11-16-2015

Computational Methods for Biomarker Identification in Complex Disease

Amin Ahmadi Adl
University of South Florida, amin1@mail.usf.edu

Follow this and additional works at: http://scholarcommons.usf.edu/etd
Part of the Bioinformatics Commons, and the Computer Sciences Commons

Scholar Commons Citation
Ahmadi Adl, Amin, "Computational Methods for Biomarker Identification in Complex Disease" (2015). Graduate Theses and Dissertations.
http://scholarcommons.usf.edu/etd/5895

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
Computational Methods for Biomarker Identification in Complex Disease

by

Amin Ahmadi Adl

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Co-Major Professor: Xiaoning Qian, Ph.D.
Co-Major Professor: Lawrence O. Hall, Ph.D.
Dmitry Goldgof, Ph.D.
Bo Zeng, Ph.D.
Hye-Seung Lee, Ph.D.

Date of Approval:
August 31, 2015

Keywords: computational biology, interaction, heterogeneity, disease stratification, networks

Copyright © 2015, Amin Ahmadi Adl
Dedication

In dedication to my beautiful wife Bahar, my brothers Ali and Iman, my lovely mom Azam and my dad Ghasem who showed me the joy of science and reasoning.
Acknowledgments

I would like to express the deepest appreciation to my advisor Dr. Xiaoning Qian. Without his guidance and persistent help this dissertation would not have been possible. I would also like to specially thank Prof. Lawrence Hall for his invaluable support.

In addition, I would like to thank committee members Prof. Dimitry Goldgof, Dr. Bo Zeng, and Dr. Hye Seung Lee and also the defense Chairperson Dr. Lingling Fan.
Table of Contents

List of Tables .................................................................................................................. iii
List of Figures .................................................................................................................... iv
Abstract ............................................................................................................................. vi

Chapter 1 Introduction ....................................................................................................... 1
  1.1 Background .................................................................................................................. 1
  1.2 Problem Statement ....................................................................................................... 2
    1.2.1 Biomarker Identification for Disease Outcome .................................................. 2
    1.2.2 Biomarker Identification for Subtypes of Heterogeneous Diseases ....................... 5
  1.3 Our Contribution ......................................................................................................... 6
  1.4 Organization ............................................................................................................... 7

Chapter 2 Literature Review ............................................................................................. 9
  2.1 Feature Selection for High Throughput Biomedical Data ............................................. 9
  2.2 Clustering Methods for Tumor Stratification .............................................................. 11
  2.3 Bi-clustering Methods for Simultaneous Tumor Stratification and Biomarker Identifi-
    cation ........................................................................................................................... 14

