	C4.5	81.87	72.89 ± 2.08	78.27 ± 3.10	81.54 ± 1.67
	Random Forests	83.02	88.07 ± 1.05	88.20 ± 1.41	89.74 ± 0.60
	3-Nearest Neighbour	79.63	88.22 ± 1.30	86.18 ± 1.39	88.14 ± 1.09
	7-Nearest Neighbour	79.63	86.72 ± 1.03	85.02 ± 1.49	86.69 ± 1.98
	Naive Bayes	83.95	89.62 ± 0.67	90.68 ± 1.28	91.50 ± 0.67
	Mean	81.62	85.10	85.67	87.52
	Majority Voting	83.88	88.38 ± 0.97	89.02 ± 1.71	91.08 ± 0.96

An apparent question is whether such improvements with multiple filters justify the additional computational expenses? This question can be answered from two aspects. Firstly, the multi-filter score calculation in the MF-GE system is done only once at the start of the algorithm. This step will not be involved in the genetic iteration and optimization processes. Therefore, it is computationally efficient to incorporate this initial information. Secondly, by closely observing the classification results produced by individual classifiers, we can see that the MF-GE system achieved better classification results in almost all cases than those alternative methods, regardless of which inductive algorithm is used for evaluation. Moreover, such improvement is consistent throughout all datasets used for evaluation. This demonstrates that the gene subsets selected by the MF-GE system have a better generalization property and thus are more informative for unseen data classification. From the biological perspective, the selected genes and gene subsets are more likely to have genuine association with the disease of interest. Hence, they are more valuable for future biological analysis.

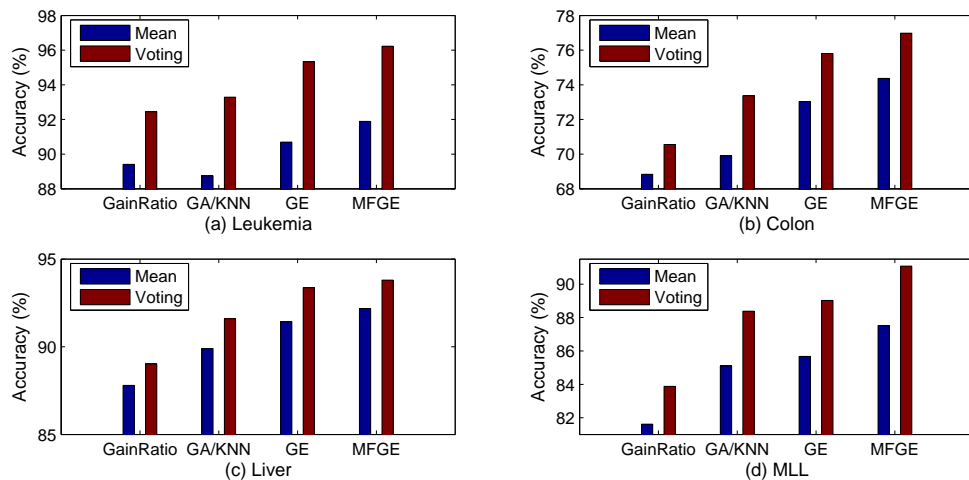


Figure 5.3: The comparison of average classification and majority voting classification of the five classifiers with different gene selection methods in each microarray dataset.

Figure 5.3 gives the comparison of the mean classification accuracy and the majority voting accuracy of these five classifiers with different gene ranking methods in each microarray dataset. In all cases, integrating classifiers with majority voting gives better classification results than the average of individuals. Therefore, majority voting can be considered as a useful classifier integration method for improving the overall classification accuracy. Figure 5.4 depicts the multi-filter scores of the 200 genes pre-filtered

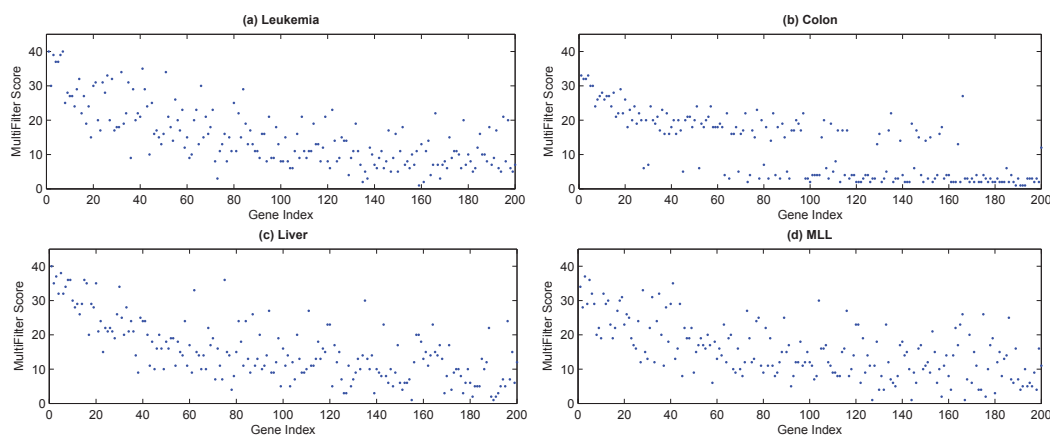


Figure 5.4: The multi-filter consensus scores of the 200 pre-filtered genes.

by BSS/WSS. It is evident that many genes with relatively low BSS/WSS ranking have shown very high multi-filter scores. Interestingly, in the Colon dataset, genes are fractured into two groups with respect to the multi-filter scores. It would be interesting to conduct further study on finding the causality of such inconsistency.

Table 5.7: Generation of convergence & subset size for each dataset using MF-GE and GE.

Dataset	Comparison Criterion	MF-GE	GE	p -value*
Leukemia	Mean Generation of Convergence	21.2	23.4	1×10^{-2}
	Mean Subset Size	4.7	5.4	4×10^{-3}
Colon	Mean Generation of Convergence	25.5	27.1	5×10^{-2}
	Mean Subset Size	6.0	6.6	3×10^{-3}
Liver	Mean Generation of Convergence	27.1	27.4	1×10^{-1}
	Mean Subset Size	7.2	7.7	1×10^{-3}
MLL	Mean Generation of Convergence	25.0	26.1	8×10^{-2}
	Mean Subset Size	6.8	7.2	3×10^{-2}

* p -values are calculated using student t -test with one tail.

The second set of experiments is conducted to compare the mean generation of convergence (termination generation), and the mean gene subset size collected in each iteration of the MF-GE and the original GE hybrid. We formulate these two criteria for comparison because the biological relationship with the target disease is more easily identified when the number of the selected genes is small [55], and a shorter termination generation implies that the method is more computationally efficient.

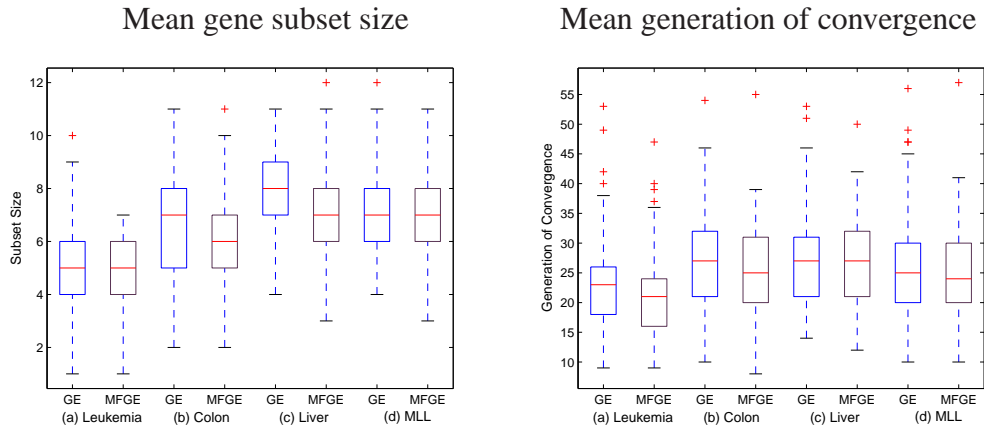


Figure 5.5: Mean gene subset size selected by GE and MF-GE, and mean generation of convergence of GE and MF-GE from each microarray dataset.

As illustrated in Table 5.7, it is clear that the MF-GE system is capable of converging with fewer generations while also generating smaller gene subsets. Specifically, the mean gene subset size given by MF-GE is about 0.4 to 0.7 of a gene less than those of GE, while the mean generation of convergence is about 1 to 2 generations fewer. Essentially, the improvement on producing more compact gene subsets is more significant as demonstrated by the p -value of the one-tail Student t -test. The results are also shown in a boxplot in Figure 5.5. One interesting finding is that these figures indicate a dataset-dependent relationship, that is, the optimal subset size and the convergence of the genetic component is partially determined by the given dataset. Nevertheless, significant improvements can be achieved by fusion of prior data information into the system.

Lastly, in Table 5.8, we list the top 5 genes with the highest selection frequency of each microarray dataset respectively.

Table 5.8: Top 5 genes with the highest selection frequency from each microarray data.

Dataset	Identifier	Gene Description
Leukemia	X95735_at	Zyxin
	M31523_at	TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
	Y07604_at	Nucleoside-diphosphate kinase
	M92287_at	CCND3 Cyclin D3
	M27891_at	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
Colon	Hsa.549	P03001 TRANSCRIPTION FACTOR IIIA
	Hsa.3016	S-100P PROTEIN (HUMAN)
	Hsa.8147	Human desmin gene, complete cds
	Hsa.36689	H.sapiens mRNA for GCAP-II/uroguanylin precursor
	Hsa.6814	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)
Liver	AA232837	Plasmalemma vesicle associated protein (PLVAP)
	AA464192	PDZ domain containing 11 (PDZD11)
	AA486817	Shisa homolog 5 (Xenopus laevis) (SHISA5)
	R43576	Basic leucine zipper nuclear factor 1 (BLZF1)
	H62781	Ficolin (collagen/fibrinogen domain containing lectin) 2 (hucolin) (FCN2)
MLL	33412_at	vicpro2.D07.r Homo sapiens cDNA, 5' end
	1389_at	Human common acute lymphoblastic leukemia antigen (CALLA) mRNA, complete cds
	32847_at	Homo sapiens myosin light chain kinase (MLCK) mRNA, complete cds
	39318_at	H.sapiens mRNA for Tcell leukemia
	40763_at	Human leukemogenic homolog protein (MEIS1) mRNA, complete cds

5.5 Summary

Traditionally, filter and wrapper algorithms are treated as competitors in gene selection for data classification. In this study, we embrace an alternative view and attempt to combine them as the building blocks of a more advanced hybrid system. The proposed MF-GE system applied several novel integration ideas to strengthen the advantages of each component while avoiding their weaknesses. The experimental results indicate the following:

- By fusing evaluation feedbacks of multiple filtering algorithms, the system not only seeks for high classification accuracy of training datasets greedily, but takes into consideration other characteristics of the data. The overfitting problem can then be circumvented and a better generalization of the selected gene and gene subsets can be achieved.
- By weighing the goodness of each candidate gene from multiple aspects, we reduce the chance of identifying false-positive genes while producing a more compact gene subset. This is useful since future biological experiments can be more easily conducted to validate the importance of the selected genes.
- With the use of multiple filtering information, the MF-GE is able to converge more quickly without sacrificing the sample classification accuracy and thus saves computational expense.

The MF-GE system provides an effective measure for incorporating different algorithm components. It allows any filters or classifiers with new or special capabilities to be added to the system and those no longer useful or inappropriate to be removed from the system, based on the data requirements or user preferences.

Chapter 6

A Self-boosted Semi-supervised Learning Algorithm for Post-processing Mass Spectrometry-based Proteomics Data

This chapter is based on the following manuscript:

Pengyi Yang, Jie Ma, Penghao Wang, Yunping Zhu, Bing B. Zhou, Yee Hwa Yang, Improving X!Tandem on peptide identification from mass spectrometry by self-boosted Percolator, IEEE/ACM Transactions on Computational Biology and Bioinformatics, accepted.

6.1 Peptide-spectrum match post-processing

In previous chapters, we have looked at different computational approaches for analyzing large-scale genomic data and transcriptomic data. From this chapter, we turn our attention to the mass spectrometry (MS)-based proteomics, and study proteins—the functional products of genes and transcripts.

One of the main computational challenges in MS-based proteomics is the identification of peptides from the spectra produced by the mass spectrometer. There are three main approaches for peptide identification, the database search approach [62]: the spectral library search approach [45, 109], and the *de novo* sequencing approach [64].

The *de novo* sequencing approach is often only applicable to very high precision mass spectrometry [64] and the remaining two approaches are more common. The library search approach relies on the initial results from the database search, and the *de novo* sequencing approach can benefit from incorporating database search results [14]. Thus, improvement on the database search approach will also enhance the library search approach and the *de novo* sequencing approach. This suggests that it is important that our initial focus for improving peptide identification results is to concentrate on achieving better and more efficient database search results.

In the database search approach, a search algorithm is applied to produce a list of peptide-spectrum matches (PSMs), in which the peptides and proteins are inferred. Popular database search algorithms include SEQUEST [62], MASCOT [150], X!Tandem [44], OMSSA [66], and Paragon [179]. Several studies have reviewed and compared their performance on different datasets [10, 97].

All these algorithms involve comparing observed spectra to a list of theoretical enzymatic digested peptides from a specified protein database. The comparison is based on a “search score” measuring the degree of agreement between the observed spectra to a theoretical spectrum generated from enzymatic digested peptide. Each pair of observed spectra and a theoretical peptide is known as a peptide spectrum match (PSM). Each PSM is assigned a search score and different algorithms vary in their definition of the score. For example, SEQUEST calculates an Xcorr score for each PSM by evaluating the correlation between the experimental spectrum and the theoretically constructed spectrum from the database [62]; X!Tandem [44] counts the number of matched peaks and then calculates a score using the matched ions and their intensities.

Each search score is an indication of the quality of match between the theoretical peptides and the observed spectra. One typically expects that the higher the score, the more likely that the PSM is a correct match, that is, the observed spectrum is correctly identified as the corresponding peptide of the PSM. Due to the varying quality of the spectra, the characteristics of the search algorithm and scoring metrics, and the incompleteness of the protein database, typically, only a fraction of the PSMs are correct [141]. Moreover, the search scores are often not directly interpretable in terms of statistical significance [95]. Therefore, it is necessary to determine a critical value above which ranking scores are to be considered significant. This filtering process is also seen as an independent validation of the PSM and thus the whole process is often known as PSM post-processing.

For PSM post-processing, algorithms such as PeptideProphet [101] and Percolator [94] are probably the most popular ones. PeptideProphet learns a linear discriminant analysis (LDA) classifier from database search results and fits an expectation maximization (EM) model from which a posterior probability for each PSM being a correct peptide identification is generated. Percolator uses a semi-supervised learning (SSL) algorithm for training a support vector machine (SVM) iteratively. The training data is filtered subsequently with a predefined false discovery rate (FDR) threshold, and the SVM model from the last iteration is used for classifying PSMs.

Both Percolator and PeptideProphet were originally designed for SEQUEST [94, 101]. Recent extensions to PeptideProphet include the incorporation of more flexible models (e.g. variable component mixture model) [35] and other database search algorithms [51]. In comparison, the extensions of Percolator include a wrapper interface for MASCOT [23], and the reformulation of the learning algorithm [183].

While these validation and filtering algorithms have been found to be very useful, they are predominantly designed for commercial database search algorithms i.e. SEQUEST and MASCOT. So far, there has been no extension of Percolator for open source search algorithms such as X!Tandem. Therefore, it is highly desirable to extend and optimize these PSM post-processing algorithms for open source algorithms, given their increasing popularity in the proteomics community [51].

In this chapter, we describe a self-boosted Percolator for post-processing X!Tandem search results. We discover that the current Percolator algorithm relies heavily on decoy PSMs and their rankings in the initial PSM list [23]. The iterative FDR filtering of PSMs is the key to enhance the discriminant ability of the final SVM model. If the decoy PSMs are poorly ranked in the initial PSM list, the performance of the algorithm may degrade, resulting in a suboptimal SVM model and reduced PSM classification accuracy. One potential solution could be to apply the SVM model from Percolator to re-rank the PSM list and re-run Percolator on the re-ranked PSM list.

We implement such a cascade learning procedure for the original Percolator algorithm. By repeating the learning and re-ranking process a few times, the algorithm “boosts” itself to a stable state, overcoming the poor initial PSM ranking and identify more PSMs which translate into more protein identifications. We integrated the self-boosted Percolator with ProteinProphet [140] in Trans-Proteomic Pipeline (TPP) [51] by generating PSM filtering results in a ProteinProphet readable format. With such an integration, the proposed algorithm can be used conveniently as a key component in

large-scale protein identification.

6.2 Experiment settings and implementations

6.2.1 Evaluation datasets

Several large-scale proteomics datasets generated by mass spectrometry experiments are publicly available and commonly used for algorithmic validations [126]. The first is a Universal Proteomics Standard (UPS) Set (**UPS1**). This dataset contains the tandem MS spectra of 48 known proteins generated by the LTQ mass spectrometer. The corresponding target database for database searching is the human specific protein sequences extracted from the SWISS-PROT sequence library (release-2010_05), and the decoy database is generated by reversing the sequences of the entries in the target database. Another two complex sample datasets [94] are also included for evaluation and they are known as the **Yeast** dataset and the **Worm** dataset (refer to Supplement of [94] for details). Specifically, we utilize the datasets generated from trypsin digestion. The corresponding target databases are obtained from the authors (<http://noble.gs.washington.edu/proj/percolator>) and the decoy databases are built by reversing the sequences in the target databases, respectively.

6.2.2 Database searching

We use the concatenated target-decoy database search approach, in which the reverse protein sequences are combined with the target database [61]. The estimated false discovery rate (FDR) is calculated as follows:

$$\text{FDR} = 2 \times \frac{N_D}{N_D + N_T} \quad (6.1)$$

where N_D and N_T are the number of decoy and target matches from the concatenated database, respectively, which pass the predetermined filtering threshold. The q -value is defined as the minimal FDR at which a PSM is accepted. For the control dataset of UPS1, the actual FDR is defined as follows and can be directly calculated using known proteins [23]:

$$\text{FDR}_{\text{Actual}} = \frac{N_{\text{FP}}}{N_{\text{T}}} \quad (6.2)$$

where N_{FP} is the number of false positive identifications from the total target assignments N_{T} that do not match to the control proteins.

Raw spectra files were searched against the concatenated database using X!Tandem (2009.10.01.1 from TPP v4.4). The average mass was used for both peptide and fragment ions, with fixed modification (Carbamidomethyl, +57.02 Da) on Cys and variable modification (Oxidation, +15.99 Da) on Met. Tryptic cleavage at Lys or Arg only was selected and up to two missed cleavage sites were allowed. The mass tolerance for precursor ions and fragments were 3.0 Da and 1.0 Da for all datasets.

6.2.3 Percolator for X!Tandem search results

We extend Percolator for filtering X!Tandem search results. Specifically, Percolator extracts a set of discriminant features from the data and each PSM is represented as a vector \mathbf{x}_i and a class label $y_i (i = 1, \dots, M)$ where M is the total number of PSMs. Each component in \mathbf{x}_i is a feature $x_{ij} (j = 1, \dots, N)$ interpreted as the i^{th} feature of the j^{th} PSM, where N is the dimension of the feature space.