Chapter 3 Detecting Pairwise Interactive Effects For Continuous Random Variables ........ 21
  3.1 Existing Methods ........................................................................................................ 22
    3.1.1 Information Theoretic Measures ........................................................................... 22
      3.1.1.1 Quantization Methods ................................................................................. 23
      3.1.1.2 Simplified Gaussian Assumptions ............................................................... 24
    3.1.2 Classification-based Methods .............................................................................. 24
      3.1.2.1 Logistic Regression ..................................................................................... 24
      3.1.2.2 Relative Expression Analysis ...................................................................... 25
      3.1.2.3 Linear Classifiers ....................................................................................... 25
  3.2 Our Proposed Methods ............................................................................................... 25
    3.2.1 New Information Theoretic Measures ................................................................. 25
      3.2.1.1 Supervised Quantization ........................................................................... 26
      3.2.1.2 Estimating $P(y|x)$ by Supervised Learning ............................................... 27
    3.2.2 Methods Based on Classification Accuracy ......................................................... 28
    3.2.3 Methods Based on Association Analysis ............................................................ 29
    3.2.4 Performance Evaluation by Simulated Data ......................................................... 30
  3.3 Run Time Analysis ...................................................................................................... 34
  3.4 Statistical Significance ............................................................................................... 37
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Gene set enrichment analysis for identified biomarkers by network-based ranking.</td>
<td>46</td>
</tr>
<tr>
<td>Table 2</td>
<td>Comparing the performance of the network-based spectral feature ranking with individual-based feature ranking based on the DPT-1 dataset.</td>
<td>48</td>
</tr>
<tr>
<td>Table 3</td>
<td>Final sets of bio-markers and their corresponding performance for the DPT-1 dataset using individual ranking, network-based ranking, and exhaustive search methods.</td>
<td>48</td>
</tr>
<tr>
<td>Table 4</td>
<td>The NMI calculated for subtypes identified by each method compared to the actual subtypes of Breast Cancer, based on 20 different sub-sampled datasets.</td>
<td>55</td>
</tr>
<tr>
<td>Table 5</td>
<td>Biomarkers associated with Breast Cancer subtypes identified by our GO Biological Processes representation method.</td>
<td>56</td>
</tr>
<tr>
<td>Table 6</td>
<td>Performance comparison based on curated subtypes for Uterine cancer.</td>
<td>67</td>
</tr>
<tr>
<td>Table 7</td>
<td>Performance comparison with Ovarian cancer.</td>
<td>70</td>
</tr>
<tr>
<td>Table 8</td>
<td>Important identified genes for each subtype based on Ovarian cancer dataset together with the enriched pathways.</td>
<td>72</td>
</tr>
<tr>
<td>Table A.9</td>
<td>Final sets of biomarkers for both network-based ranking and individual-based ranking.</td>
<td>92</td>
</tr>
<tr>
<td>Table A.10</td>
<td>Gene set enrichment analysis for biomarkers identified by the network-based ranking method.</td>
<td>92</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>An illustration of different computational approaches for biomarker identification.</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2</td>
<td>An illustration of high interactive effect between two features $x_1$ and $x_2$ with respect to the outcome $y$.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3</td>
<td>All the methods studied in this chapter with their relationships illustrated in a tree graph.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Illustration for how different quantization methods work.</td>
<td>26</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Scatter plots illustrating the relationships between two interacting variables associated with the outcome $y$ in our simulated datasets (Exaggerated) coded by different shapes.</td>
<td>31</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Simulation dataset results.</td>
<td>32</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Simulation dataset results with 20 and 140 samples.</td>
<td>33</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Run time comparison of all methods using simulated datasets.</td>
<td>35</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Run time comparison of all methods of this chapter for different sample sizes using simulated datasets.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Performance comparison between individual-based and network-based ranking for 20 simulated datasets in which we have weak individual effects and significant synergistic effects.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Performance comparison between individual-based and network-based ranking for 20 simulated datasets in which we have both significant individual effects and significant synergistic effects.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 12</td>
<td>The AUC for predicting breast cancer metastasis based on network-based ranking and individual-based ranking.</td>
<td>45</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Venn diagram illustrating the identified biomarkers using different methods.</td>
<td>49</td>
</tr>
<tr>
<td>Figure 14</td>
<td>The pairwise distances between 504 Breast Cancer tumors are depicted with a gray scale map (the distance values are quantile normalized to a normal distribution to achieve better contrast for visualization purposes).</td>
<td>51</td>
</tr>
</tbody>
</table>
Figure 15  Tumor stratification by finding densely connected sub-bipartite-graphs (relaxed bicliques) in a bipartite graph representation of the data which are constrained by prior knowledge in the gene-gene interaction network. .......................... 58

Figure 16  Performance comparison with Ovarian cancer by survival analysis. ..................... 69

Figure A.17  Simulation dataset results with 200 randomly simulated variables in each sample. . 91
Abstract

In a modern systematic view of biology, cell functions arise from the interaction between molecular components. One of the challenging problems in systems biology with high-throughput measurements is discovering the important components involved in the development and progression of complex diseases, which may serve as biomarkers for accurate predictive modeling and as targets for therapeutic purposes. Due to the non-linearity and heterogeneity of these complex diseases, traditional biomarker identification approaches have had limited success at finding clinically useful biomarkers. In this dissertation we propose novel methods for biomarker identification that explicitly take into account the non-linearity and heterogeneity of complex diseases. We first focus on the methods to deal with non-linearity by taking into account the interactions among features with respect to the disease outcome of interest. We then focus on the methods for finding disease subtypes with their subtype-specific biomarkers for heterogeneous diseases, where we show how prior biological knowledge and simultaneous disease stratification and personalized biomarker identification can help achieve better performance. We develop novel computational methods for more accurate and robust biomarker identification including methods for estimating the interactive effects, a network-based feature ranking algorithm that takes into account the interactive effects between biomarkers, different approaches for finding distances between somatic mutation profiles for better disease stratification using prior knowledge, and a network-regularized bi-clique finding algorithm for simultaneous subtype and biomarker identification. Our experimental results show that our proposed methods perform better than the state-of-the-art methods for both problems.
Chapter 1
Introduction

1.1 Background

Many of complex diseases such as cancer and diabetes are influenced by a combination of several genetic and environmental factors. One of the major challenges in modern biomedicine is to understand the biological processes that drive such complex diseases. To enable systematic study of complex biological processes in terms of interacting molecular components, the focus has been on discovering the important cellular components, including genes or proteins, that are involved in development and progression of complex diseases. These components can serve as biomarkers for better disease diagnosis and prognosis and as targets for therapeutic purposes. Furthermore, discovering these biomarkers can help understand aberrant changes that cause the disease, and enables targeted therapeutic and personalized medicine strategies. However, screening and validation using biomedical experiments or clinical trials is costly and demands a lot of resources. With the ever-increasing high-throughput data from systematic profiling of thousands of patients for diverse genome-scale omics measurements, including mRNA expression, micro-RNA (miRNA) expression, DNA sequences, somatic mutation profiles, and DNA copy number [1], there is a pressing need to develop computational approaches that help in identifying important components or biomarkers related to a certain phenotype of interest, among tens of thousands of measurements [2, 3].

Living systems are non-linear with highly interacting cellular components and heterogeneous with highly different molecular activities among similar phenotypic states [4, 5, 6]. Many complex diseases, such as cancer and diabetes, are conjectured to have complicated underlying disease mechanisms, which are neither static nor manifest as a linear systems [4, 5, 7] and exhibit complex genotypic and phenotypic heterogeneity within patients diagnosed with the same disease [6, 8, 9, 10, 11]. Multiple candidate risk factors, either genetic or environmental, and their interactions have been considered to play critical roles in triggering and determining the development of diseases [4, 5, 7, 3]. Owing to these challenges blind data mining without specifically modeling possible non-linearity and heterogeneity in disease mechanisms may lead to fruit-
less efforts even with more and “bigger” biomedical data. Previous computational methods for biomarker identification that ignore the potential interactive effects among risk factors and the heterogeneity in different disease stages and subtypes have had limited success in identifying clinically useful information, even though significant resources and efforts have been exerted to collect as “big” and complete as possible omics data, such as in many epigenetic studies [12]. In addition, other characteristics in typical biomedical data, such as high noise, small sample size, extreme dimension of measurements under investigation, and the intrinsic heterogeneity, impose challenges for computational methods for accurate and reproducible biomarker identification [13]. In this dissertation we discuss the specific challenges of computational approaches for biomarker identification in dealing with such complex and heterogeneous diseases and propose novel approaches to address them.

1.2 Problem Statement

In this dissertation we specifically explore biomarker identification for complex diseases in two general situations. In the first problem, biomarkers are identified among high-dimensional profiled measurements by finding a subset that possess high predictive power on the disease outcome (Figure 1 (A)). This problem is typically formulated as a computational problem of feature selection by considering each measurement as a feature in machine learning problems. In the second problem, we look at a heterogeneous population of patients, where we are interested in identifying biomarkers among high-dimensional profiled measurements for different disease subtypes while the actual subtypes or the outcome of interest, that can guide biomarker identification, are unknown. This problem is commonly referred to as tumor stratification for personalized medicine, where unsupervised machine learning methods such as clustering and bi-clustering are usually used for finding subtypes, and subtype specific biomarkers (Figure 1 (B)).