A linear SVM with a soft margin is trained to generate a credibility score for each PSM. Linear SVMs with a soft margin are robust tools for data classification [13]. The hyperplane in SVM is formed by optimizing the following objective function with constraints:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \mathcal{C} \sum_{i=1}^M \xi_i$$

$$\text{subject to : } y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle) + b \geq 1 - \xi_i$$

where \mathbf{w} is the weight vector, ξ_i are slack variables that allow misclassification, \mathcal{C} determines the penalty of misclassification, and b is the bias.

The key component in Percolator is to label each PSM so as to train a SVM. Since we do not know *a priori* which PSMs are correct/incorrect identifications, a target-decoy approach is used to construct positive and negative PSMs for SVM training. Particularly, a subset of PSMs regarded as “correct identifications” from the target database

are used as positive training examples while all PSMs from the decoy database are used as the negative examples. In order to build a high-quality training dataset, the Percolator algorithm attempts to iteratively remove potential false positive identifications from the target database (**Algorithm 3**). This is done by calculating a FDR in each iteration and removing the target hits that appear below the expected FDR threshold (**Algorithm 4**).

Algorithm 3 Percolator

```

1: Input: PSM list  $L$ 
2: Output: PSM probability list  $L'$ 
3: while number of removed target PSMs  $> 0$  do
4:    $D = \text{getTrainSet}(L)$ ;
5:    $\text{svm} = \text{trainSVM}(D)$ ;
6:    $L = \text{probability}(\text{svm}, L)$ ;
7: end while
8: // use the SVM model from the last iteration to re-classify PSM list
9:  $L' = \text{probability}(\text{svm}, L)$ ;
10: return  $L'$ ;

```

From X!Tandem’s search results, we extract 14 features for training SVM in Percolator. Table 6.1 summarizes the features used by our Percolator for X!Tandem. These features are selected according to previous studies on Percolator for SEQUEST and MASCOT [23, 94]. Particularly, these features are evaluated and well supported by Käll *et al.* (see Supplementary Table 1 in [94] for details).

Table 6.1: Summary of features used by Percolator for X!Tandem search results.

Feature	Description
Hyperscore	the first Hyperscore reported by X!Tandem
Δ score	the difference between the first Hyperscore and the second score
expect	the expectation reported by X!Tandem
$\ln(\text{rHyper})$	the natural logarithm of the rank of the match based on the Hyperscore
mass	the observed monoisotopic mass of the identified peptide
Δ mass	the difference in calculated and observed mass
$\text{abs}(\Delta\text{mass})$	the absolute value of the difference in calculated and observed mass
ionFrac	the fraction of matched b and y ions
enzN	a Boolean value indicating if the peptide is preceded by a tryptic site
enzC	a Boolean value indicating if the peptide has a tryptic C-terminus
enzInt	the number of missed internal tryptic sites
pepLen	the length of the matched peptide, in residues
charge	the predicted charge state of the peptide
$\ln(\text{numProt})$	number of times the matched protein matches other PSMs

Algorithm 4 getTrainSet

```

1: Input: PSM list  $L$ 
2: Output: train set  $D$ 
3:  $positives = \emptyset$ ;
4:  $negatives = \emptyset$ ;
5:  $p = 0$ ; // a pointer that go through the PSM list
6: while  $FDR < 0.01$  do
7:    $p = p + 1$ ;
8:   if  $L[p] \in targets$  then
9:     // PSM is from target database, collect it as positive examples
10:     $positives = positives \cup L[p]$ ;
11:   else
12:     // PSM is from decoy database, collect it as negative examples
13:     $negatives = negatives \cup L[p]$ ;
14:   end if
15:    $FDR = \text{getCurrentFDR}(positives, negatives)$ ;
16: end while
17: // collect the rest of decoy matches as negative examples
18: while  $L[p] \neq null$  do
19:    $p = p + 1$ ;
20:   if  $L[p] \in decoys$  then
21:      $negatives = negatives \cup L[p]$ ;
22:   end if
23: end while
24:  $D = \text{createTrainSet}(positives, negatives)$ ;
25: return  $D$ ;

```

Following the same configuration as in Percolator for SEQUEST and MASCOT [23, 94], we implemented the iterative PSM filtering procedure (**Algorithm 3** and **4**). The result of Percolator is a list of PSM scores reported by the trained SVM model from the last iteration.

6.2.4 Semi-supervised learning on creating training dataset

In Percolator, the training set is built by removing ambiguous PSMs from the target database using a FDR threshold (**Algorithm 4**). However, since the FDR is estimated by using PSMs from the decoy database, the rankings of the decoy PSMs determine how many PSMs from the target database will be removed and which of them will be used as positive training examples in each iteration.

As an example, assume that the PSM list in Figure 6.1a is the initial ranking using

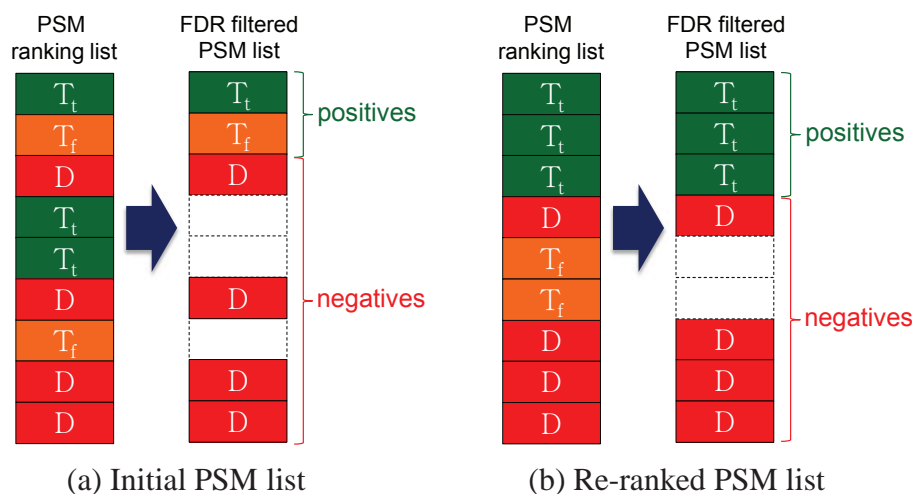


Figure 6.1: Schematic illustration of PSM rank effect on creating training dataset. (a) Initial PSM list ranked by search score from database search algorithm. (b) A re-ranked PSM list by, e.g. PeptideProphet. T_t and T_f are true positive and false positive identifications from target database. D denotes identification from decoy database. Empty rectangles indicate that the corresponding PSM is removed after FDR filtering.

PSM search scores of a database search algorithm whereas the PSM list in Figure 6.1b is the re-ranking after further processing. Identifications from the target database are denoted as “T”, from which true positive identifications and false positive identifications are denoted as “ T_t ” and “ T_f ”, respectively. Any identification from the decoy database is denoted as “D”. In both cases (Figure 6.1a,b), by estimating FDR (Equation 6.2.2) and using any threshold smaller than 0.5, we will remove any PSMs from the target database that appear below one or more PSMs from the decoy database. Therefore, the resulting training set from Figure 6.1a includes only two positive training examples where one of them is a false positive identification that will be treated incorrectly by SVM as a positive example. In contrast, the resulting training set from Figure 6.1b includes three positive training examples and all of them are true identifications.

In this study, we evaluate the number of PSMs included for SVM training using the control dataset of UPS1 and two complex proteomics datasets of Yeast and Worm. The FDR threshold of 0.01 is used for PSM filtering in each iteration.

6.2.5 Self-boosted Percolator

As described above, the SSL algorithm used by Percolator for SVM training is sensitive to the initial PSM ranking list. That is, a poor initial ranking will have a reduced number

of target PSMs passing the predefined FDR filtering threshold, causing an under representation of positive training examples. This under representation of positive training examples persists through the iteration of the training process since once a target PSM is removed by FDR filtering, it will not be considered in follow up interactions.

One way to overcome this inefficiency is to repeat the Percolator training and filtering process multiple times each on the PSM ranking list generated in its previous runs. The assumption is that if Percolator could improve the ranking of PSMs, then by each time repeating the Percolator training on the PSM ranking list generated in its previous run, we can obtain more target PSMs with potentially less false positives. We call this cascade learning procedure “self-boosting” and the algorithm “self-boosted Percolator” (**Algorithm 5**).

Algorithm 5 Self-boosted Percolator

```

1: Input: Initial PSM list  $L$ , number of boost runs  $b$ 
2: Output: PSM probability list  $L'$ 
3: while  $b > 0$  do
4:    $L = \text{Percolator}(L)$ ;
5:    $b = b - 1$ ;
6: end while
7: // record the ranking list from the last boost run
8:  $L' = L$ 
9: return  $L'$ ;

```

6.2.6 Performance comparison on PSM post-processing

For PSM filtering, we compare the performance of self-boosted Percolator with PeptideProphet and the original Percolator algorithm. The results from the database search algorithms (without further processing) are used as the baselines. Specifically, we calculate the number of accepted PSMs reported by each PSM filtering algorithm with respect to the estimated FDR (denoted as q -value) threshold ranging from (0, 0.2]. Since the proteins are known beforehand in UPS1 dataset, we used it to verify whether the q -value reported by each PSM filtering algorithm resembles the actual FDR. This is done by directly calculating the actual FDR (Equation 6.2.2) for the UPS1 dataset using the known proteins and comparing it with the q -value. For PeptideProphet, we used TPP v4.4 [100]. The database search outputs from X!Tandem are preprocessed

by `msconvert.exe` to generate `mzXML` files for running `PeptideProphet`. For `Percolator`, the self-boosted `Percolator` is run with the boost runs set to 1. This, in essence, is equivalent to the original implementation of the `Percolator` algorithm in `MASCOT` and `SEQUENT`.

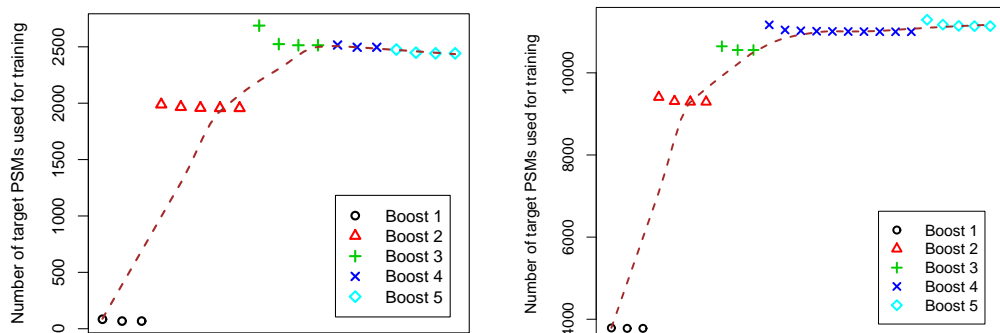
For protein identification, we compared the combinations of (1) self-boosted `Percolator` + `ProteinProphet`, and (2) `PeptideProphet` + `ProteinProphet`. We only included PSMs that passed FDR of 0.01 filtering for protein inference, and the FDR is recalculated on the protein level using the same equation as for PSM filtering.

6.3 Results and discussion

6.3.1 Percolator is sensitive to PSM ranking

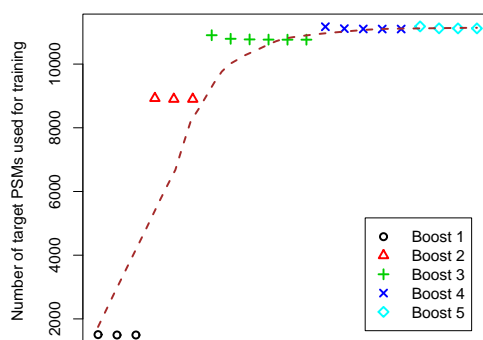
We evaluate the number of target PSMs included in each boost run of `Percolator`. Figure 6.2a shows the result from the `UPS1` dataset. As can be seen, in the first boost run, very few target PSMs are included as positive training examples. The number increases to ~ 2000 in the second boost run and plateaus at ~ 2500 in the third, fourth, and fifth boost runs. For the `Yeast` dataset (Figure 6.2b), `Percolator` starts with less than 4000 target PSMs and plateaus at $\sim 11,000$ target PSMs. A similar pattern is observed from the `Worm` dataset (Figure 6.2c), where less than 2000 target PSMs are included for training in the first boost run and more than 10,000 target PSMs are included for training in the last boost run. Notice that FDR is controlled at the same level (i.e. 1%) among each boost run. These results suggest that the original `Percolator` algorithm is sensitive to the initial PSM ranking, and the self-boosted `Percolator` is able to overcome this inefficiency by extracting increasingly more target PSMs from each boost run for SVM model training and PSM re-ranking.

In Figure 6.2, multiple iterations of filtering within each boost run are denoted by points with the same shape. Within each boost run, target PSMs are filtered iteratively by a predefined FDR threshold (1% in our experiments). It is clear that within each boost run, the SSL algorithm of `Percolator` generally converges after a few iterations. Note that the iterative filtering of SSL does not increase the number of target PSMs for SVM training.



(a) Self-boosting on UPS1 dataset

(b) Self-boosting on Yeast dataset

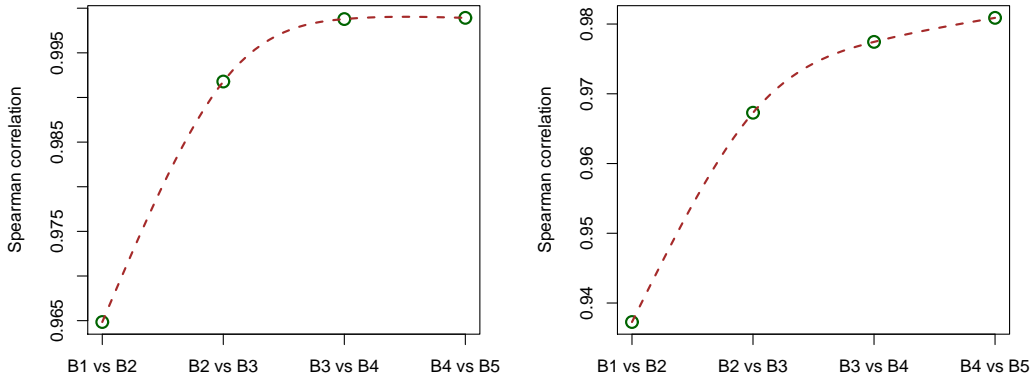


(c) Self-boosting on Worm dataset

Figure 6.2: Self-boosting of Percolator on (a) UPS1 dataset, (b) Yeast dataset, and (c) Worm dataset. For each dataset, 5 boost runs are conducted. Within a boost run, FDR filtering iterations are denoted by points with the same shape. For each dataset, a locally weight regression line is fitted to all points.

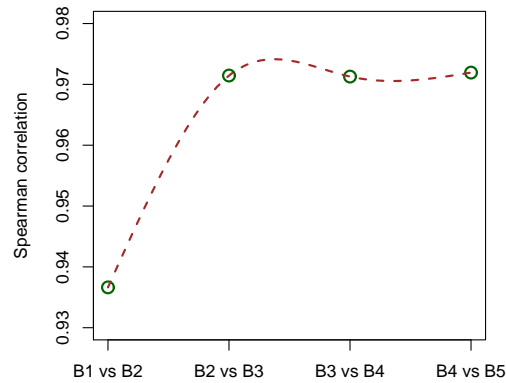
6.3.2 Determining the number of boost runs

We investigate the number of boost runs required for self-boosted Percolator to produce stable PSM filtering results. This is done by calculating a Spearman correlation of the PSM rankings from each boost run with its previous boost run. Figure 6.3 shows the results. By linear extrapolation, the Spearman correlation appears to plateau after the



(a) Correlation of boost runs on UPS1

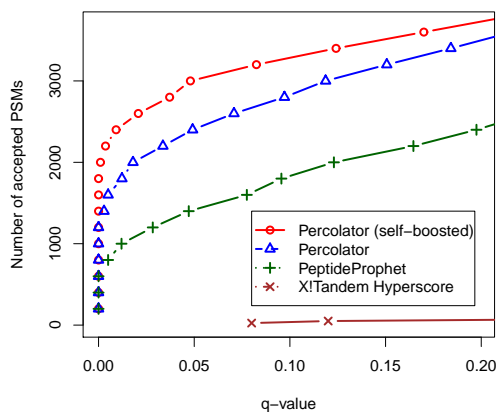
(b) Correlation of boost runs on Yeast



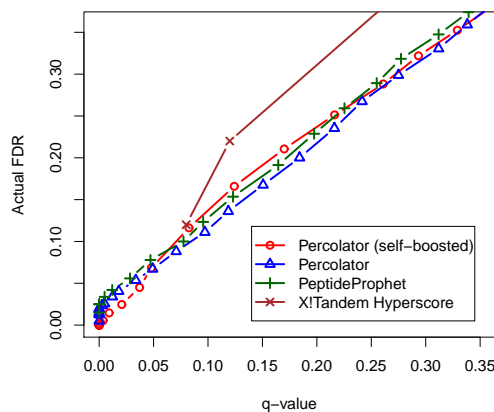
(c) Correlation of boost runs on Worm

Figure 6.3: Spearman correlations of PSM rankings from each boost run with its previous boost run for (a) UPS1 dataset, (b) Yeast dataset, and (c) Worm dataset. For each dataset, a linear extrapolation line is fitted to the points.

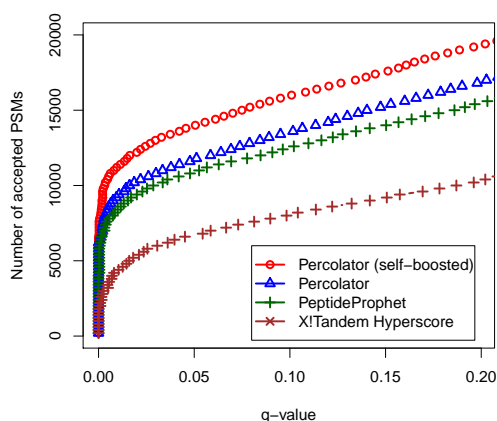
fifth boost run in all three datasets. Therefore, it is evident that five boost runs are sufficient for self-boosted Percolator to reach the stable state. The subsequent experiments are conducted with boost runs set to 5.



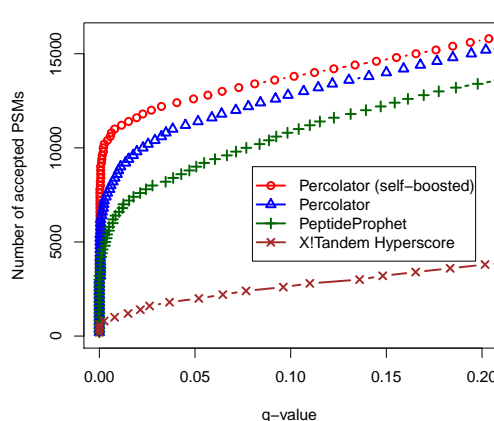
(a) PSM identification on UPS1 dataset



(b) FDR estimation UPS1 dataset



(c) PSM identification Yeast dataset



(d) PSM identification Worm dataset

Figure 6.4: The number of accepted PSMs is determined at each q -value threshold on X!Tandem search results using X!Tandem modified Hyperscore, PeptideProphet, Percolator without self-boosting, and self-boosted Percolator. (a) UPS1 dataset. (b) The estimated q -value is plotted against the FDR as reported by the UPS1 dataset. (c) Yeast dataset. (d) Worm dataset.