In the following two sections we provide brief background on the above two problems (a thorough literature review is provided in Chapter 2) and point out the main shortcomings of the existing computational methodologies that we address in this dissertation.

1.2.1 Biomarker Identification for Disease Outcome

Feature selection techniques have been used to identify disease biomarkers based on high-throughput biomedical data [2, 3]. Given a set of \( p \) measured components as features for \( n \) samples, \( X_{n \times p} \), together with their phenotype or disease outcome of interest \( y_{n \times 1} \), the goal of feature selection is to identify a subset
A: Biomarker identification for disease outcome (feature selection)

B: Biomarker identification for subtypes of a heterogeneous disease

Figure 1. An illustration of different computational approaches for biomarker identification. The color map is used to show imaginary gene expression levels: red representing high expression, green meaning low expression, and black for normal expression.
from those $p$ features that can help build predictive diagnosis and prognosis models for the disease (Figure 1 (A)). Despite recent advances in feature selection techniques, most of the studies focus on univariate analysis of high-throughput biomedical data [14, 15, 16, 17, 18, 19, 20, 21, 3]. The individual power of the features are estimated and the features are filtered based on this score. Most of the advanced multivariate filter or wrapper feature selection techniques are ineffective in dealing with high-throughput biomedical which can be explained by:

- The high-dimensionality of biomedical data makes the multivariate analysis almost impossible due to the exponential growth of the number of subsets that need to be evaluated.
- The small sample size and noise in the data makes the statistics based on multiple features less reliable.
- The domain experts in biology prefer the analysis based on individual features because it is more intuitive and easier to understand and interpret.
- And more importantly, exhaustive search approaches tend to find spurious features due to overfitting mainly because no prior assumption is made about the underlying model.

However, these analyses focusing on individual or marginal effects may not be sufficient as stated both in Genome Wide Association Studies (GWAS) [16, 18, 17] and in gene expression analysis [4, 5, 7] since two features with no significant individual effects can be highly “synergistic” and informative of the outcome when considered together [22] due to non-linearity as shown in Figure 2. Identifying such interactive effects among features not only helps identify more accurate biomarkers for outcome prediction, but also helps reveal functional interactions among cellular components which are specifically related to the phenotype or outcome of interest [23, 22]. To overcome this problem with univariate analysis while avoiding the problems faced by previous multivariate feature selection, we can make a simplified assumption about the underlying model. Based on the nature of biological processes in cells we can assume that the components in cells (based on which the features are measured) regulate the biological processes either individually or by pairwise interactions with other components. This will enable more systematic analysis of biomedical data related to complex disease and more biologically interpretable feature selection as opposed to exhaustive search approaches. Feature selection based on this framework can be accomplished by measuring the individual effects and pairwise interactive effects between features and then filtering the features based on both scores. To find biomarkers considering interactive effects, most of the existing approaches first focus on detecting or measuring interactive effects among all the features and then use simple greedy approaches.
Figure 2. An illustration of high interactive effect between two features $x_1$ and $x_2$ with respect to the outcome $y$. The features $x_1$ and $x_2$ are not able to discriminate the outcome $y$ individually.

based on the ranked list of the pairs to select biomarkers [5, 4]. The performance of these methods is highly dependent on the accurate and robust measurement of interactive effects among features.

1.2.2 Biomarker Identification for Subtypes of Heterogeneous Diseases

There has been increasing evidence that many types of cancer are highly heterogeneous in the sense that associated somatic genes or other molecules may differ in different patients which usually results in different disease subtypes with varying behavior, including different survival time, different responses to drugs, and different recurrence rates [1, 8, 9, 10, 11, 6]. One of the important problems in cancer informatics is tumor stratification, where the goal is to find the tumor subtypes in a heterogeneous population of samples. In addition to finding relevant tumor subtypes, one other important problem is to identify the biomarkers (genes for example) related to each subtype for the purpose of personalized treatment and prognosis. Unlike supervised learning for previously proposed biomarker identification with respect to disease outcome, here the aim is to find biomarkers that better discriminate the disease subtypes in an unsupervised manner where the outcome or subtypes are not known.

There are two general computational approaches for identifying features (genes) associated with each subtype as biomarkers: (1) Clustering methods can be used to first stratify the tumor samples into subtypes and then for each subtype the genes that show significantly different values compared to other subtypes can be selected as the biomarkers. Here, identifying accurate subtypes is a crucial first step towards identifying
meaningful personalized biomarkers associated with heterogeneous diseases. (2) Bi-clustering approaches can be used to simultaneously find the subtypes and their corresponding biomarkers. The benefit of using bi-clustering approaches is that the clustering methods may fail to find the correct clusters as they are utilizing all the genes for clustering, not specifically the relevant genes; and they may also fail to find the relevant genes after clustering as the performance depends on those clusters which may not be correct in the first place.

Using “big” data provided by high-throughput omics profiling techniques, researchers have used standard unsupervised clustering [1, 8, 9, 10, 11, 6] and bi-clustering approaches [24, 25, 26] to find subtypes with significantly different survival rates, tumor stages or grades, histological types, and drug responses mostly based on mRNA expression profiles [1, 8, 9, 10, 11]. However, these methods have limited performance because usually the datasets contain a very small number of samples and the gene expression measurements are often noisy. A relatively more promising and less noisy source of biomedical data is the somatic mutation profiles [1], where each gene is represented by a binary vector of mutation profiles. However, the problem with mutation profiles is that the number of mutations for each tumor is usually very small compared to the total number of genes under investigation and the tumors rarely share the same mutations, due to the extreme heterogeneity of the disease. It is in fact very common for two clinically identical tumors to share only one mutation [1, 27]. Owing to these properties it is very difficult to compare the tumors by calculating the distance or similarity between mutation profiles [6]. Due to these critical issues traditional clustering approaches usually fail in achieving meaningful subtypes by simply clustering the mutation profiles. Furthermore, in traditional bi-clustering [24, 25, 26] the goal is to simultaneously cluster both rows (tumors) and columns (genes) of data matrix by finding sub-matrices (or so called bi-clusters) in which the subset of tumors behave similarly over the subset of genes. However, for binary somatic mutation matrix data (or the smoothed somatic mutation data in which the mutation signals are smoothed), we are only interested in bi-clusters (or sub-matrices) containing mostly 1s (or large values). In other words, although a sub-matrix containing all zeros (or small values) is considered as a perfect bi-cluster in bi-clustering approaches, it does not indicate a meaningful subtype.

1.3 Our Contribution

In this dissertation we focus on addressing the main shortcomings of the existing computational methodologies for the aforementioned two main problems.
For the first problem, we focus on the problem of estimating the interactive effect between features, specially focusing on finding methods for better estimation for continuous random variables (numerical features) when we have a limited number of samples. Based on the estimated interactive effects, we further propose a novel feature ranking approach based on a constructed synergy network [22] that takes into account interactive effects among features in addition to their individual effects.

For the second problem, we first focus on clustering somatic mutations and explore new ways of incorporating biological knowledge to better estimate the distance between binary somatic mutation profiles. We show that information from GO annotations (please refer to Chapter 5) and other curated gene-sets such as pathway databases can help achieve better stratification for Breast Cancer. The benefit of this approach, as opposed to using gene-gene interaction network, is that we directly use separate gene-sets or modules instead of using the network as whole. We then focus on simultaneous identification of both subtypes and their corresponding biomarkers based on somatic mutation profiles, by solving a novel bi-clique finding problem in a bipartite graph.