6.3.3 PSM post-processing

The motivation of extracting more target PSMs through self-boosting is to create a more robust and accurate PSM filtering model which could lead to the identification of more PSMs without sacrificing FDR. Figure 6.4 shows the performance of self-boosted Percolator in comparison with PeptideProphet and Peculator without self-boosting. We observe that in all three datasets self-boosted Percolator identifies consistently more

PSMs at any given q -value thresholds. The improvement is significant compared to PeptideProphet and Percolator without self-boosting. In general, the performance of Percolator (without self-boosting) is better than PeptideProphet. This is consistent with the result obtained by Käll *et al.* [94]. In all cases, using the raw score of X!Tandem for PSM filtering gives low sensitivity. This implies that the self-boosted Percolator is robust to the noise of initial PSM ranking and can fully recover the performance of Percolator without self-boosting.

To verify whether the estimated FDR (q -value) reported by each PSM filtering algorithm resembles the actual FDR, the FDR_{Actual} is calculated using the UPS1 dataset with known proteins and plotted against the q -value (Figure 6.4b). All lines after PSM validation and filtering are approximately straight along the 45-degree lines; this indicates that PeptideProphet, Percolator, and self-boosted Percolator can provide a fairly accurate FDR estimation. The FDR estimated directly based on X!Tandem Hyperscore alone deviated from the actual FDR substantially.

6.3.4 Protein identification

The post-processing results from PeptideProphet and self-boosted Percolator are filtered by controlling PSM level FDR at 0.01. Then ProteinProphet from TPP is used to infer proteins using the PSMs that passed FDR filtering. Figure 6.5 compare the results from using PeptideProphet with ProteinProphet for protein identification with using self-boosted Percolator with ProteinProphet for protein identification. It is clear that in most cases, the combination of self-boosted Percolator and ProteinProphet gives more protein identifications, and the proteins identified by using results from self-boosted Percolator have many more PSMs assigned to.

6.4 Summary

Database searching is a key step in protein identification from MS-based proteomics. The post-processing of database search results is critical for quality control where spurious identifications are removed, while only informative PSMs are reserved for protein inference. In this chapter, we look at the post-processing of X!Tandem database search results. X!Tandem is an open source database search algorithm. However, unlike commercial database search softwares, X!Tandem is not well supported by sophisticated

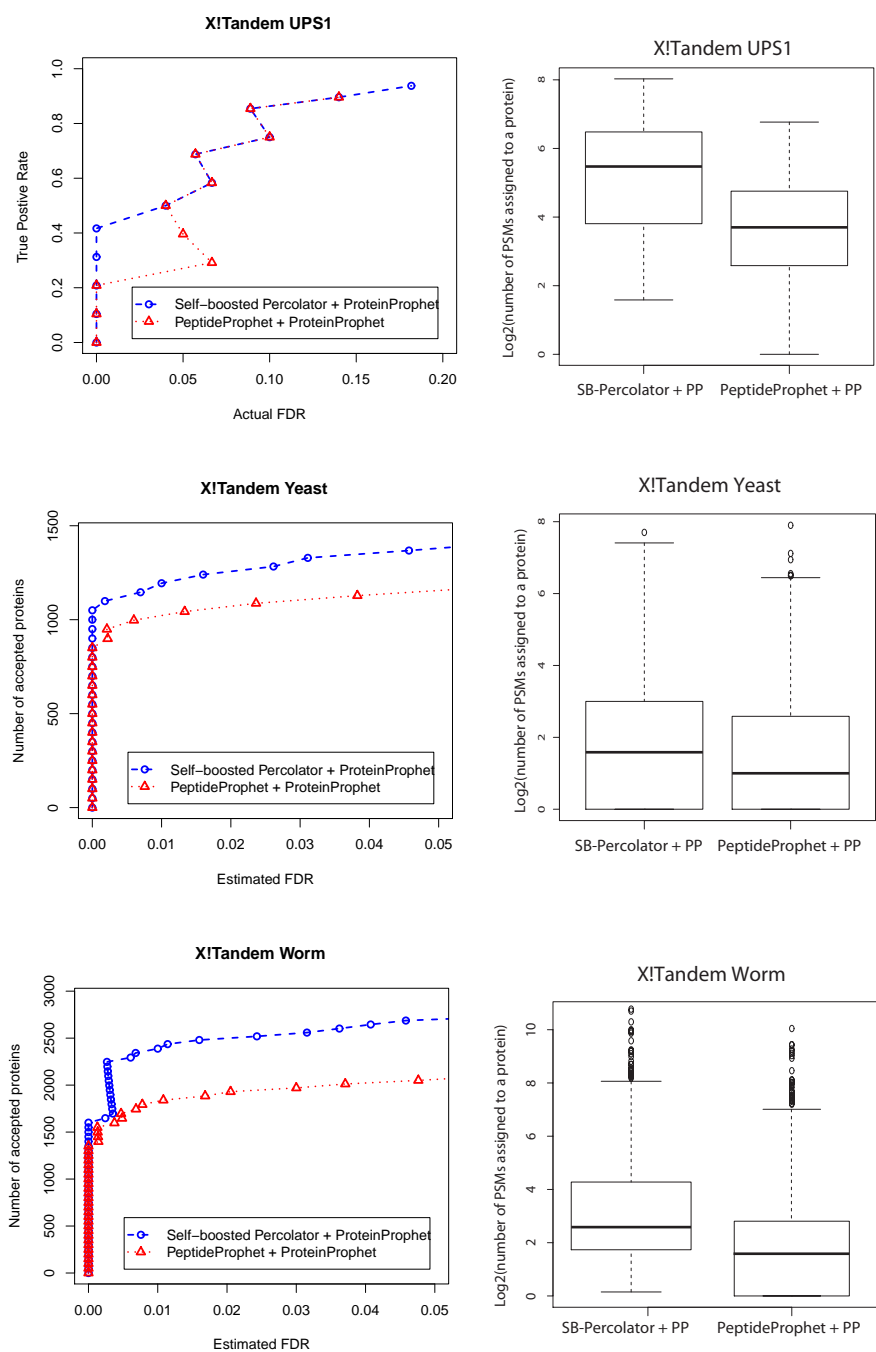


Figure 6.5: The number of accepted proteins is determined at different FDR thresholds on X!Tandem search results using the combination of PeptideProphet + ProteinProphet (or “PeptideProphet + PP”) and self-boosted Percolator + ProteinProphet (or “SB-Percolator + PP”). The Boxplot on the right hand side show the number of PSMs assigned to each protein.

post-processing algorithms such as Percolator. For this reason, we extend the Percolator algorithm for post-processing X!Tandem search results.

In addition, we found that the learning procedure used by Percolator relies heavily on the guidance of the decoy PSMs and their ranking among target PSMs. The iterative FDR filtering of PSMs is the key to enhance the discriminant ability of final SVM models. If the decoy PSMs are poorly ranked in the initial PSM list, the performance of the SVM model may degenerate. We propose to overcome the inefficiency of the original Percolator algorithm by using a cascade learning approach where the performance is boosted by using the PSM ranking from the previous boost run as the input of the next boost run. The consistent improvement of performance on a benchmark dataset and two complex sample datasets indicates that the proposed self-boosted Percolator is effective for improving X!Tandem on peptide and protein identification from tandem mass spectrometry.

In conclusion, we proposed a self-boosted Percolator algorithm for post-processing X!Tandem search results and intergraded it with ProteinProphet in TPP. X!Tandem is open source software, but not originally supported by either PeptideProphet or Percolator. With our new self-boosted Percolator package freely provided to the research community, proteomics researchers can now set up a complete commercial free software pipeline for mass spectrometry analysis.

6.5 Software availability

The self-boosted Percolator package is freely available from:

<http://code.google.com/p/self-boosted-percolator>

Chapter 7

A Clustering-Based Hybrid Algorithm for Extracting Complementary Biomarkers From Proteomics Data

This chapter is based on the following publication:

Pengyi Yang, Zili Zhang, Bing B. Zhou, Albert Y. Zomaya, A clustering based hybrid system for biomarker selection and sample classification of mass spectrometry data. Neurocomputing, 73:2317-2331, 2010

7.1 Biomarker discovery from MS-based proteomics data

In the previous chapter, we described the post-processing of PSMs for quality control of mass spectrometry search results. In this chapter, we look at the method for extracting key protein sets that will be used for disease and control classification.

Compared to gene profiling using microarray technologies, MS-based proteomics enables a more direct proteome-level view of the cellular functionality and pathogenesis. According to the types of the data, a biomarker could be defined as a protein, a peptide, or a mass-to-charge (m/z) ion ratio. Here we refer to them collectively as proteomic biomarkers. The quantification of a proteomic biomarker could be performed by using isotopic or isobaric labelling such as stable isotope labeling with amino acids

in cell culture (SILAC) [144] and isobaric tag for relative and absolute quantitation (i-TRAQ) [167], or by a label-free approach where the spectrum counts [119] or spectrum intensity [143] can be used as the estimation of abundance. The goal is to select a set of proteomic biomarkers that jointly distinguish disease and normal samples.

Similar to microarrays in case-control studies, MS-based proteomics datasets are plagued by the curse-of-dimensionality and curse-of-data-sparsity [182]. Without intensive feature filtering or dimension reduction, standard supervised classification algorithms cannot be properly employed [114]. Clearly, most of the common feature selection approaches that are used in microarray data analysis could also be applied to MS data. This is reviewed by Hilario and Kalousis [84].

7.2 Feature correlation and complementary feature selection

One of the key findings in previous experience with microarray data analysis is that aggressive feature reduction using a filter-based approach may lead to the selection of highly correlated features [90]. This is because filter-based algorithms commonly evaluate each feature individually, and features selected in this way often have high correlation with each other, limiting the extraction of complementary information. Under the assumption that genes with high correlations could potentially belong to the same biological pathway, if a disease-associated pathway has a large number of genes involved, the gene selection results may be dominated by such a pathway, while other informative pathways will be ignored [28].

As the central dogma indicates, proteins are the functional products of genes expressed in certain time and conditions. Therefore, MS datasets may have similar properties as microarray datasets with many correlated m/z features could possibly come from several dominated pathways. If this assumption is true, the selection of m/z biomarkers may also be hampered by issues such as highly correlated features. In order to take other informative pathways into account, special strategies must be employed to generate a redundancy-reduced and information-enriched feature selection result. Such procedures are aimed at facilitating the followup sample classification and biomarker validation.

Clustering algorithms has been demonstrated to be useful for reducing correlation

in feature sets. Specifically, Hanczar *et al.* [80] proposed a prototype-based feature extraction procedure for microarray data analysis. In their algorithm, a k -means clustering procedure is applied to the initial microarray dataset to cluster the genes with similar expressions. Then the mean expression level of a group of genes is calculated and used as the “prototype” gene for the followup classification process. However, the “prototype” genes are the transformed feature vectors that compounded the biological interpretation. Furthermore, the algorithm uses all “prototype” genes, each from a different cluster, for sample classification, but it is most unlikely that all “prototype” genes are relevant to the disease of interest. This inevitably introduces undesired redundancy, and potentially affects the classification results.

Wang *et al.* [196] applied a hierarchical clustering hybrid algorithm for gene selection from microarrays. Their method firstly ranks 50 to 100 genes using a given filter algorithm and then uses a hierarchical clustering algorithm to produce a dendrogram with these top-ranked genes. Key genes are selected by cutting the dendrogram into pieces at different levels and selecting a representative for each piece. This procedure is exhaustively investigated from the bottom to the top of the dendrogram to select the best feature subset in a wrapper manner. Due to the intensive computations, this hybrid algorithm suffers from scalability problem. A prefiltering of 50 to 100 genes potentially restricts its ability to include as much pathway information as possible.

Another recent study applied a similar idea for selecting discriminative genes for multi-class microarray data analysis [28]. The gene ranking and gene clustering processes are conducted independently, and the final gene sets are determined by using gene ranking and clustering information collaboratively. However, the experimental results across four datasets illustrated that the classification accuracy increases almost monotonically with the increase of the gene size used for classification. In order to achieve the highest classification accuracy, the number of genes used for classification has to be very large. These results indicate that the essential pattern of the datasets is still not well captured.

Built on previous studies on microarray data analysis, we proposed a k -means clustering-based hybrid system for MS data analysis [204]. Our hybrid algorithm utilizes a k -means clustering-based feature extraction and selection procedure to bridge the filter selection algorithm and the genetic ensemble algorithm, as used in our SNP and microarray data analyses in Chapter 4 and Chapter 5. We named this hybrid algorithm FCGE short for “filtering, clustering, and genetic ensemble selection”. It combines

the advantages of both filter and wrapper algorithms while also incorporating the extra benefits from clustering-based correlation reduction and information enrichment. By implementing an iterative procedure, the proposed system is robust to random initialization and able to automatically stabilize the feature selection results.

7.3 A clustering-based hybrid approach

Here we define a m/z ion ratio as a proteomic feature from MS data, and the goal is to select a set of m/z features that jointly distinguishes disease and normal samples. The selected m/z features are the potential biomarkers for the disease of interest. Nevertheless, the system could also be applied to MS/MS data where the definition of feature could be a peptide or protein.

Figure 7.1 illustrates the proposed system. It executes the following steps:

- A filter-based m/z feature ranking algorithm is utilized to prefilter the potential m/z biomarkers, by ranking m/z features according to their goodness in sample discrimination.
- After the prefiltering step, k -means clustering is conducted on the prefiltered sets to group the m/z features with similar intensity across different samples into clusters; m/z features within the same cluster will have higher correlation to those in a different cluster.
- The mean intensity pattern of each cluster is calculated, and a m/z feature with the most similar intensity pattern to the mean intensity pattern, as well as a m/z feature with the most different intensity pattern to the mean intensity pattern, are selected as the representatives of each cluster.
- The genetic ensemble wrapper is then invoked to further minimize feature redundancy by identifying the most informative representatives while discarding the uninformative ones, guided by the sample classification accuracy of an internal cross-validation.
- Steps 2-4 are repeated multiple times (30 in our experiments) and the selected highly differential m/z features are collected and ranked by their selection frequency.

- Finally, a ranking list of m/z features is obtained and the top ranked m/z features that are regarded as the most informative biomarkers to the essential pattern of the underlying dataset are evaluated in unseen data classification.

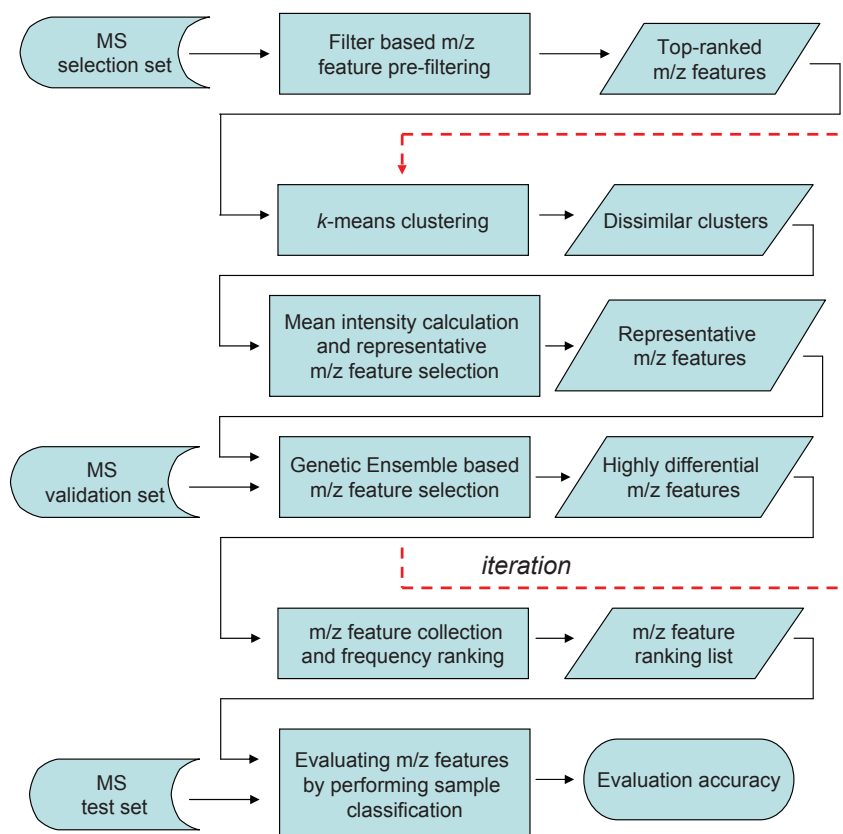


Figure 7.1: The overall work flow of the FCGE hybrid system.

Particularly, the iterative procedure of FCGE overcomes the instability of the k -means clustering and genetic ensemble selection because the clustering procedure is repeated with different initialization and the selection results are not determined by a single run of the system but averaged and ranked by their relative importance to the sample classification in multiple runs. **Algorithm 6** summarizes the above steps in pseudocode; m/z feature evaluation is excluded from the main loop since it is independent from the feature selection procedure.

Algorithm 6 FCGE main loop

```

1: Input: selectionData
2: Output: rankList
3: preSet =  $\emptyset$ ;
4: for  $i=1$  to numFeature do
5:   filteringScorei = filterEvaluation(selectionData, i);
6:   if filteringScorei > cutoff then
7:     preSet = preSet  $\cup$   $i$ ;
8:   end if
9: end for
10:  $k$  = setClusterSize();
11: resultSet =  $\emptyset$ ;
12: for  $i=1$  to iteration do
13:   clusterSet = clustering(preSet, k);
14:   representativeSet =  $\emptyset$ ;
15:   for  $j=1$  to  $k$  do
16:     representativej = selectClusterRepresentative(clusterSet, j);
17:     representativeSet = representativeSet  $\cup$  representativej;
18:   end for
19:   selectSeti = geneticEnsembleSelect(representativeSet);
20:   resultSet = resultSet  $\cup$  selectSeti
21: end for
22: rankList = rank(resultSet);

```

7.3.1 Filter-based prefiltering

It is widely agreed that, of the several tens of thousands of candidates in MS dataset, only a small portion of m/z features are disease-related biomarkers [172]. Thus, a prefiltering process can help us to eliminate the unrelated features that will exert considerable computational burden if included. However, the main concern here is to ensure that the reduction is carried out without sacrificing any critical information. Here we evaluate two filtering algorithms for our hybrid composition: the between-group to within-group sum of square (BWSS) algorithm [56] and the χ^2 -test.

Given a data matrix with m samples, n m/z features, and c classes, the goodness of a m/z feature j is evaluated as follows using BWSS:

$$BWSS(j) = \sum_{i=1}^m \sum_{l=1}^c \frac{I(y_i = l)(\bar{x}_{\cdot j}^{(l)} - \bar{x}_{\cdot j})^2}{I(y_i = l)(x_{ij} - \bar{x}_{\cdot j}^{(l)})^2}, \quad (x \in \mathcal{R}^{m \times n}) \quad (7.1)$$

where $I(\cdot)$ is the indicator function, i is the sample index, and y_i is the class label of sample i . x_{ij} is the value of the j th m/z feature in the i th sample, while $\bar{x}_{\cdot j}$ and $\bar{x}_{\cdot j}^{(l)}$ are the average value of m/z feature j across all samples and across samples belonging to class l only, respectively.