1.4 Organization

The rest of this dissertation is organized as follows. In Chapter 2, we provide technical background on recent related computational methods used for biomarker identification and tumor stratification in the field of bioinformatics. Then in next two chapters we focus on the problem of considering interactive effects in biomarker identification. In Chapter 3, we first review previously proposed methods for interactive effect estimation. We then propose new approaches, along with a discussion of the problems and challenges in previous methods and explain how these new approaches can address them. We then evaluate the performance of the interactive effect estimation methods on simulation experiments and compare their power in detecting actual interactions among simulated random variables (as features) with data of different sample sizes from small to large. The results show that our proposed new methods can detect interactive effects more accurately than other methods with both large and small sample sizes. Based on these estimated interactive effects, we further propose a novel feature ranking in Chapter 4, that takes into account interactive effects among features in addition to their individual effects to rank the features. We first test the performance of our network-based feature ranking method based on simulated datasets. We then show that it can help in finding more accurate biomarkers for prediction of breast cancer metastasis and type 1 diabetes compared to traditional individual-based ranking approaches. We further show that the identified biomarkers, by considering
interactive effects, contain genes that are involved in important pathways related to cancer metastasis. Then in two chapters, we focus on biomarker identification for heterogeneous diseases. In Chapter 5, we focus on separate subtype identification and biomarker identification approaches. We propose novel methods for calculating the distance between somatic mutation profiles. We propose the use of prior biological knowledge in pathway databases and GO Annotations to address the non-linearity in disease and heterogeneity of the mutation profiles which can help achieve more accurate clustering for somatic mutation profiles. Furthermore, we discuss the problems in calculating the distance between sparse binary vectors and suggest a method to alleviate that. We show that our approach based on pathways and GO terms can perform better than the traditional clustering methods as well as the recent state-of-the-art NBS method in identifying actual subtypes of Breast Cancer using somatic mutation profiles. In Chapter 6, we propose a novel bi-clique finding method for simultaneous tumor stratification and biomarker identification based on somatic mutation profiles. We propose a novel formulation for bi-clique finding which can help take into account prior biological knowledge in the gene-gene interaction network to alleviate the noise and small sample size problem in biomedical data. And finally, in Chapter 7 we summarize the dissertation and provide possible new directions for research in the computational biomarker identification area.
Chapter 2
Literature Review

In this chapter, we provide a summary of related computational work in biomarker identification. We first discuss the recent feature selection approaches used for biomarker identification based on high-throughput biomedical data. We then review recent clustering and bi-clustering methods used for subtype and biomarker identification for heterogeneous diseases.

2.1 Feature Selection for High Throughput Biomedical Data

Feature selection has been studied in data mining and machine learning literature [28, 29, 3]. The feature selection techniques are usually organized in two main categories: filter methods and wrapper methods. In filtering techniques, the relevance of the features are measured by looking at the statistical association between the features (or a subset of features) and the outcome. A score is calculated for each feature or subset and then the high scoring features or subsets of features are selected. Filter techniques are usually simple and fast, as they are independent of the classification algorithm. However, most of the filtering methods are univariate and focus only on the individual power of the features and ignore the possible interactions between features that might be informative of the outcome. To overcome this problem, a number of multivariate analysis methods have been proposed to try to incorporate the interaction between features such as Correlation-based feature selection [28] and Markov blanket filter [29]. The wrapper techniques [30] are usually multivariate and search for a subset of features that provides the best classification accuracy for a given classifier. Several subsets of features are generated and evaluated by learning a classifier and estimating the empirical error rate. The wrapper methods take into account the specific classifier for feature selection and therefore provide more relevant features to achieve more accurate classifiers. However, they are very computationally intensive and they suffer from overfitting.

Despite recent advances in feature selection techniques, most of the studies in biomarker identification focus on univariate analysis of high-throughput biomedical data [14, 15, 16, 17, 18, 19, 20, 21, 3]. The indi-
individual power of the features are estimated and the features are filtered based on this score. Most of the advanced multivariate feature selection techniques are ineffective in dealing with high-throughput biomedical data due to high-dimensionality of biomedical data, small sample size problem, lack of biological interpretability, and no effective prior assumption about the underlying model, as discussed in Section 1.2.1 of Chapter 1. However, these analyses focusing on the individual effects may not be sufficient since two features with no significant individual effects can be highly “synergistic” and informative of the outcome when considered together [22] (Figure 2 of Chapter 1). For example, in [31], a study was done to discover multivariate logical predictive relations among gene expressions. They have found several XOR relationships similar to what is shown in Figure 2. For example, they showed that if both Interleukin18 and MRC1 genes are up-regulated or if both are down-regulated, then the EHHADH gene is down-regulated.