When used for feature evaluation, χ^2 -test can be considered to calculate the occurrence of a particular value of a feature and the occurrence of a class associated with this value. Formally, the discriminative power of a m/z feature j is quantified as follows:

$$\chi^2(j) = \sum_{v \in V} \sum_{i=1}^m \sum_{l=1}^c \frac{I(y_i = l)(O(x_{ij} = v) - E(x_{ij} = v))^2}{I(y_i = l)E(j = v)}, \quad (x \in \mathcal{R}^{m \times n}) \quad (7.2)$$

where j has a set of possible values denoted as $v \in V$, and $O(x_{ij} = v)$ and $E(x_{ij} = v)$ are the observed and the expected co-occurrence of $x_{ij} = v$, respectively. Other notations are as those defined above.

Initial tests find that the prefiltering size of one fifth of the total feature size (around 3000 for typical low-resolution MS datasets) is large enough to capture most differential features while also suitable for the k -means algorithm to work with [204]. Therefore, we apply the above two filtering algorithms to prefilter each dataset with one fifth of the total m/z features, respectively.

7.3.2 k -means clustering

The k -means clustering algorithm is an important component in our hybrid system. The main purpose of applying k -means clustering is to reduce the feature correlation and redundancy. This goal is achieved by clustering m/z features with similar intensity patterns, while also increasing the dissimilarity among different clusters by using a given measure of similarity and cluster mean. In our hybrid system, a k -means clustering with Euclidean distance is employed to compute the similarity. Formally, given two m/z features j and k , the distance value of $d(j, k)$ is computed as follows:

$$d(j, k) = \sum_{i=1}^m \sqrt{(x_{ij} - x_{ik})^2}, \quad (x \in \mathcal{R}^{m \times n}) \quad (7.3)$$

where i is the sample index, x_{ij} is the value of the j th m/z marker of the i th sample, and x_{ik} is the value of k th m/z marker of the i th sample.

The first challenge of applying the k -means clustering algorithm is that different initial partitions of the dataset can result in different clustering outcomes. This can be overcome by clustering the given dataset multiple times with different initialization. The second challenge is that the number of the clusters k must be determined before conducting the clustering process [28]. Therefore, a set of experiments is conducted to evaluate the effects of different k values on the feature selection and sample classification.

7.3.3 Cluster feature extraction and representative selection

Followed by k -means clustering, we calculate the mean intensity pattern of each cluster by averaging the intensity value of m/z features within the same cluster. After obtaining the mean intensity pattern of each cluster, we choose a m/z feature with the most similar pattern to the mean pattern and a m/z feature with the most divergent pattern from the mean pattern for each cluster as the representatives of the cluster. This process can be formulated as follows:

$$r_{k_{min}} = \min_{x_i \in C_k} d(x_i, mean_k), \quad (r_{k_{min}} \in \mathcal{R}^m) \quad (7.4)$$

$$r_{k_{max}} = \max_{x_i \in C_k} d(x_i, mean_k), \quad (r_{k_{min}} \in \mathcal{R}^m) \quad (7.5)$$

where $d(\cdot)$ is the Euclidean distance defined in Equation (7.3), C_k is the k th cluster, while $r_{k_{min}}$ and $r_{k_{max}}$ are the two representatives with the most similar and the most divergent expression patterns to the mean of the k th cluster $mean_k$ which is calculated as follows:

$$mean_k = \frac{\sum_{i=1}^n I(x_i \in C_k)x_i}{S_k}, \quad (x_i, mean_k \in \mathcal{R}^m) \quad (7.6)$$

where $I(\cdot)$ is the indicator function, n is the number of m/z features, and S_k is the size of the k th cluster.

With the above extraction and selection process, our method selects two representative features per cluster, and the representative m/z features of each cluster are then combined into the clustering processed set for further selection with genetic ensemble.

7.3.4 Using genetic ensemble for m/z biomarker identification

The clustering and representative selection procedures provide us with a set of dissimilar m/z features that potentially represent different biopathway information. However, it is worth noting that not all biological pathway information in the dataset is related to the disease or the biological trait of interest. Therefore, an extra step is required to remove those unrelated representatives, which could cause negative effect on sample classification and biomarker identification if included. The genetic ensemble used for gene set selection from microarray data (Section 5.3.2) is incorporated in FCGE to further minimize the feature size by selecting those highly discriminative m/z features in a combinatorial way.

7.4 Evaluation datasets and experiment designs

This section describes the MS datasets used for algorithm evaluation, the data preprocessing details, and the evaluation methods.

7.4.1 Datasets

We use four low-resolution MS datasets for evaluation. We named each dataset by the type of disease it investigated, the protein chip type, and the mass spectrometer type if available.

7.4.1.1 OC-WCX2

This is an ovarian cancer discriminating dataset generated by study [153]. It includes 100 disease and 100 healthy samples. Unlike the dataset reported in [153] that was generated by using the H4 protein chip, this dataset was generated by using the WCX2 protein chip to cope with the discontinuation of the H4 chip. Each sample in the dataset was processed (washing, incubation, etc.) by hand and represented by 15,154 m/z features.

7.4.1.2 OC-WCX2-PBSII-a

This dataset was also generated by the WCX2 protein chip (dated as 8-7-02). Unlike the samples of OC-WCX2 dataset that were processed by hand, the samples in this dataset were processed by a robotic sample-handling instrument to explore the impact of robotic sample-handling on the spectral quality. In addition, an upgraded PBSII SELDI-TOF mass spectrometer was employed to generate the spectra. The dataset contains 91 control and 162 ovarian cancer samples, which were not randomized so that the effect of robotic automation on the spectral variance within each phenotypic group could be evaluated. Samples in the dataset are represented by 15,154 m/z features.

7.4.1.3 OC-WCX2-PBSII-b

This dataset (dated as 6-19-02) is an initial version of OC-WCX2-PBSII-a dataset. It contains the same 91 control and 162 ovarian cancer samples, and the total number of m/z features is again 15,154. However, the intensity values were normalized according to the formula:

$$NV = (V - Min)/(Max - Min) \quad (7.7)$$

where NV is the normalized value, V the raw value, Min the minimum intensity and Max the maximum intensity [151]. This equation linearly normalizes the peak intensities to the range of $[0, 1]$, and the normalization is done over all the 253 samples for all 15,154 m/z features.

7.4.1.4 PC-H4-PBS1

The last dataset was generated from the study of prostate cancer [152]. This dataset was collected using the H4 protein chip and a Ciphergen PBS1 SELDI-TOF mass spectrometer. The samples were prepared by hand and each sample is represented by 14,321 m/z features. There are a total of 322 serum samples, which are categorized into four classes. The first class contains 190 serum samples that have been diagnosed as benign prostate hyperplasia with serum prostate-specific antigen (PSA) level greater than or equal to 4 ng/ml. The second class has 63 serum samples diagnosed as no evidence of disease with serum PSA level less than 1 ng/ml. The third class contains 26 serum samples diagnosed as prostate cancer with serum PSA level between 4 and 10 ng/ml. The last 43 serum samples were categorized as the fourth class with serum PSA level greater than 10 ng/ml.

Table 7.1 summarizes each dataset used in evaluation.

Table 7.1: MS datasets used in evaluation.

Dataset	# Features	# Samples	# Class
OC-WCX2	15,154	200	2 disease: 100 healthy: 100
OC-WCX2-PBSII-a	15,154	253	2 control: 91 cancer: 162
OC-WCX2-PBSII-b	15,154	253	2 control: 91 cancer: 162
PC-H4-PBS1	14,321	322	4 no evidence: 63 benign: 190 cancer(4-10): 26 cancer(10+): 43

The study of OC-WCX2-PBSII-a and OC-WCX2-PBSII-b datasets will show us the effects of the different pre-processing and normalization procedures upon the biomarker identification and sample classification. The study of OC-WCX2 and the two OC-WCX2-PBSII datasets will demonstrate the reproducibility of the MS-based profiling, while the study of the PC-H4-PBS1 dataset will reveal the capability of the evaluated

algorithms on multi-class MS data analysis.

7.4.2 Data pre-processing

Datasets OC-WCX2, OC-WCX2-PBSII-a, and PC-H4-PBS1 are obtained from:

<http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp>

while OC-WCX2-PBSII-b is obtained from:

<http://sdmc.lit.org.sg/GEDatasets/index.html>

All datasets have been processed with baseline correction, peak detection, and peak quantification. Therefore, we used a simplified pre-processing procedure that applies the following two steps to each dataset (except OC-WCX2-PBSII-b, which is processed by Equation 7.7):

- Standardize each m/z feature to zero mean and unit variance.
- Normalize the value of each m/z feature to the range of $[0, 1]$.

After the pre-processing step, each dataset is split into selection and test sets with an external stratified 3-fold cross validation. The selection sets are then further split into training and evaluation sets with an internal stratified 3-fold cross validation for m/z feature selection. The selection sets from external cross validation are subject to prefiltering, clustering, and m/z selection, while the test sets are excluded from these processes and reserved for final m/z feature evaluation in order to provide unbiased results.

7.4.3 Results evaluation

In the m/z selection phase, the fitness of each m/z subset s is evaluated by the average score of blocking fitness (Equation 5.1) and voting fitness (Equation 5.2). The score of blocking fitness is also used as the indicator for finding optimal k of k -means clustering algorithm. In the sample classification phase, the classification accuracy of a classifier with a given m/z subset s is calculated using the balanced accuracy (Equation 5.3).

In order to compare the correlation of m/z features selected by FCGE and other alternative algorithms, we quantify the correlation of m/z features by calculating their averaged pairwise Pearson correlation coefficient:

$$avgCorr = \sum_{i=1}^{t-1} \sum_{j=i+1}^t \frac{2\sqrt{r(j,k)^2}}{t(t-1)} \quad (7.8)$$

where t is the number of m/z features from the ranking result and $r(j, k)$ is the Pearson correlation of a pair of m/z features which is computed as follows:

$$r(j, k) = \frac{\sum_i (x_{ij} - \bar{x}_{.j})(x_{ik} - \bar{x}_{.k})}{\sqrt{\sum_i (x_{ij} - \bar{x}_{.j})^2} \sqrt{\sum_i (x_{ik} - \bar{x}_{.k})^2}}; \quad (x \in \mathcal{R}^{m \times n}) \quad (7.9)$$

where i is the sample index, $\bar{x}_{.j}$ is the average value of m/z feature j across all samples, and $\bar{x}_{.k}$ is the average value of m/z feature k across all samples.

The value of average correlation varies from 0 to 1. A large value (close or equal to 1) indicates a high correlation of the selection results, while a small value (close or equal to 0) indicates a low correlation of the selection results.

7.5 Experimental results

7.5.1 Evaluating k value of k -means clustering

The k value of 50, 100, 200, 300, and 400 is tested for the k -means clustering algorithm. The size of the top ranked m/z features used in evaluation ranges from 5 to 100. The blocking accuracy of the ensemble classifier is used as the performance indicator, and the results with respect to each dataset are summarized in Figure 7.2. As can be seen, the k -means clustering algorithm with the k value of 200 and 300 seems to give the highest accuracy with the ensemble classifier. This is clarified by averaging the results of different sizes of m/z subsets according to the value of k (Figure 7.3). However, it is also realized that the change of the k value had only a limited impact on the classification results. Therefore, the k value of 200 is considered a good trade-off between the accuracy and the computation, and subsequently used in our followup feature selection and sample classification experiments.

By viewing the results of each MS dataset individually, we find that the overall blocking accuracy of the OC-WCX2 dataset is relatively steady with only a few m/z features reaching a very high classification accuracy (Figure 7.2a). The overall blocking accuracy of the OC-WCX2-PBSII-a (Figure 7.2b) and the OC-WCX2-PBSII-b (Figure 7.2c) datasets are similar in that the highest accuracy is achieved using only 10 to 20

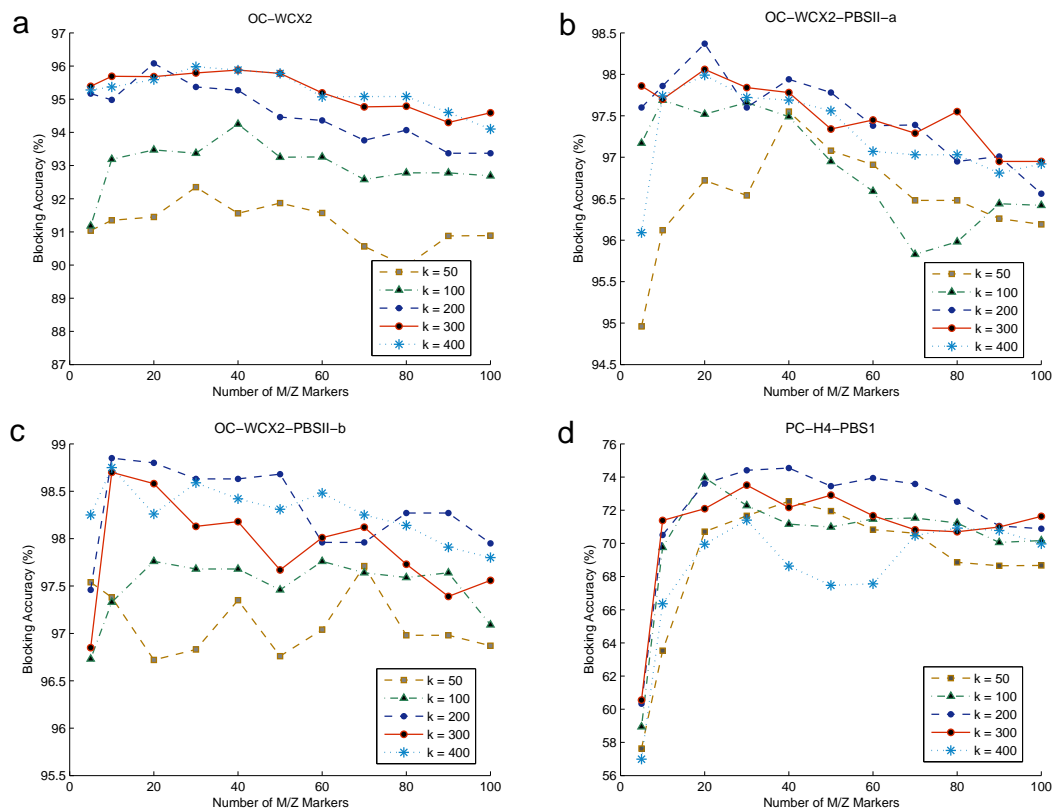


Figure 7.2: k value evaluation of FCGE hybrid system. The k value of the k -means clustering component ranging from 50 to 400 is evaluated using m/z subset with size ranging from 5 to 100.

high ranked m/z features, and both figures show a notable decline with large fluctuation when more m/z features are included. The trend of the PC-H4-PBS1 (Figure 7.2d) dataset indicates a sharp increase of blocking accuracy from subset size of 5 to size of 10, and it remains relatively stable when more m/z features are included.

A careful observation of Figure 7.2 also reveals that, in most cases, the highest fitness is achieved by using less than 40 m/z features, and the performance declines when extra m/z features are added. These results indicate that the FCGE hybrid algorithm is able to group the most differential m/z features into a relatively small and compact feature subset for sample classification.

7.5.2 Sample classification

The sample classification accuracy of the proposed FCGE hybrid system is compared with those achieved by using univariate Information Gain [69], ReliefF [157], BWSS

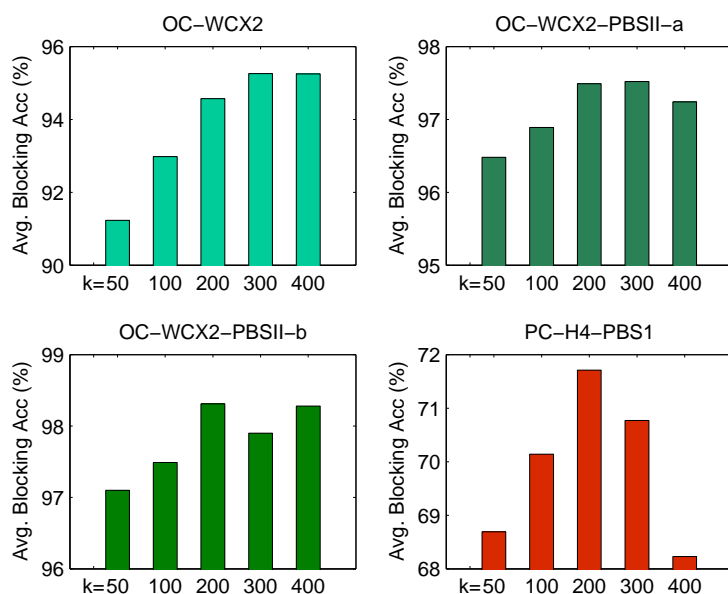


Figure 7.3: Average blocking accuracy according to different k values.

[56], and the GA/ k NN algorithm [116], all of which have been applied to m/z feature selection and classification of MS datasets previously.

Since the highest classification accuracy of all four MS datasets can be achieved within the size of 40 top ranked m/z features, we compare different feature selection algorithms using the m/z subsets with size of 5, 10, 20, 30, and 40, respectively. Ten different supervised classification algorithms are used for classification accuracy comparison. They are trained using the top ranked m/z features obtained by each selection algorithm. These 10 classification algorithms are *decision tree* (J4.8), *1-nearest neighbour* (1-NN), *3-nearest neighbour* (3-NN), *7-nearest neighbour* (7-NN), *naive bayes* (NB), *support vector machine* (SVM), *multi-layer perceptron* (MLP), *random forests* (RF), *multinomial logistic regression* (Logistic), and *radial basis function network* (RBFnet). The default parameters of Weka for each classification algorithm are used [78]. The purpose of using such a wide range of classifiers is to obtain an unbiased and general evaluation of the m/z feature selection algorithms that play the role of identifying informative m/z biomarkers that help the classification algorithm to achieve high classification accuracy.

The detailed classification results (shown as classification error rates) of the 10 classifiers by using the m/z features ranked by FCGE with BWSS (FCGE(BWSS)), FCGE

Table 7.2: OC-WCX2 dataset. Error rate comparison of six different m/z feature selection algorithms using 10 different classifiers with size of the top ranked m/z features from 5 to 40

Classifier	FCGE(BWSS)						FCGE(χ^2)					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	10.02	9.09	9.01	8.50	9.01	9.13	9.09	9.09	10.02	10.02	10.02	9.65
1-NN	6.99	2.53	2.02	2.02	2.02	3.12	2.53	3.54	2.53	4.04	2.02	2.93
3-NN	4.02	3.03	3.03	2.53	3.03	3.13	3.54	3.03	2.02	3.03	3.54	3.03
7-NN	4.97	4.51	3.54	3.54	3.54	4.02	4.97	4.97	2.02	3.54	3.54	3.81
NB	5.07	4.57	2.53	2.48	2.53	3.44	4.04	3.03	3.03	2.53	4.55	3.44
SVM	4.51	3.03	2.53	2.53	2.53	3.03 [†]	4.50	3.03	2.02	2.53	2.53	2.92 [†]
MLP	4.49	1.52	4.04	2.53	4.04	3.32	3.54	3.54	2.53	2.53	2.53	2.93
RF	5.49	6.02	5.05	3.03	5.05	4.93	5.05	5.01	4.55	3.54	6.48	4.93
Logistic	4.50	5.05	2.53	3.03	3.03	3.63	3.03	3.99	2.53	3.03	2.53	3.02
RBFnet	3.98	4.05	2.53	2.53	2.53	3.12	3.03	4.55	2.02	3.03	5.56	3.64
S avg	5.40	4.34	3.68	3.27*	3.73	4.09	4.33	4.38	3.32*	3.78	4.33	4.03
Classifier	BWSS						GA/kNN					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	9.01	10.02	10.52	10.52	10.52	10.12	10.02	10.02	10.52	10.52	10.52	10.32
1-NN	4.04	5.01	4.04	4.04	3.54	4.13	5.05	4.04	4.04	3.54	4.04	4.14
3-NN	3.54	3.54	3.03	3.03	3.54	3.37	3.54	2.53	3.03	3.03	3.54	3.13
7-NN	4.50	3.54	3.54	4.04	4.04	3.93	5.51	3.03	3.54	3.03	4.04	3.83
NB	4.00	4.04	3.03	3.03	3.03	3.43	5.01	3.54	3.03	2.53	3.49	3.52
SVM	4.00	3.03	2.53	3.03	2.53	3.02 [†]	3.49	2.53	2.53	2.53	2.53	2.72 [†]
MLP	4.04	6.06	3.03	3.54	2.53	3.84	4.55	4.55	5.05	3.03	2.53	3.94
RF	6.02	6.48	5.51	6.57	6.02	6.12	6.02	3.53	5.05	5.51	5.93	5.21
Logistic	8.59	8.08	7.53	7.53	4.00	7.15	7.58	5.51	5.56	4.04	3.03	5.14
RBFnet	3.49	3.03	4.00	6.02	5.51	4.41	4.00	3.03	4.04	4.00	6.02	4.22
S avg	5.12	5.28	4.68	5.13	4.52*	4.95	5.48	4.23	4.64	4.18*	4.57	4.62
Classifier	Information Gain						ReliefF					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	9.01	9.01	10.52	10.52	10.52	9.92	8.00	10.02	10.52	10.52	10.52	9.92
1-NN	4.04	4.04	5.56	5.56	3.54	4.55	6.57	5.51	4.50	3.54	3.03	4.63
3-NN	3.54	2.53	3.54	3.03	3.03	3.13	5.05	4.00	3.54	4.04	3.54	4.03
7-NN	3.99	3.54	3.54	3.54	5.05	3.93	4.04	4.50	4.55	4.04	5.05	4.44
NB	3.03	3.54	4.50	4.04	4.50	3.92	3.54	2.99	2.99	4.00	4.50	3.60
SVM	3.03	2.53	2.53	2.53	2.53	2.63 [†]	3.54	3.03	3.03	3.03	3.54	3.23 [†]
MLP	3.54	4.55	3.03	4.04	2.53	3.54	4.04	5.01	4.04	4.04	4.04	4.23
RF	7.53	5.05	7.03	7.03	6.52	6.63	7.03	6.02	7.53	5.56	9.05	7.04
Logistic	5.56	6.99	6.52	7.07	5.01	6.23	9.09	11.57	5.56	6.02	5.01	7.45
RBFnet	4.04	2.53	3.03	4.50	5.51	3.92	4.04	4.50	6.52	5.01	7.03	5.42
S avg	4.73	4.43*	4.98	5.19	4.87	4.84	5.49	5.72	5.28	4.98*	5.53	5.40

[†] classifier with the lowest classification error rate across different m/z subset sizes.

* m/z subset size with the lowest classification error rate across all classification algorithms.

with χ^2 (FCGE(χ^2)), GA/kNN, BWSS, Information Gain, and ReliefF are presented in Tables 7.2-7.5. The column of “C avg” shows the average error rates with a given classifier using different m/z feature sizes, while the row of “S avg” shows the average error rates with a given size of m/z set across different classifiers. The first value gives an average indication of a specific classifier’s power on sample classification while the second value gives an average indication of the effect of the m/z subset size on MS data classification. The grand mean error rates across all m/z feature sizes and all classifiers are marked in bold. As can be seen, the proposed FCGE hybrid algorithm is able to

Table 7.3: OC-WCX2-PBSII-a dataset. Error rate comparison of four different m/z feature selection algorithms using 10 different classifiers with size of the top ranked m/z features from 5 to 40

Classifier	FCGE(BWSS)						FCGE(χ^2)					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	4.09	4.39	4.39	3.53	3.53	3.99	2.96	5.06	5.06	5.57	5.57	4.84
1-NN	1.73	1.67	0.56	1.11	0.56	1.13	2.22	1.67	0.00	1.67	1.67	1.45
3-NN	0.86	1.67	0.56	1.11	1.11	1.06	2.22	1.67	0.56	2.22	1.11	1.56
7-NN	1.67	1.67	0.56	1.67	1.67	1.45	2.78	1.67	1.67	1.11	1.11	1.67
NB	0.86	0.56	0.86	1.42	1.67	1.11	1.79	0.62	0.86	1.42	0.86	1.11
SVM	1.67	1.11	0.00	0.00	1.11	0.78 [†]	1.11	1.11	0.00	0.00	0.00	0.44 [†]
MLP	1.73	1.42	0.56	1.11	0.56	1.08	1.11	0.56	0.00	0.56	0.56	0.56
RF	4.02	3.47	2.91	3.71	3.15	3.45	3.65	3.47	4.01	2.59	3.15	3.37
Logistic	2.35	0.00	0.86	0.56	0.56	0.87	0.00	0.31	0.31	1.17	1.17	0.59
RBFnet	2.23	2.04	0.31	0.86	1.48	1.38	1.48	1.42	1.11	1.67	1.11	1.36
S avg	2.12	1.80	1.16*	1.51	1.54	1.63	1.93	1.76	1.36*	1.80	1.63	1.69
Classifier	BWSS						GA/kNN					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	4.83	6.37	5.01	5.01	5.01	5.25	4.83	5.01	5.01	5.01	5.01	4.97
1-NN	5.13	4.58	2.59	3.46	2.28	3.61	3.77	4.07	0.56	0.56	2.22	2.24
3-NN	3.34	3.15	1.98	2.28	2.53	2.66	3.34	2.59	0.56	1.11	1.67	1.85
7-NN	3.34	3.70	1.98	1.98	2.22	2.64	2.53	2.90	1.67	2.22	2.22	2.31
NB	4.02	4.27	2.28	2.28	1.98	2.97	2.90	3.52	2.35	2.28	2.28	2.67
SVM	3.34	2.53	1.73	2.59	0.56	2.15 [†]	2.53	2.35	0.56	0.00	0.56	1.20 [†]
MLP	4.83	4.63	2.04	2.28	1.42	3.04	3.46	2.35	2.35	1.11	1.11	2.08
RF	4.52	4.27	3.96	3.47	3.71	3.99	5.99	2.85	3.47	2.54	1.48	3.27
Logistic	6.12	5.62	3.77	3.52	1.42	4.09	4.75	4.75	1.85	0.62	0.62	2.52
RBFnet	4.33	5.44	3.90	3.04	2.48	3.84	4.02	4.27	2.53	1.42	1.42	2.73
S avg	4.38	4.46	2.92	2.99	2.36*	3.42	3.81	3.47	2.09	1.69*	1.86	2.58
Classifier	Information Gain						ReliefF					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	5.44	5.01	5.01	5.01	5.01	5.10	4.83	6.37	5.01	5.01	5.01	5.25
1-NN	7.17	4.58	3.52	2.91	2.84	4.20	5.13	6.31	3.10	4.01	3.28	4.37
3-NN	4.21	3.65	2.91	2.79	2.53	3.22	3.34	4.46	3.34	2.84	3.40	3.48
7-NN	4.21	3.65	2.59	1.98	2.53	2.99	3.34	3.90	2.53	2.53	2.78	3.02
NB	4.02	3.71	2.91	2.91	1.98	3.11	4.02	4.27	2.28	2.59	2.28	3.09
SVM	4.46	2.59	2.91	2.59	2.28	2.97 [†]	3.34	3.34	1.98	1.98	1.67	2.46 [†]
MLP	5.69	3.21	2.91	2.28	1.98	3.21	4.83	4.58	2.59	1.98	1.42	3.08
RF	6.06	3.71	4.09	4.02	4.95	4.57	4.52	4.58	3.41	4.27	3.71	4.10
Logistic	5.75	4.58	3.52	2.91	2.28	3.81	6.12	7.48	4.38	3.15	2.65	4.76
RBFnet	3.71	2.79	3.96	3.34	2.79	3.32	4.33	4.89	3.90	3.70	4.01	4.17
S avg	5.07	3.75	3.43	3.07	2.92*	3.65	4.38	5.02	3.25	3.21	3.02*	3.78

achieve the lowest grand mean error rates (which is the highest classification accuracy) in all four MS datasets. Specifically, grand mean error rates of FCGE(BWSS) and FCGE(χ^2) in OC-WCX2, OC-WCX2-PBSII-a, and OC-WCX2-PBSII-b datasets classification are 4.09, 1.63, 1.10, and 4.03, 1.69, 1.34, respectively, which are consistently better than those obtained by GA/kNN, BWSS, Information Gain, and ReliefF. As for the PC-H4-PBS1 dataset, the improvement is about 3% to GA/kNN, 5-6% to BWSS and ReliefF algorithms, and a significant 16% over Information Gain.

It is also clear that the classification results of FCGE(BWSS) and FCGE(χ^2) are very similar. The results indicate that the effect of different filter algorithms is similar

Table 7.4: OC-WCX2-PBSII-b dataset. Error rate comparison of six different m/z feature selection algorithms using 10 different classifiers with size of the top ranked m/z features from 5 to 40

Classifier	FCGE(BWSS)						FCGE(χ^2)					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	4.27	4.27	3.46	3.46	3.46	3.78	5.86	3.46	3.46	3.45	3.46	3.94
1-NN	1.68	0.56	0.00	0.56	0.56	0.67	1.68	0.56	0.56	1.11	1.11	1.00
3-NN	1.42	1.11	0.56	0.00	0.00	0.62	1.68	0.00	0.56	0.56	0.56	0.67
7-NN	1.11	1.11	0.56	0.00	0.56	0.67	1.68	1.11	0.56	0.56	0.56	0.89
NB	2.65	0.56	0.31	0.31	0.31	0.83	1.79	0.62	0.86	1.17	1.17	1.12
SVM	1.67	0.56	0.00	0.00	0.00	0.45	1.68	0.00	0.56	0.56	0.56	0.67
MLP	1.42	0.56	0.00	0.00	0.00	0.40 [†]	0.62	0.00	0.56	0.56	0.56	0.46 [†]
RF	4.48	1.11	3.70	1.98	1.62	2.58	3.77	2.53	2.53	2.22	2.28	2.67
Logistic	2.04	0.00	0.31	0.00	0.00	0.47	1.23	0.31	0.31	0.86	1.17	0.78
RBFnet	1.17	0.00	0.31	0.31	0.62	0.48	1.17	0.56	1.11	1.67	1.73	1.25
S avg	2.19	0.98	0.92	0.66*	0.71	1.10	2.12	0.91*	1.10	1.27	1.31	1.34
Classifier	BWSS						GA/kNN					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	3.41	3.41	4.14	4.14	4.14	3.85	3.10	3.83	3.83	3.83	3.83	3.68
1-NN	4.27	2.68	1.11	2.17	0.56	2.16	3.29	2.79	1.11	1.11	1.11	1.88
3-NN	3.80	2.68	1.11	1.11	1.11	1.96	2.79	1.73	1.11	1.11	1.11	1.57
7-NN	3.54	2.99	1.11	1.62	2.73	2.40	2.79	2.23	1.11	1.11	1.67	1.78
NB	3.71	3.71	1.42	1.42	1.42	2.34	3.41	2.35	1.48	1.42	1.98	2.13
SVM	4.05	2.99	1.11	1.11	0.56	1.96	3.10	1.73	1.11	1.11	0.56	1.52
MLP	3.54	1.73	0.86	0.86	0.56	1.51 [†]	2.68	0.86	0.00	0.56	0.56	0.93 [†]
RF	3.49	2.99	4.15	4.44	2.22	3.46	3.74	4.21	4.20	4.52	1.98	3.73
Logistic	4.47	3.77	2.35	2.04	1.17	2.76	6.11	2.35	1.23	0.62	0.31	2.12
RBFnet	3.10	3.21	1.42	1.98	1.98	2.34	3.10	1.73	0.86	1.67	1.11	1.69
S avg	3.74	2.99	1.88	2.09	1.64*	2.47	3.41	2.38	1.60	1.70	1.48*	2.11
Classifier	Information Gain						ReliefF					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	3.10	2.59	3.83	3.83	3.83	3.44	3.15	3.15	4.38	4.38	4.38	3.89
1-NN	3.10	2.54	1.73	1.11	1.67	2.03	4.27	1.98	1.67	1.67	1.67	2.25
3-NN	5.03	2.23	2.23	1.11	1.11	2.34	3.80	1.42	3.79	2.73	2.72	2.89
7-NN	4.16	3.85	2.74	2.12	2.17	3.01	3.54	2.48	3.79	2.73	2.72	3.05
NB	4.64	3.16	1.73	1.98	1.98	2.70	3.71	2.59	3.46	2.28	2.53	2.91
SVM	4.91	2.79	1.73	1.11	1.11	2.33	3.74	1.92	1.11	0.56	0.56	1.58
MLP	2.68	1.42	1.42	0.86	0.56	1.39 [†]	3.54	0.56	1.11	1.11	1.11	1.49 [†]
RF	3.71	3.15	4.20	5.01	3.33	3.88	3.49	2.48	4.75	3.04	3.09	3.37
Logistic	5.95	2.90	2.35	2.04	2.04	3.06	4.47	2.10	0.86	0.86	1.17	1.89
RBFnet	3.41	3.21	2.23	1.42	2.28	2.51	3.10	1.42	1.67	2.53	2.53	2.25
S avg	4.07	2.78	2.42	2.06	2.01*	2.67	3.68	2.01*	2.66	2.19	2.25	2.56

for the purpose of prefiltering, and obtaining a size of one fifth of m/z features in prefiltering is large enough to preserve most useful m/z features for followup classification processing.

We marked the lowest C value for finding the best classifier and the lowest S value for finding the best m/z feature size for each MS dataset, respectively. One interesting finding is that an association seems to exist between the type of the classifier and the dataset. In OC-WCX2 dataset and OC-WCX2-PBSII-a dataset classification, SVM classifier is identified as the best classifier consistently with all six m/z feature selection algorithms, while MLP is identified as the best classifier consistently for the OC-WCX2-PBSII-b dataset. As for the PC-H4-PBS1 dataset, the best classifier is 1-NN

Table 7.5: PC-H4-PBS1 dataset. Error rate comparison of six different m/z feature selection algorithms using 10 different classifiers with size of the top ranked m/z features from 5 to 40

Classifier	FCGE(BWSS)						FCGE(χ^2)					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	31.49	33.94	35.17	33.71	33.77	33.62	39.57	32.90	29.41	31.42	35.23	33.71
1-NN	30.43	27.75	19.29	16.28	16.26	22.00 [†]	39.58	25.59	22.03	20.19	18.27	25.13 [†]
3-NN	29.49	26.08	23.65	19.52	21.67	24.08	38.81	27.34	23.21	24.39	23.30	27.41
7-NN	36.03	31.01	27.94	31.51	31.75	31.65	42.73	33.15	30.42	30.73	29.69	33.34
NB	28.69	27.85	25.18	24.15	23.97	25.97	37.66	28.44	26.87	21.21	20.81	26.99
SVM	44.82	43.09	24.11	18.67	20.50	30.24	42.69	43.57	25.44	20.44	21.12	30.65
MLP	29.27	25.11	20.08	27.92	21.98	24.87	37.09	31.61	24.85	19.75	20.66	26.79
RF	28.98	28.45	29.80	29.40	30.57	29.44	39.84	28.57	34.05	36.81	29.95	33.84
Logistic	35.54	34.56	32.95	34.89	33.69	34.33	37.91	32.21	35.61	31.09	28.26	33.02
RBFnet	40.15	37.24	26.69	28.27	35.41	33.55	39.14	35.01	25.48	29.39	25.65	30.93
S avg	33.49	31.51	26.48	26.43*	26.96	28.97	39.50	31.84	27.74	26.54	25.29*	30.18
Classifier	BWSS						GA/kNN					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	33.68	33.18	35.19	35.09	35.22	34.47	36.42	41.72	35.22	35.90	33.72	36.60
1-NN	35.11	32.26	34.46	34.07	30.29	33.24	36.36	23.15	17.43	17.35	16.92	22.24 [†]
3-NN	33.59	29.93	34.42	35.11	37.59	34.13	37.74	33.78	26.61	24.36	21.74	28.85
7-NN	38.74	35.73	40.60	39.73	42.19	39.40	42.82	45.60	37.51	32.03	32.19	38.03
NB	33.02	35.95	37.06	33.79	34.71	34.91	41.74	37.74	33.08	30.81	25.17	33.71
SVM	55.76	45.27	42.86	35.60	30.28	41.95	55.25	41.43	31.88	25.42	22.45	35.29
MLP	31.57	26.94	34.57	29.06	27.10	29.85 [†]	35.63	29.38	27.23	25.81	26.18	28.85
RF	33.98	32.19	32.53	35.83	34.78	33.86	42.98	33.80	31.30	30.99	34.44	34.70
Logistic	34.20	34.06	37.21	30.43	32.48	33.68	42.22	40.51	30.65	35.70	31.15	36.05
RBFnet	33.72	39.74	38.03	37.77	41.58	38.17	45.29	39.90	32.98	31.37	28.15	35.54
S avg	36.34	34.53*	36.69	34.65	34.62	35.37	41.64	36.70	30.39	28.97	27.21*	32.98
Classifier	Information Gain						ReliefF					
	5	10	20	30	40	C avg	5	10	20	30	40	C avg
J4.8	48.22	44.88	44.70	42.39	42.16	44.47	30.23	35.06	34.12	33.01	32.74	33.03
1-NN	50.04	42.54	46.27	46.91	44.48	46.05	34.89	24.64	25.96	29.11	28.75	28.67
3-NN	50.16	41.68	46.07	44.68	45.21	45.56	35.51	32.32	38.45	37.51	35.29	35.82
7-NN	49.22	43.39	48.81	49.76	45.76	47.39	34.95	41.28	43.34	42.09	41.67	40.67
NB	45.53	45.02	44.84	45.51	46.25	45.43	33.59	35.25	37.34	39.80	39.54	37.10
SVM	56.75	56.06	53.94	49.74	47.79	52.86	48.51	41.32	38.07	38.60	38.34	40.97
MLP	48.49	39.37	44.47	43.18	37.17	42.54 [†]	32.01	30.51	26.67	25.87	23.44	27.70 [†]
RF	50.03	46.12	43.43	44.65	46.68	46.18	40.39	41.18	36.83	39.56	33.63	38.32
Logistic	48.19	44.85	51.07	41.87	41.66	45.53	34.37	41.99	36.20	32.38	31.50	35.29
RBFnet	47.89	46.94	47.11	46.93	46.07	46.99	37.75	42.15	39.15	37.14	40.77	39.39
S avg	49.45	45.09	47.07	45.56	44.32*	46.30	36.22	36.57	35.61	35.51	34.57*	35.70

when using FCGE(BWSS), FCGE(χ^2) and GA/kNN, while the classifier of MLP is the most successful when using BWSS, Information Gain, and ReliefF algorithms. Since the number of datasets is limited, it is hard to interpret whether there is a classifier-dataset specific relationship. Nonetheless, it is arguable that SVM and MLP are the most competitive classifiers for MS data classification. For FCGE hybrid algorithm, the lowest error rates are achieved in all three ovarian cancer datasets using only 10 to 30 top ranked m/z features. This indicates that the FCGE hybrid algorithm is capable of selecting the most important m/z features that can effectively represent the underlying patterns.