As explained in Section 1.2.1 of Chapter 1, to overcome this problem with univariate analysis while avoiding the problems faced by previous multivariate feature selection, we can make a simplified assumption about the underlying model. Based on the nature of biological processes in cells we can assume that the components in a cell (based on which the features are measured) regulate the biological processes either individually or by pairwise interactions with other components in the cell. Therefore, the cell can be viewed as individual components and pairwise interaction among them. This will enable a more systematic analysis of biomedical data related to complex disease and more biologically interpretable feature selection as opposed to exhaustive search approaches. Feature selection based on this framework can be accomplished by measuring the individual effect and pairwise interactive effects between features and filtering the features based on both scores.

Several methods have been proposed both in GWAS [16, 18, 17] and other -omic data analyses [4, 5, 7] to take into account interactive effects among features. However, the performance of these methods highly depends on accurate and reliable interactive effect estimation, which is very difficult to achieve compared to individual effect, due to the small sample size problem in biomedical data. Recently, researchers have proposed different statistical and computational methods for measuring the interactive effect [32, 16, 17, 18, 33, 34, 23, 22, 4, 5, 35]. However, most of these proposed methods are designed for quantized or discrete random variables (nominal features) [16, 17, 18, 33, 32, 34, 35, 23] or have made simplified data distribution assumptions [32].

For better readability of the dissertation, we leave a more detailed literature review on methods for estimating and incorporating interactive effects to Chapter 3, where we propose new methods for better interactive...
effect estimation and compare the performance of all methods. We further propose a novel feature ranking in Chapter 4 that integrates both individual and interactive effects to sort the features.

2.2 Clustering Methods for Tumor Stratification

Using the data provided by high-throughput omics profiling techniques, researchers have used standard unsupervised clustering such as hierarchical clustering and Non-negative Matrix Factorization (NMF) to cluster the samples as subtypes [1, 8, 9, 10, 11, 6] to find subtypes with significantly different survival rates, tumor stages or grades, histological types, and drug responses mostly based on mRNA expression profiles [1, 8, 9, 10, 11].

In this section we summarize clustering approaches recently used for identifying subtypes in complex disease based on high-throughput biomedical data.

The traditional k-means clustering [36] is used by [37] to stratify tumor gene expression profiles. For a specified number of clusters $K$, k-means clusters the data into $K$ groups such that following objective function is minimized

$$E = \sum_{i=1}^{K} \sum_{O \in C_i} |O - \mu_i|^2$$

where $O$ is a data point and $C_i$ is the set of all data points categorized in the $i$th group and $\mu_i$ is the cluster centroid or the mean of the data points in $C_i$. To optimize the above objective function, k-means uses a two step procedure: first each data point is assigned to the closest cluster based on the distance from the centroids; in the second step the cluster centroids are updated based on the new data point assignments.

Hierarchical clustering is another traditional clustering method widely used for tumor stratification [8, 9, 10, 11]. Based on the distances between data points and between clusters, the hierarchical clustering generates a nested tree structure called dendrogram. The dendogram then can be cut at any level to obtaining any specified number of clusters. There are two ways of generating the tree: bottom-up (Agglomerative) approach in which initially each data point is considered as a separate cluster and, for example, in each further step the two closest clusters are merged until only one cluster remains; top-down (divisive) approach where initially all data points are put in one cluster and in each further step the large clusters are split, based on a certain criteria, until only singleton clusters remain. One widely used variation for clustering biomedical data is called UPGMA [38] where the distance between two clusters is calculated as the average distance between all pair of points across the two clusters.
Other model based clustering methods are also frequently used for clustering biomedical data [39]. In latent model based clustering, the data points are assumed to come from a mixture of distributions with each component representing one cluster. Assuming the probability density function for the $i$th component $C_i$ is $f(x, \theta_i)$, the mixture model is