Lastly, we applied a pairwise t -test to calculate p -values for BWSS, GA/ k NN, Information Gain, and ReliefF against FCGE(BWSS) and FCGE(χ^2), respectively. Suppose the error rates given by a feature selection algorithm F^i using classifiers $\langle L_1^i \dots L_n^i \rangle$ are $\langle e_1^i \dots e_n^i \rangle$. Then, the difference between two feature selection algorithms with respect to sample classification can be represented as $Diff = \langle e_1^i - e_1^j \dots e_n^i - e_n^j \rangle$. Given the null hypothesis $H_0 : Diff = 0$ and the alternative hypothesis $H_1 : Diff > 0$, we can evaluate whether the error rates given by a feature selection algorithm F^i are significantly higher than F^j . Table 7.6 shows the p -values for each pairwise test. It is clear that in most cases the error rates given by FCGE(BWSS) and FCGE(χ^2) are significantly lower than those given by alternative methods ($p < 0.05$).

Table 7.6: Significance test of error rate for feature selection algorithms in terms of sample classification using each MS dataset, respectively. The calculations are performed using 5-40 selected m/z features, respectively. Each number is a p -value calculated using a pairwise one-tail Student t -test to 3 decimal places.

OC-WCX2	5	10	20	30	40
BWSS vs FCGE(BWSS); FCGE(χ^2)	0.686; 0.097	0.070; 0.062	0.046; 0.007	0.001; 0.011	0.029; 0.251
GA/kNN vs FCGE(BWSS); FCGE(χ^2)	0.431; 0.018	0.582; 0.644	0.008; 0.001	0.007; 0.069	0.039; 0.189
Information Gain vs FCGE(BWSS); FCGE(χ^2)	0.911; 0.200	0.427; 0.453	0.015; 0.000	0.001; 0.006	0.010; 0.053
ReliefF vs FCGE(BWSS); FCGE(χ^2)	0.442; 0.077	0.047; 0.050	0.002; 0.000	0.000; 0.002	0.001; 0.001
OC-WCX2-PBSII-a	5	10	20	30	40
BWSS vs FCGE(BWSS); FCGE(χ^2)	0.000; 0.000	0.000; 0.000	0.000; 0.001	0.000; 0.001	0.001; 0.001
GA/kNN vs FCGE(BWSS); FCGE(χ^2)	0.000; 0.000	0.003; 0.002	0.001; 0.016	0.231; 0.671	0.160; 0.224
Information Gain vs FCGE(BWSS); FCGE(χ^2)	0.000; 0.000	0.000; 0.000	0.000; 0.000	0.000; 0.000	0.000; 0.000
ReliefF vs FCGE(BWSS); FCGE(χ^2)	0.000; 0.000	0.000; 0.000	0.001; 0.000	0.000; 0.000	0.000; 0.000
OC-WCX2-PBSII-b	5	10	20	30	40
BWSS vs FCGE(BWSS); FCGE(χ^2)	0.003; 0.007	0.000; 0.000	0.000; 0.001	0.000; 0.000	0.000; 0.094
GA/kNN vs FCGE(BWSS); FCGE(χ^2)	0.012; 0.029	0.001; 0.000	0.000; 0.014	0.000; 0.000	0.000; 0.300
Information Gain vs FCGE(BWSS); FCGE(χ^2)	0.003; 0.007	0.001; 0.000	0.000; 0.000	0.000; 0.009	0.000; 0.000
ReliefF vs FCGE(BWSS); FCGE(χ^2)	0.004; 0.010	0.005; 0.000	0.000; 0.001	0.000; 0.001	0.000; 0.001
PC-H4-PBS1	5	10	20	30	40
BWSS vs FCGE(BWSS); FCGE(χ^2)	0.037; 0.942	0.002; 0.017	0.000; 0.003	0.000; 0.000	0.000; 0.000
GA/kNN vs FCGE(BWSS); FCGE(χ^2)	0.000; 0.101	0.007; 0.009	0.010; 0.009	0.061; 0.000	0.406; 0.023
Information Gain vs FCGE(BWSS); FCGE(χ^2)	0.000; 0.000	0.000; 0.000	0.000; 0.000	0.000; 0.000	0.000; 0.000
ReliefF vs FCGE(BWSS); FCGE(χ^2)	0.037; 0.981	0.005; 0.000	0.000; 0.003	0.000; 0.000	0.004; 0.001

7.5.3 Correlation reduction

In our previous work, the k -means clustering component was employed in the hope that the correlation of the selected m/z features would be reduced for redundancy control. However, no measure has been proposed to assess the level of correlations of the selected m/z features. In order to compare the correlation level of the top ranked m/z features with each m/z ranking algorithm, in this study, we quantify the correlation among m/z features by calculating the Pearson correlation coefficient in a pairwise manner using each selection algorithm. The ranking size of the m/z features, again, ranges from 5 to 40, and the correlation values of each pair of m/z features are averaged for comparison using Equation 7.8. This value ranges from 0 to 1 with the low value indicating low overall correlation and the high value indicating high overall correlation. The results grouped by selection algorithms and m/z feature size are presented in Table 7.7. Figure 7.4 is the visualization of the results.

It is easily observed that the proposed FCGE system is able to reduce the overall correlation among the selected m/z features considerably. In three ovarian cancer datasets classification, essentially, the correlation decreases with the increase of the m/z feature size. As for the prostate dataset, no significant changes of correlation with respect to different m/z feature sizes are observed.

Table 7.7: Correlation evaluation details. Pearson correlation of the m/z feature selection results are calculated in a pairwise manner and grouped by the type of selection algorithm and the feature size.

	OC-WCX2					OC-WCX2-PBSII-a				
	5	10	20	30	40	5	10	20	30	40
FCGE (BWSS)	0.235	0.295	0.283	0.279	0.243	0.532	0.517	0.464	0.408	0.363
FCGE (χ^2)	0.404	0.357	0.274	0.254	0.235	0.462	0.435	0.377	0.363	0.349
GA/kNN	0.659	0.598	0.510	0.461	0.438	0.906	0.811	0.700	0.635	0.577
BWSS	0.688	0.609	0.580	0.549	0.522	0.968	0.903	0.777	0.742	0.684
ReliefF	0.582	0.543	0.518	0.512	0.509	0.973	0.888	0.765	0.698	0.658
InfoGain	0.612	0.604	0.581	0.546	0.518	0.964	0.852	0.776	0.730	0.683
	OC-WCX2-PBSII-b					PC-H4-PBSI				
	5	10	20	30	40	5	10	20	30	40
FCGE (BWSS)	0.451	0.473	0.438	0.394	0.357	0.163	0.241	0.244	0.245	0.219
FCGE (χ^2)	0.540	0.443	0.401	0.362	0.347	0.184	0.229	0.221	0.234	0.243
GA/kNN	0.905	0.825	0.698	0.638	0.556	0.349	0.292	0.295	0.285	0.273
BWSS	0.948	0.911	0.754	0.752	0.667	0.379	0.423	0.416	0.402	0.422
ReliefF	0.973	0.852	0.778	0.695	0.658	0.276	0.420	0.450	0.427	0.441
InfoGain	0.957	0.876	0.782	0.729	0.687	0.781	0.793	0.778	0.776	0.760

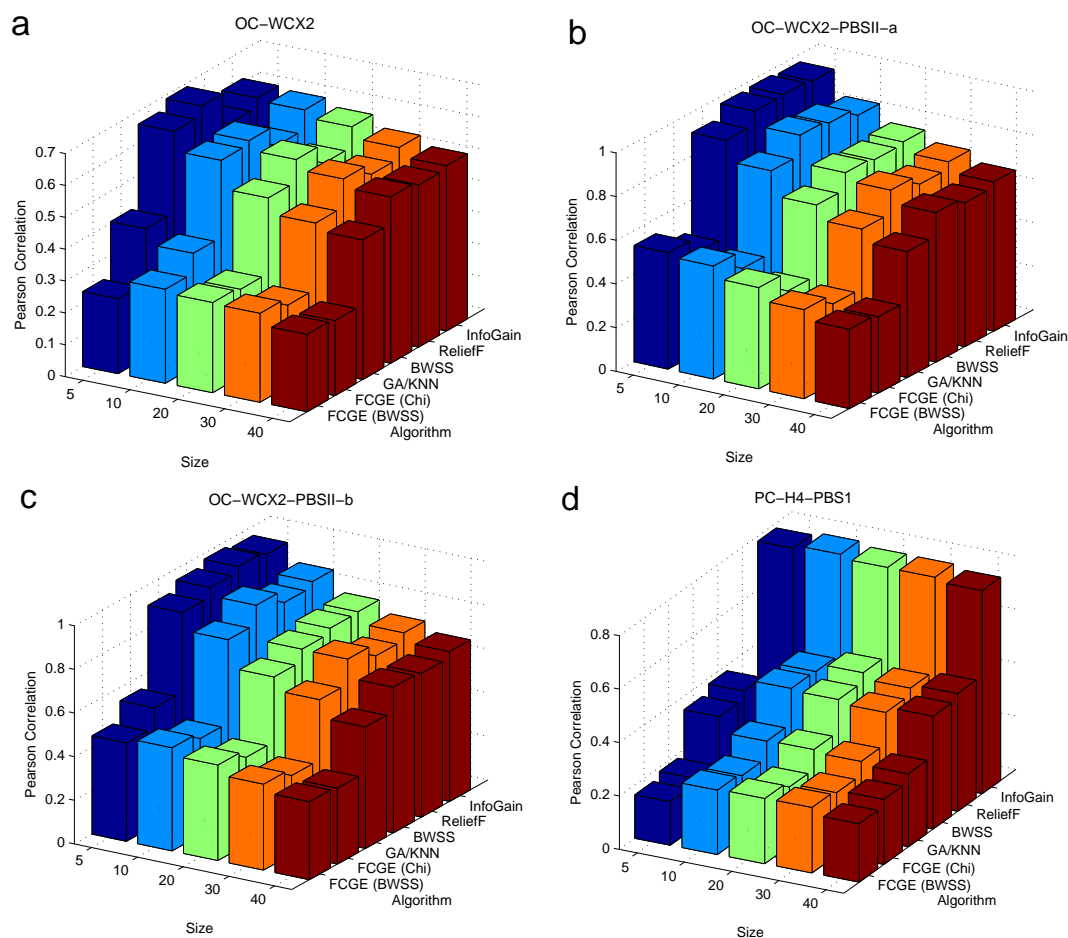


Figure 7.4: Correlation evaluation. Different correlation levels are plotted by m/z subset size and selection algorithms in a pairwise manner.

Comparing the correlation of the selection results and the error rate in sample classification according to m/z feature ranking algorithms, we find that there is a positive association in that the increase of the correlation goes with the increase of the error rate. That is, the ranking algorithm that generates lower correlation results can often achieve higher sample classification accuracy. Although most datasets reveal a decrease of the correlation with the increase of the m/z feature size, as observed in Tables 7.2-7.5 the increase of the m/z feature size does not always bring higher classification accuracy. This implies that the decrease of the correlation does not always accompany the increase of the classification accuracy within a given m/z feature ranking algorithm. This is natural because not all m/z features are informative/useful in sample classification. Although, including those low correlated m/z features will decrease the overall correlation of the selection results, they will not bring any gain in sample classification because they lack

relevance to the trait we are investigating.

7.6 Discussion and summary

In PC-H4-PBS1 dataset analysis, we see that that Information Gain algorithm performs much worse than the other ranking algorithms. This result suggests a lack of power for univariate methods in multi-classes data classification. The BWSS algorithm and the multivariate ReliefF algorithm are able to improve the result by 11%, and the wrapper-based GA/*k*NN algorithm gives an improvement of nearly 13%. However, our FCGE algorithm provides an even better result with another 3-4% percent improvement over GA/*k*NN. This could be attributed to the use of *k*-means clustering for correlation reduction and the use of ensemble of classifiers for *m/z* feature selection and ranking.

The advantage of using ensemble classifier for feature selection can be justified by the assumption that different classification models tend to misclassify a different portion of samples if the proper integration strategies are employed. These model combination and model averaging strategies have long been known in the machine learning community [53,199], and determining which attributes should be used as the input is important for improvement of overall classification accuracy [145]. While genetic ensemble hybrid aims to select useful *m/z* features to improve the overall classification accuracy, it provides a natural way to identify biologically significant *m/z* biomarkers. As a consequence, since the *m/z* feature selection and evaluation are accomplished by using multiple classifiers, they are less subject to the bias of certain inductive algorithms but more likely to reflect the genuine association with the disease of interest.

By viewing the sample classification results and the correlation reduction results, we have the following conclusion: correlation reduction may be the key to promote sample classification and the identification of disease associated biomarkers from the bio-pathway level, but evidently not all pathway information is associated with the disease of interest. Therefore, algorithms that use all representatives the that together minimize the correlation to the minimum may not only include redundancy but could also exert negative effects on the selection and classification results. The FCGE hybrid system provides a framework to incorporate correlation minimization as an intermediate step, which circumvents the disadvantage of relying solely on the correlation reduction by using it as an information enrichment and pattern enhancement measure.

In summary, we proposed a k -means clustering-based feature extraction and selection approach for the analysis of MS dataset. This hybrid system incorporated filter-based prefiltering, k -means clustering-based correlation reduction and representative selection, and genetic ensemble-based wrapper selection procedures. The k -means clustering process serves as the bridge between filter-based pre-selection and wrapper-based feature selection processes. It helps to decrease the dimensionality of the pre-filtered dataset while also reducing the correlation of the selected m/z features, outputting a noise-reduced and information-enriched dataset. The experimental results suggest that the proposed FCGE system has good capability in sample classification and m/z biomarker identification for MS dataset analysis.

Chapter 8

Conclusions and Future Work

In this chapter, we summarize the thesis and propose potential future research directions that could be extended from this thesis.

8.1 Conclusions of the thesis

Computational and systems biology is a fast growing research field, driven by the continuous development in both high-throughput technologies and computational methods. This thesis focuses in particular on ensemble learning methods and hybrid algorithms and their application to some of the most representative research topics in computational and systems biology.

- In Chapter 3, we studied the reproducibility and success rate in functional SNP and SNP pair filtering from GWA studies using both simulation and real-world datasets. We demonstrated that some of the most popular SNP filtering algorithms such as ReliefF and TuRF are sensitive to the order of the samples in the dataset, causing a significant change of SNP rankings when the order of the samples are changed. Such an undesirable artifact originated from the k -nearest neighbour algorithm employed by ReliefF and TuRF for choosing learning examples. By harnessing this artificial effect as the diversity of the ensemble learning models, we proposed an ensemble of filters that is capable of overcoming the low reproducibility of the original filter algorithms while also improving the success rate on functional SNP and SNP interaction pair filtering.

- In Chapter 4, we developed a genetic ensemble (GE) algorithm for identifying gene-gene interaction and gene-environmental factor interaction. The GE algorithm incorporates multiple classification algorithms in a genetic framework using three integration functions, namely blocking, majority voting, and double fault diversity. Blocking and majority voting were used to combine the prediction from multiple classifiers, whereas double fault diversity was used to promote diversities among these classifiers. We showed that the GE algorithm has much higher power in terms of identifying interactions compared to any single classifiers. Furthermore, we proposed a novel function for evaluating the degree of complementarity of results generated by different gene-gene interaction identification algorithms. We demonstrated that the proposed GE algorithm gives complementary results to other algorithms such as MDR and PIA whereas the results from MDR and PIA are very similar to each other. Therefore, the proposed GE algorithm provides a unique means to identify many more gene-gene interactions when used together with other identification algorithms.
- We moved to the transcriptomic level in Chapter 5 where the focus is on selecting gene sets from microarray-based expression profiling for disease and normal sample classification. In this chapter, we introduced a score mapping method for combining multiple filter algorithms with the GE algorithm. The system, called “MF-GE”, is able to fuse the pre-filtering scores of each gene computed by each filter algorithm to the initialization and mutation operations of the genetic algorithms. MF-GE is therefore fast in terms of convergence and is able to identify smaller gene subsets that give higher prediction power. This indicates that MF-GE is capable of selecting the most discriminative genes while also reducing the redundant ones. The selected gene subsets may contain key biomarkers for disease diagnosis and prevention, and are potential candidates for followup validation.
- Chapter 6 focused on post-processing spectrum-peptide matches (PSMs) generated from MS-based proteomics studies. In this chapter, we showed that a semi-supervised learning algorithm called “Percolator” is sensitive to the initial PSM ranking in PSM filtering. It performs suboptimally when the initial PSMs are ranked poorly. We extended Percolator for X!Tandem (an open source search algorithm) and proposed a cascade ensemble learning approach for Percolator in PSM filtering. We named this algorithm “self-boosted” Percolator because the

algorithm boosts its performance by repeatedly learning the filtering model using the outputs from its previous iterations. By comparing the proposed algorithm with the state-of-the-art algorithms such as an empirical modelling algorithm called “PeptideProphet” and the original Percolator algorithm on a ground truth dataset and two complex sample datasets, we demonstrated that self-boosted Percolator identifies many more PSMs at the same level of false discovery rate.

- Chapter 7 dealt with feature selection and sample classification from MS-based proteomics studies. Specifically, we described that highly correlated genes and proteins may dominate the feature selection result when using conventional feature selection algorithms. By introducing a k -means clustering procedure, we can bridge filter and wrapper algorithms and at the same time reduce the correlation of features by selecting dissimilar features from each feature cluster. This hybrid system is called “FCGE” because it combines filtering, clustering, and genetic ensemble learning components. We demonstrated that FCGE enhances the biological signal in the dataset by extracting dissimilar m/z markers, and performs consistently better than several other feature selection algorithms across a large number of classification algorithms in all four tested MS-based proteomics datasets.

8.2 Future directions

While this thesis has presented a number of novel ensemble methods and hybrid algorithms for a variety of applications in computational and systems biology, it also indicates promising research directions that can be extended for our future work.

- It is widely accepted that diversity among the individual models is the key driving force for ensemble learning. For example, in ensemble classification, different classifiers are encouraged to give different predictions of a given sample, provided the classification accuracy is maintained at a relatively high level [108]. While there are several studies on model diversity in ensemble classification [24, 30, 108], the effect of diversity in ensemble feature selection has not been explored and warrants further studies.
- Ensemble size is a key parameter in any ensemble learning. It determines the

number of base models used to form the ensemble model, and therefore, may significantly affect the performance of the ensemble model. There are several approaches for determining ensemble size. One approach is to predefine the ensemble size based on some prior knowledge. Another solution is to test different ensemble sizes and select the size that appears to be the best according to certain criteria. While the first approach is inflexible and is unlikely to be optimal in all cases and studies, the second trial and error approach is often computationally intensive and subject to the evaluation criteria used for optimization. Therefore, finding a computationally efficient approach that optimizes the ensemble size in a case-by-case manner could be a key to improve the performance of the ensemble model.

- Beside measuring and analysing gene expressions and protein quantitations, increasingly more studies have been done on global profiling other biological molecules such as phosphorylation of proteins [211] and microRNA regulations [176]. These added layers introduce complexity and new challenges in data analysis. Novel computational algorithms are required to fully utilize and integrate those new high-throughput datasets with large-scale gene expression and proteomics data to reveal the connections and regulations of biological systems and signalling.
- Numerous studies have generated the gene, transcript, and protein profiles of various biological systems and diseases. Those studies and the computational analysis associated with them are often performed separately. There is a strong need for combining multiple data types generated from two or more systems for integrative analysis. We saw a fast growth in this research direction as evidenced by several recent publications [47, 163, 185]. However, the design of effective computational approaches for general integrative analysis is still at its infancy. Research effort in this direction is critical to understand the biological systems at their full scale.

Bibliography

- [1] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2010.
- [2] R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, 2003.
- [3] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.
- [4] D.B. Allison, X. Cui, G.P. Page, and M. Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics*, 7(1):55–65, 2006.
- [5] U. Alon, N. Barkai, DA Notterman, K. Gish, S. Ybarra, D. Mack, and AJ Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, pages 6745–6750, 1999.
- [6] N.L. Anderson and N.G. Anderson. Proteome and proteomics: new technologies, new concepts, and new words. *Electrophoresis*, 19(11):1853–1861, 1998.
- [7] A.S. Andrew, J. Gui, A.C. Sanderson, R.A. Mason, E.V. Morlock, A.R. Schned, K.T. Kelsey, C.J. Marsit, J.H. Moore, and M.R. Karagas. Bladder cancer SNP panel predicts susceptibility and survival. *Hum. Genet.*, 125(5):527–539, 2009.
- [8] P.J. Antsaklis and X.D. Koutsoukos. Hybrid systems: review and recent progress. *Software-Enabled Control*, pages 273–298, 2003.

- [9] S.A. Armstrong, J.E. Staunton, L.B. Silverman, R. Pieters, M.L. den Boer, M.D. Minden, S.E. Sallan, E.S. Lander, T.R. Golub, and S.J. Korsmeyer. Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30(1):41–47, 2002.
- [10] B.M. Balgley, T. Laudeman, L. Yang, T. Song, and C.S. Lee. Comparative evaluation of tandem MS search algorithms using a target-decoy search strategy. *Molecular & Cellular Proteomics*, 6(9):1599, 2007.
- [11] J C Barrett, B Fry, J Maller, and M J Daly. Haploview: analysis and visualization of ld and haplotype maps. *Bioinformatics*, 21:263–265, 2005.
- [12] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7(3-4):559–583, 2000.
- [13] A. Ben-Hur, C.S. Ong, S. Sonnenburg, B. Schölkopf, and G. Rätsch. Support vector machines and kernels for computational biology. *PLoS Computational Biology*, 4(10):e1000173, 2008.
- [14] M. Bern, Y. Cai, and D. Goldberg. Lookup peaks: a hybrid of de novo sequencing and database search for protein identification by tandem mass spectrometry. *Analytical Chemistry*, 79(4):1393–1400, 2007.
- [15] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 97–104. ACM, 2006.
- [16] N. Blow. Transcriptomics: The digital generation. *Nature*, 458(7235):239–242, 2009.
- [17] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.
- [18] G Bontempi. A blocking strategy to improve gene selection for classification of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4:293–300, 2007.

- [19] A.L. Boulesteix and M. Slawski. Stability and aggregation of ranked gene lists. *Briefings in Bioinformatics*, 10(5):556–568, 2009.
- [20] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [21] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [22] L Briollais, Y Wang, I Rajendram, V Onay, E Shi, J Knight, and H Ozcelik. Methodological issues in detecting gene-gene interactions in breast cancer susceptibility: a population-based study in ontario. *BMC Medicine*, 5:22, 2007.
- [23] M. Brosch, L. Yu, T. Hubbard, and J. Choudhary. Accurate and sensitive peptide identification with Mascot Percolator. *Journal of Proteome Research*, 8(6):3176–3181, 2009.
- [24] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.
- [25] A. Bureau, J. Dupuis, K. Falls, K.L. Lunetta, B. Hayward, T.P. Keith, and P. Van Eerdewegh. Identifying snps predictive of phenotype using random forests. *Genetic Epidemiology*, 28(2):171–182, 2005.
- [26] A. Bureau, J. Dupuis, B. Hayward, K. Falls, and P. Van Eerdewegh. Mapping complex traits using random forests. *BMC Genetics*, 4(Suppl 1):S64, 2003.
- [27] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.
- [28] Z. Cai, R. Goebel, M. Salavatipour, and G. Lin. Selecting dissimilar genes for multi-class classification, an application in cancer subtyping. *BMC Bioinformatics*, 8(1):206, 2007.
- [29] S Cantor and M Kattan. Determining the area under the roc curve for a binary diagnostic test. *Medical Decision Making*, 20:468–470, 2000.
- [30] A.M.P. Canuto, M.C.C. Abreu, L. de Melo Oliveira, J.C. Xavier, et al. Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. *Pattern Recognition Letters*, 28(4):472–486, 2007.

- [31] B. Carvalho, H. Bengtsson, T.P. Speed, and R.A. Irizarry. Exploration, normalization, and genotype calls of high-density oligonucleotide snp array data. *Biostatistics*, 8(2):485–499, 2007.
- [32] Tsong Yueh Chen, Joshua W. K. Ho, Huai Liu, and Xiaoyuan Xie. An innovative approach for testing bioinformatics programs using metamorphic testing. *BMC Bioinformatics*, 10:24, 2009.
- [33] X. Chen, S.T. Cheung, S. So, S.T. Fan, C. Barry, J. Higgins, K.M. Lai, J. Ji, S. Dudoit, I.O.L. Ng, et al. Gene expression patterns in human liver cancers. *Molecular Biology of the Cell*, 13(6):1929–1939, 2002.
- [34] X. Chen, C.T. Liu, M. Zhang, and H. Zhang. A forest-based approach to identifying gene and gene–gene interactions. *Proceedings of the National Academy of Sciences*, 104(49):19199–19203, 2007.
- [35] H. Choi, D. Ghosh, and A.I. Nesvizhskii. Statistical validation of peptide identifications in large-scale proteomics using the target-decoy database search strategy and flexible mixture modeling. *Journal of Proteome Research*, 7(01):286–292, 2007.
- [36] L.Y. Chuang, C.H. Yang, J.C. Li, and C.H. Yang. A hybrid BPSO-CGA approach for gene selection and classification of microarray data. *Journal of Computational Biology*, page ahead of print, 2011.
- [37] J.G. Cleary and L.E. Trigg. K*: An instance-based learner using an entropic distance measure. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 108–114, 1995.
- [38] J. Colinge and K.L. Bennett. Introduction to computational proteomics. *PLoS Computational Biology*, 3(7):e114, 2007.
- [39] F.S. Collins, E.D. Green, A.E. Guttmacher, and M.S. Guyer. A vision for the future of genomics research. *Nature*, 422(6934):835–847, 2003.
- [40] E. Corchado, A. Abraham, and J.M. Corchado. *Innovations in hybrid intelligent systems*. Springer Verlag, 2007.

- [41] H.J. Cordell. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genetics*, 11(20):2463, 2002.
- [42] H.J. Cordell. Detecting gene–gene interactions that underlie human diseases. *Nature Reviews Genetics*, 10(6):392–404, 2009.
- [43] J. Cox, N. Neuhauser, A. Michalski, R.A. Scheltema, J.V. Olsen, and M. Mann. Andromeda: a peptide search engine integrated into the maxquant environment. *Journal of Proteome Research*, 10(4):1794–1805, 2011.
- [44] R. Craig and R.C. Beavis. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, 2004.
- [45] R. Craig, JC Cortens, D. Fenyo, and RC Beavis. Using annotated peptide mass spectrum libraries for protein identification. *Journal of Proteome Research*, 5(8):1843–1849, 2006.
- [46] F. Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.
- [47] B.T.S. Da Wei Huang and R.A. Lempicki. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature Protocols*, 4(1):44–57, 2008.
- [48] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156, 1997.
- [49] J.L. DeRisi, V.R. Iyer, and P.O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, 1997.
- [50] M. Dettling and P. Bühlmann. Boosting for tumor classification with gene expression data. *Bioinformatics*, 19(9):1061–1069, 2003.
- [51] E.W. Deutsch, L. Mendoza, D. Shteynberg, T. Farrah, H. Lam, N. Tasman, Z. Sun, E. Nilsson, B. Pratt, B. Prazen, et al. A guided tour of the trans-proteomic pipeline. *Proteomics*, 10(6):1150–1159, 2010.
- [52] R. Díaz-Uriarte and S.A. De Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3, 2006.

- [53] T.G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag, 2000.
- [54] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.
- [55] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2):185–205, 2005.
- [56] S. Dudoit, J. Fridlyand, and T.P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.
- [57] S. Dudoit, Y.H. Yang, M.J. Callow, and T.P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12(1):111–140, 2002.
- [58] K. Dunne, P. Cunningham, and F. Azuaje. Solutions to instability problems with sequential wrapper-based approaches to feature selection. *Technical Report TCD-CS-2002-28, Department of Computer Science, Trinity College, Dublin, Ireland*, 2002.
- [59] E.E. Eichler, J. Flint, G. Gibson, A. Kong, S.M. Leal, J.H. Moore, and J.H. Nadeau. Missing heritability and strategies for finding the underlying causes of complex disease. *Nature Reviews Genetics*, 11(6):446–450, 2010.
- [60] E. Elbeltagi, T. Hegazy, and D. Grierson. Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*, 19(1):43–53, 2005.
- [61] J.E. Elias and S.P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4(3):207–214, 2007.

- [62] J.K. Eng, A.L. McCormack, and J.R. Yates III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5(11):976–989, 1994.
- [63] S Fisher, A Rivera, L Fritsche, G Babadjanova, S Petrov, and B Weber. Assessment of the contribution of cfh and chromosome 10q26 amd susceptibility loci in a russian population isolate. *British Journal of Ophthalmology*, 91:576–578, 2007.
- [64] A.M. Frank, M.M. Savitski, M.L. Nielsen, R.A. Zubarev, and P.A. Pevzner. De novo peptide sequencing and identification with precision mass spectrometry. *Journal Proteome Research*, 6(1):114–123, 2007.
- [65] Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 124–133, 1999.
- [66] L.Y. Geer, S.P. Markey, J.A. Kowalak, L. Wagner, M. Xu, D.M. Maynard, X. Yang, W. Shi, and S.H. Bryant. Open mass spectrometry search algorithm. *Journal of Proteome Research*, 3(5):958–964, 2004.
- [67] J. Gertheiss and G. Tutz. Supervised feature selection in mass spectrometry-based proteomic profiling by blockwise boosting. *Bioinformatics*, 25(8):1076–1077, 2009.
- [68] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences*, 97(22):12079–12084, 2000.
- [69] P. Geurts, M. Fillet, D. De Seny, M.A. Meuwis, M. Malaise, M.P. Merville, and L. Wehenkel. Proteomic mass spectra classification using decision tree based ensemble methods. *Bioinformatics*, 21(14):3138–3145, 2005.
- [70] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

- [71] C.S. Greene, N.M. Penrod, J. Kiralis, and J.H. Moore. Spatially uniform ReliefF (SURF) for computationally-efficient filtering of gene-gene interactions. *BioData Mining*, 2(1):5, 2009.
- [72] K.L. Gunderson, F.J. Steemers, G. Lee, L.G. Mendoza, and M.S. Chee. A genome-wide scalable SNP genotyping assay using microarray technology. *Nature Genetics*, 37(5):549–554, 2005.
- [73] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [74] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.
- [75] M.T. Hagan, H.B. Demuth, M.H. Beale, and Boulder University of Colorado. *Neural network design*. PWS Pub, 1996.
- [76] L W Hahn, M D Ritchie, and J H Moore. Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. *Bioinformatics*, 19:376–382, 2003.
- [77] J L Haines, M A Hauser, S Schmidt, W K Scott, and L M Olson. Complement factor h variant increases the risk of age-related macular degeneration. *Science*, 308:419–421, 2005.
- [78] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [79] X. Han, A. Aslanian, and J.R. Yates III. Mass spectrometry for proteomics. *Current Opinion in Chemical Biology*, 12(5):483–490, 2008.
- [80] B. Hanczar, M. Courtine, A. Benis, C. Hennegar, K. Clément, and J.D. Zucker. Improving classification of microarray data using prototype-based feature selection. *ACM SIGKDD Explorations Newsletter*, 5(2):23–30, 2003.
- [81] T. Hastie, R. Tibshirani, and J. Friedman. Ensemble learning. *The Elements of Statistical Learning*, pages 605–624, 2009.

- [82] Z. He and W. Yu. Review article: Stable feature selection for biomarker discovery. *Computational Biology and Chemistry*, 34(4):215–225, 2010.
- [83] A G Heidema, J M Boer, N Nagelkerke, E C Mariman, A D L van der, and E J Feskens. The challenge for genetic epidemiologists: how to analyze large numbers of snps in relation to complex diseases. *BMC Genetics*, 7:23, 2006.
- [84] M. Hilario and A. Kalousis. Approaches to dimensionality reduction in proteomic biomarker studies. *Briefings in Bioinformatics*, 9(2):102–118, 2008.
- [85] J.N. Hirschhorn, M.J. Daly, et al. Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2):95–108, 2005.
- [86] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
- [87] J. Hoh, A. Wille, and J. Ott. Trimming, weighting, and grouping SNPs in human case-control association studies. *Genome Res.*, 11(12):2115–2119, 2001.
- [88] M.M. Iles. What can genome-wide association studies tell us about the genetics of common disease? *PLoS Genet.*, 4(2):e33, 2008.
- [89] G. Izmirlian. Application of the random forest classification algorithm to a seldi-tof proteomics study in the setting of a cancer prevention trial. *Annals of the New York Academy of Sciences*, 1020(1):154–174, 2004.
- [90] J. Jaeger, R. Sengupta, and WL Ruzzo. Improved gene selection for classification of microarrays. In *Pacific Symposium on Biocomputing*, pages 53–64, 2003.
- [91] R. Jiang, W. Tang, X. Wu, and W. Fu. A random forest approach to the detection of epistatic interactions in case-control studies. *BMC Bioinformatics*, 10(Suppl 1):S65, 2009.
- [92] T. Jirapech-Umpai and S. Aitken. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6(1):148, 2005.

- [93] K. Jong, J. Mary, A. Cornuéjols, E. Marchiori, and M. Sebag. Ensemble feature ranking. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 267–278. Springer-Verlag New York, Inc., 2004.
- [94] L. Käll, J.D. Canterbury, J. Weston, W.S. Noble, and M.J. MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4(11):923–925, 2007.
- [95] M. Kallberg and H. Lu. An improved machine learning protocol for the identification of correct sequest search results. *BMC bioinformatics*, 11(1):591, 2010.
- [96] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116, 2007.
- [97] E.A. Kapp, F. Schütz, L.M. Connolly, J.A. Chakel, J.E. Meza, C.A. Miller, D. Fenyo, J.K. Eng, J.N. Adkins, G.S. Omenn, et al. An evaluation, comparison, and accurate benchmarking of several publicly available ms/ms search algorithms: sensitivity and specificity analysis. *Proteomics*, 5(13):3475–3490, 2005.
- [98] E. Keedwell and A. Narayanan. Discovering gene networks with a neural-genetic hybrid. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(3):231–242, 2005.
- [99] E. Keedwell and A. Narayanan. *Intelligent bioinformatics*. Wiley Online Library, 2005.
- [100] A. Keller, J. Eng, N. Zhang, X. Li, and R. Aebersold. A uniform proteomics ms/ms analysis platform utilizing open xml file formats. *Molecular Systems Biology*, 1(1), 2005.
- [101] A. Keller, AI Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytical Chemistry*, 74(20):5383–5392, 2002.
- [102] J Kittler, M Hatef, R P Duin, and J Mates. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

- [103] R.J. Klein, C. Zeiss, E.Y. Chew, J.Y. Tsai, R.S. Sackler, C. Haynes, A.K. Henning, J.P. SanGiovanni, S.M. Mane, S.T. Mayne, et al. Complement factor H polymorphism in age-related macular degeneration. *Science*, 308(5720):385–389, 2005.
- [104] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
- [105] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25–41, 2000.
- [106] L I Kuncheva and L C Jain. Designing classifier fusion system by genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4:327–336, 2000.
- [107] L.I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, 2004.
- [108] L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.
- [109] H. Lam, E.W. Deutsch, J.S. Eddes, J.K. Eng, N. King, S.E. Stein, and R. Aebersold. Development and validation of a spectral library searching method for peptide identification from ms/ms. *Proteomics*, 7(5):655–667, 2007.
- [110] L. Lam and SY Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27(5):553–568, 1997.
- [111] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J.A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, et al. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, 2006.
- [112] J.W. Lee, J.B. Lee, M. Park, and S.H. Song. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, 48(4):869–885, 2005.

- [113] K.E. Lee, N. Sha, E.R. Dougherty, M. Vannucci, and B.K. Mallick. Gene selection: a bayesian variable selection approach. *Bioinformatics*, 19(1):90–97, 2003.
- [114] I. Levner. Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics*, 6(1):68, 2005.
- [115] L. Li, T.A. Darden, CR Weingberg, AJ Levine, and L.G. Pedersen. Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. *Combinatorial Chemistry & High Throughput Screening*, 4(8):727–739, 2001.
- [116] L. Li, D.M. Umbach, P. Terry, and J.A. Taylor. Application of the ga/knn method to seldi proteomics data. *Bioinformatics*, 20(10):1638–1640, 2004.
- [117] L. Li, C.R. Weinberg, T.A. Darden, and L.G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17(12):1131–1142, 2001.
- [118] B. Liu, Q. Cui, T. Jiang, and S. Ma. A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC bioinformatics*, 5(1):136, 2004.
- [119] H. Liu, R.G. Sadygov, and J.R. Yates III. A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Analytical Chemistry*, 76(14):4193–4201, 2004.
- [120] P.M. Long and V.B. Vega. Boosting and microarray data. *Machine Learning*, 52(1):31–44, 2003.
- [121] K. Lunetta, L.B. Hayward, J. Segal, and P. Van Eerdewegh. Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics*, 5(1):32, 2004.
- [122] M.I. McCarthy, G.R. Abecasis, L.R. Cardon, D.B. Goldstein, J. Little, JP Ioannidis, and J.N. Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature Reviews Genetics*, 9(5):356–369, 2008.

- [123] B.A. McKinney, J.E. Crowe Jr, J. Guo, and D. Tian. Capturing the spectrum of interaction effects in genetic association studies by simulated evaporative cooling network analysis. *PLoS Genet.*, 5(3):e1000432, 2009.
- [124] B.A. McKinney, D.M. Reif, M.D. Ritchie, and J.H. Moore. Machine learning for detecting gene-gene interactions: a review. *Applied Bioinformatics*, 5(2):77–88, 2006.
- [125] BA McKinney, DM Reif, BC White, JE Crowe Jr, and JH Moore. Evaporative cooling feature selection for genotypic data involving interactions. *Bioinformatics*, 23(16):2113–2120, 2007.
- [126] J.A. Mead, L. Bianco, and C. Bessant. Recent developments in public proteomic MS repositories and pipelines. *Proteomics*, 9(4):861–881, 2009.
- [127] L E Mechanic, B T Luke, J E Goodman, S J Chanock, and C C Harris. Polymorphism interaction analysis (pia): a method for investigating complex gene-gene interactions. *BMC Bioinformatics*, 9:146, 2008.
- [128] Y. Meng, Y. Yu, L.A. Cupples, L. Farrer, and K. Lunetta. Performance of random forest when snps are in linkage disequilibrium. *BMC Bioinformatics*, 10(1):78, 2009.
- [129] T.M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [130] J. Moore and B. White. Tuning relief for genome-wide genetic analysis. *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 166–175, 2007.
- [131] J.H. Moore, F.W. Asselbergs, and S.M. Williams. Bioinformatics challenges for genome-wide association studies. *Bioinformatics*, 26(4):445–455, 2010.
- [132] J.H. Moore, L.W. Hahn, M.D. Ritchie, T.A. Thornton, and B.C. White. Application of genetic algorithms to the discovery of complex models for simulation studies in human genetics. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1150–1155, 2002.
- [133] J.H. Moore and S.M. Williams. Epistasis and its implications for personal genetics. *Am. J. Hum. Genet.*, 85(3):309–320, 2009.

- [134] A A Motsinger and M D Ritchie. Multifactor dimensionality reduction: an analysis strategy for modelling and detecting gene-gene interactions in human genetics and pharmacogenomics studies. *Human Genomics*, 2:318–328, 2006.
- [135] A.A. Motsinger-Reif, S.M. Dudek, L.W. Hahn, and M.D. Ritchie. Comparison of approaches for machine-learning optimization of neural networks for detecting gene-gene interactions in genetic epidemiology. *Genetic Epidemiology*, 32(4):325–340, 2008.
- [136] S.K. Musani, D. Shriner, N. Liu, R. Feng, C.S. Coffey, N. Yi, H.K. Tiwari, and D.B. Allison. Detection of gene \times gene interactions in genome-wide association studies of human population data. *Hum. Hered.*, 63(2):67–84, 2007.
- [137] N. Nagarajan and G. Yona. Automatic prediction of protein domains from sequence information using a hybrid learning system. *Bioinformatics*, 20(9):1335–1360, 2004.
- [138] M R Nelson, S L Kardia, R E Ferrell, and C F Sing. A combinatorial partitioning method to identify multilocus genotypic partitions that predict quantitative trait variation. *Genome Research*, 11:458–470, 2001.
- [139] A.I. Nesvizhskii and R. Aebersold. Interpretation of shotgun proteomic data: the problem inference problem. *Molecular & Cellular Proteomics*, 4(10):1419–1440, 2005.
- [140] A.I. Nesvizhskii, A. Keller, E. Kolker, and R. Aebersold. A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry*, 75(17):4646–4658, 2003.
- [141] A.I. Nesvizhskii, F.F. Roos, J. Grossmann, M. Vogelzang, J.S. Eddes, W. Gruissem, S. Baginsky, and R. Aebersold. Dynamic spectrum quality assessment and iterative computational analysis of shotgun proteomic data. *Molecular & Cellular Proteomics*, 5(4):652, 2006.
- [142] D.M. Nielsen, M.G. Ehm, and B.S. Weir. Detecting marker-disease association by testing for hardy-weinberg disequilibrium at a marker locus. *The American Journal of Human Genetics*, 63(5):1531–1540, 1998.

- [143] W.M. Old, K. Meyer-Arendt, L. Aveline-Wolf, K.G. Pierce, A. Mendoza, J.R. Sevensky, K.A. Resing, and N.G. Ahn. Comparison of label-free methods for quantifying human proteins by shotgun proteomics. *Molecular & Cellular Proteomics*, 4(10):1487–1502, 2005.
- [144] S.E. Ong, B. Blagoev, I. Kratchmarova, D.B. Kristensen, H. Steen, A. Pandey, and M. Mann. Stable isotope labeling by amino acids in cell culture, silac, as a simple and accurate approach to expression proteomics. *Molecular & Cellular Proteomics*, 1(5):376–386, 2002.
- [145] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11(1):169–198, 1999.
- [146] M.Y. Park and T. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1):30–50, 2008.
- [147] S.D. Patterson and R.H. Aebersold. Proteomics: the first decade and beyond. *Nature Genetics*, 33:311–323, 2003.
- [148] T.A. Pearson and T.A. Manolio. How to interpret a genome-wide association study. *JAMA: the journal of the American Medical Association*, 299(11):1335–1344, 2008.
- [149] S. Peng, Q. Xu, X.B. Ling, X. Peng, W. Du, and L. Chen. Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. *FEBS Letters*, 555(2):358–362, 2003.
- [150] D.N. Perkins, D.J.C. Pappin, D.M. Creasy, and J.S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, 1999.
- [151] E.F. Petricoin and L.A. Liotta. Seldi-tof-based serum proteomic pattern diagnostics for early detection of cancer. *Current Opinion in Biotechnology*, 15(1):24–30, 2004.
- [152] E.F. Petricoin, D.K. Ornstein, C.P. Paweletz, A. Ardekani, P.S. Hackett, B.A. Hitt, A. Velasco, C. Trucco, L. Wiegand, K. Wood, et al. Serum proteomic patterns for detection of prostate cancer. *Journal of the National Cancer Institute*, 94(20):1576–1578, 2002.

- [153] E.F. Petricoin III, A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, et al. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359(9306):572–577, 2002.
- [154] P.C. Phillips. Epistasis – The essential role of gene interactions in the structure and evolution of genetic systems. *Nat. Rev. Genet.*, 9(11):855–867, 2008.
- [155] F. Pompanon, A. Bonin, E. Bellemain, and P. Taberlet. Genotyping errors: causes, consequences and solutions. *Nature Reviews Genetics*, 6(11):847–846, 2005.
- [156] G. Potamias, L. Koumakis, and V. Moustakis. Gene selection via discretized gene-expression profiles and greedy feature-elimination. *Methods and Applications of Artificial Intelligence*, pages 256–266, 2004.
- [157] J. Prados, A. Kalousis, J.C. Sanchez, L. Allard, O. Carrette, and M. Hilario. Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents. *Proteomics*, 4(8):2320–2332, 2004.
- [158] Y. Qi, W. Niu, T. Zhu, W. Zhou, and C. Qiu. Synergistic effect of the genetic polymorphisms of the renin–angiotensin–aldosterone system on high-altitude pulmonary edema: a study from Qinghai-Tibet altitude. *Eur. J. Epidemiol.*, 23(2):143–152, 2008.
- [159] Y. Qu, B.L. Adam, Y. Yasui, M.D. Ward, L.H. Cazares, P.F. Schellhammer, Z. Feng, O.J. Semmes, and G.L. Wright Jr. Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients. *Clinical Chemistry*, 48(10):1835–1843, 2002.
- [160] J. Quackenbush. Microarray data normalization and transformation. *Nature Genetics*, 32(supp):496–501, 2002.
- [161] N. Rabbee and T.P. Speed. A genotype calling algorithm for affymetrix SNP arrays. *Bioinformatics*, 22(1):7–12, 2005.
- [162] HW Resson, RS Varghese, SK Drake, GL Hortin, M. Abdel-Hamid, CA Lofredo, and R. Goldman. Peak selection from maldi-tof mass spectra using ant colony optimization. *Bioinformatics*, 23(5):619–626, 2007.

- [163] D.R. Rhodes and A.M. Chinnaiyan. Integrative analysis of the cancer transcriptome. *Nature Genetics*, 37:S31–S37, 2005.
- [164] M.D. Ritchie, L.W. Hahn, and J.H. Moore. Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. *Genetic Epidemiology*, 24(2):150–157, 2003.
- [165] M.D. Ritchie, B.C. White, J.S. Parker, L.W. Hahn, and J.H. Moore. Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. *BMC bioinformatics*, 4:28, 2003.
- [166] M. Robnik-Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53(1):23–69, 2003.
- [167] P.L. Ross, Y.N. Huang, J.N. Marchese, B. Williamson, K. Parker, S. Hattan, N. Khainovski, S. Pillai, S. Dey, S. Daniels, et al. Multiplexed protein quantitation in *saccharomyces cerevisiae* using amine-reactive isobaric tagging reagents. *Molecular & Cellular Proteomics*, 3(12):1154–1169, 2004.
- [168] R. Ruiz, J.C. Riquelme, and J.S. Aguilar-Ruiz. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition*, 39(12):2383–2392, 2006.
- [169] D Ruta and B Gabrys. Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In *Proceedings of MCS 2001, LNCS 2096*, pages 399–408, 2001.
- [170] D Ruta and B Gabrys. Classifier selection for majority voting. *Information Fusion*, 6:63–81, 2005.
- [171] Y. Saeys, T. Abeel, and Y. Peer. Robust feature selection using ensemble feature selection techniques. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases-Part II*, pages 313–325. Springer-Verlag, 2008.
- [172] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

- [173] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
- [174] M. Schena, D. Shalon, RW Davis, and PO Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470, 1995.
- [175] S Schmidt, M A Hauser, W K Scott, E A Postel, and A Agarwal. Cigarette smoking strongly modifies the association of loc387715 and age-related macular degeneration. *The American Journal of Human Genetics*, 78:852–864, 2006.
- [176] R. Shalgi, D. Lieber, M. Oren, and Y. Pilpel. Global and local architecture of the mammalian microRNA–transcription factor regulatory network. *PLoS Computational Biology*, 3(7):e131, 2007.
- [177] Q. Shen, W.M. Shi, and W. Kong. Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. *Computational Biology and Chemistry*, 32(1):53–60, 2008.
- [178] Q. Shen, W.M. Shi, W. Kong, and B.X. Ye. A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification. *Talanta*, 71(4):1679–1683, 2007.
- [179] I.V. Shilov, S.L. Seymour, A.A. Patel, A. Loboda, W.H. Tang, S.P. Keating, C.L. Hunter, L.M. Nuwaysir, and D.A. Schaeffer. The paragon algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Molecular & Cellular Proteomics*, 6(9):1638–1655, 2007.
- [180] R. Sladek, G. Rocheleau, J. Rung, C. Dina, L. Shen, D. Serre, P. Boutin, D. Vincent, A. Belisle, S. Hadjadj, et al. A genome-wide association study identifies novel risk loci for type 2 diabetes. *Nature*, 445(7130):881–885, 2007.
- [181] G.K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):3, 2004.

- [182] R.L. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics*, 19(12):1484–1491, 2003.
- [183] M. Spivak, J. Weston, L. Bottou, L. Käll, and W.S. Noble. Improvements to the percolator algorithm for peptide identification from shotgun proteomics data sets. *Journal of Proteome Research*, 8(7):3737–3745, 2009.
- [184] D. Stekel. *Microarray bioinformatics*. Cambridge University Press, 2003.
- [185] K. Stemke-Hale, A.M. Gonzalez-Angulo, A. Lluch, R.M. Neve, W.L. Kuo, M. Davies, M. Carey, Z. Hu, Y. Guan, A. Sahin, et al. An integrative genomic and proteomic analysis of pik3ca, pten, and akt mutations in breast cancer. *Cancer Research*, 68(15):6084, 2008.
- [186] A. Sturn, J. Quackenbush, and Z. Trajanoski. Genesis: cluster analysis of microarray data. *Bioinformatics*, 18(1):207–208, 2002.
- [187] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, E.S. Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.
- [188] AC Syvanen. Accessing genetic variation: genotyping single nucleotide polymorphisms. *Nature Reviews Genetics*, 2(12):930–942, 2001.
- [189] A.C. Tan and D. Gilbert. Ensemble machine learning on gene expression data for cancer classification. *Applied Bioinformatics*, 2:75–84, 2003.
- [190] The Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447:661–678, 2007.
- [191] D. Thomas. Gene–environment-wide association studies: Emerging approaches. *Nat. Rev. Genet.*, 11:259–272, 2010.
- [192] Y Tomita, S Tomida, Y Hasegawa, Y Suzuki, T Shirakawa, T Kobayashi, and H Honda. Artificial neural network approach for selection of susceptible single

- nucleotide polymorphisms and construction of prediction model on childhood allergic asthma. *BMC Bioinformatics*, 5:120, 2004.
- [193] A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1):83–98, 2005.
- [194] D.R. Velez, B.C. White, A.A. Motsinger, W.S. Bush, M.D. Ritchie, S.M. Williams, and J.H. Moore. A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. *Genetic Epidemiology*, 31(4):306–315, 2007.
- [195] P. Wang, P. Yang, J. Arthur, and J.Y.H. Yang. A dynamic wavelet-based algorithm for pre-processing tandem mass spectrometry data. *Bioinformatics*, 26(18):2242–2249, 2010.
- [196] Y. Wang, F.S. Makedon, J.C. Ford, and J. Pearlman. Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. *Bioinformatics*, 21(8):1530–1537, 2005.
- [197] G.I. Webb and Z. Zheng. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering*, 16(8):980–991, 2004.
- [198] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [199] D.H. Wolpert. Stacked generalization*. *Neural Networks*, 5(2):241–259, 1992.
- [200] B. Wu, T. Abbott, D. Fishman, W. McMurray, G. Mor, K. Stone, D. Ward, K. Williams, and H. Zhao. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics*, 19(13):1636–1643, 2003.
- [201] E.P. Xing, M.I. Jordan, and R.M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608. Morgan Kaufmann Publishers Inc., 2001.

- [202] J. Yang and V.G. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2):44–49, 1998.
- [203] P. Yang, J. Ho, Y. Yang, and B. Zhou. Gene-gene interaction filtering with ensemble of filters. *BMC Bioinformatics*, 12(Suppl 1):S10, 2011.
- [204] P. Yang and Z. Zhang. A clustering based hybrid system for mass spectrometry data analysis. In *Proceedings of the Third IAPR International Conference on Pattern Recognition in Bioinformatics*, pages 98–109. Springer-Verlag, 2008.
- [205] 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*, 30(4):e15, 2002.
- [206] Y.H. Yang, Y. Xiao, and M.R. Segal. Identifying differentially expressed genes from microarray experiments via statistic synthesis. *Bioinformatics*, 21(7):1084–1093, 2005.
- [207] Y. Yasui, M. Pepe, M.L. Thompson, B.L. Adam, G.L. Wright, Y. Qu, J.D. Potter, M. Winget, M. Thornquist, and Z. Feng. A data-analytic strategy for protein biomarker discovery: profiling of high-dimensional proteomic data for cancer detection. *Biostatistics*, 4(3):449–463, 2003.
- [208] Y. Ye, X. Zhong, and H. Zhang. A genome-wide tree-and forest-based association analysis of comorbidity of alcoholism and smoking. *BMC Genetics*, 6(Suppl 1):S135, 2005.
- [209] K.Y. Yeung, R.E. Bumgarner, and A.E. Raftery. Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics*, 21(10):2394–2402, 2005.
- [210] J. Yu and X.W. Chen. Bayesian neural network approaches to ovarian cancer identification from high-resolution mass spectrometry data. *Bioinformatics*, 21(suppl 1):i487, 2005.
- [211] Y. Yu, S.O. Yoon, G. Poulgiannis, Q. Yang, X.M. Ma, J. Villén, N. Kubica, G.R. Hoffman, L.C. Cantley, S.P. Gygi, et al. Phosphoproteomic analysis identifies

- grb10 as an mtorc1 substrate that negatively regulates insulin signaling. *Science*, 332(6035):1322–1326, 2011.
- [212] H. Zhang and G. Bonney. Use of classification trees for association studies. *Genetic Epidemiology*, 19(4):323–332, 2000.
- [213] H. Zhang, C.Y. Yu, and B. Singer. Cell and tumor classification using gene expression data: construction of forests. *Proceedings of the National Academy of Sciences*, 100(7):4168–4172, 2003.
- [214] K. Zhang, Z.S. Qin, J.S. Liu, T. Chen, M.S. Waterman, and F. Sun. Haplotype block partitioning and tag snp selection using genotype data and their applications to association studies. *Genome Research*, 14(5):908, 2004.
- [215] X. Zhang, X. Lu, Q. Shi, X. Xu, H. Leung, L. Harris, J. Iglehart, A. Miron, J. Liu, and W. Wong. Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*, 7(1):197, 2006.
- [216] Z. Zhang, P. Yang, X. Wu, and C. Zhang. An agent-based hybrid system for microarray data analysis. *Intelligent Systems, IEEE*, 24(5):53–63, 2009.
- [217] Z. Zhang, S. Zhang, M.Y. Wong, N.J. Wareham, and Q. Sha. An ensemble learning approach jointly modeling main and interaction effects in genetic association studies. *Genetic Epidemiology*, 32(4):285–300, 2008.

Index

- bagging, 15
- base classifier, 13
- blocking, 50, 78
- boosting, 16
 - AdaBoost, 17
 - LogitBoost, 17
- combinatorial methods, 29, 44
- curse-of-dimensionality, 28, 72
- curse-of-sparsity, 28, 72
- decision tree, 16, 56
- differentially expressed (DE) genes, 5
- ensemble method, 11
- false discovery rate (FDR), 92
- feature, 13
- filter, 19
 - χ^2 -test, 77
 - gain ratio, 78
 - information gain, 77, 119
 - ReliefF, 30, 119
 - symmetrical uncertainty, 77
- gene-gene interactions, 4, 24, 29, 44
- genetic algorithm, 24
- genetic ensemble (GE), 8, 46, 73
 - filtering, clustering, and genetic ensemble selection (FCGE), 108
 - multi-filter enhanced genetic ensemble (MF-GE), 73
- genome, 2
- genome-wide association (GWA), 3, 28
- genomics, 2
- hybrid algorithm, 12, 25
- hypothesis, 13
- instability, 19
- majority voting, 14, 51, 78
- mass spectrometry (MS), 90
 - MS-based proteomics, 3, 90
- mass-to-charge (m/z), 106
- microarray, 3, 71
- nearest neighbour, 30, 56
- proteome, 2
- proteomics, 2, 90
 - peptide identification, 90
 - peptide-spectrum matches (PSMs), 91
- PSM post-processing, 92
 - PeptideProphet, 92
 - Percolator, 92
 - self-boosted Percolator, 92
- random forests, 13, 56, 76, 120
- random subspace, 15

single nucleotide polymorphism (SNP), 3,
28, 45

stability, 19

support vector machine (SVM), 92, 120
SVM-RFE, 21

transcriptome, 2, 71

transcriptomics, 2, 71