$$P(x = x_r) = \sum_{i=1}^{K} \gamma_{r,i} f(x = x_r, \theta_i)$$

in which $K$ is the number of components and the hidden parameter $\gamma_{ki}$ is the mixture parameter indicating the probability of the $r$th data point coming from the $i$th component. Due to the hidden variable involved in such mixture models, a two step algorithm, Expectation Maximization (EM) can be used to find the maximum likelihood estimation of the parameters. One of the variations of these model based clustering is called the Gaussian Mixture Model in which $f(x)$ is assumed to follow the Gaussian distribution and is widely used for clustering gene expression profiles.

To achieve more robust clustering, consensus clustering [40] techniques are also extensively used for tumor stratification based on biomedical data [41, 1, 42, 43]. Consensus clustering is usually used to combine the results of several clustering methods based on different types of biomedical data [44] or to aggregate the clustering results based on several runs of clustering on a single dataset [6]. Several approaches has been proposed for combining the multiple clustering results [40]. For example, [40], proposed an approach to combine the clustering results of multiple runs of the same clustering algorithm on different sub-samplings of the dataset. They first generate a co-clustering matrix $CC_{n\times n}$ ($n$ is the number of samples), in which $CC_{i,j}$ is equal to the number of times $i$th and $j$th datapoints are clustered together divided by the number of times both appeared in a sub-sampled dataset. The resulting consensus matrix provides a similarity measure between datapoints which can be used in conjunction with agglomerative hierarchical clustering to generate a dendogram tree.

Matrix factorization techniques which are usually used to represent data in a lower dimension, can also be used for clustering proposes. Among them Non-negative Matrix Factorization (NMF) [45] is widely used for clustering gene expression profiles, however, a preprocessing step is always required to create a non-negative matrix from gene expression data. Given the non-negative representation of the data $X_{n\times m}$ and and a pre-specified number of clusters $K$, the NMF algorithm finds two non-negative matrices $W_{m\times K}$ and $H_{n\times K}$ that minimize the residual error $||X - HW^T||_F^2$. The columns of $W$ are usually called the meta-genes (a combination of $m$ genes) representing the clusters and each row of $H$ contains $K$ values indicating the probability of the data-points belonging to each one of the $K$ clusters. The maximum value in the $i$th
row of $H$ indicates the cluster to which the $i$th element is assigned. To solve this optimization problem multiplicative updating rules are used that converge to the local minimum of the above optimization problem [45]:

$$H_{ik} \leftarrow H_{ik} \frac{(XW)_{ik}}{(HW^T W)_{ik}}$$

$$W_{ik} \leftarrow W_{ik} \frac{(X^T H)_{ik}}{(WH^T H)_{ik}}$$

Most of the traditional clustering approaches are suitable for gene expression, however, as previously shown [6], these methods have limited performance because usually the datasets contain a very small number of samples and the measurements are often noisy. A relatively more promising and less noisy source of biomedical data is somatic mutation profiles, where each tumor is represented by a binary vector of mutation profiles. However, clustering binary mutation profiles is difficult due to their extreme sparseness and heterogeneity. To address this issue [6] proposed a network-based stratification (NBS) framework that takes advantage of two sources of information to find subtypes: 0-1 matrix $A_{n \times m}$ containing the somatic mutation profiles of $m$ genes for $n$ patients; and the adjacency matrix $M_{m \times m}$ of a gene-gene interaction network. They first smooth the mutations in $A$ over the network $M$ using a network propagation technique introduced in [46] using the following recursive formula:

$$A_{t+1} = \alpha A_t M' + (1 - \alpha) A_0$$

(2.1)

where $A_0$ is equal to somatic mutation matrix $A$, $M'$ is obtained by dividing each column of $M$ by the summation of elements in that column (degree normalized adjacency matrix). The tuning parameter $\alpha$ controls the distance in which mutation signal is allowed to diffuse into the network. The propagation spreads the influence of the mutation to the neighboring genes in the network. The new smoothed somatic mutation matrix is then decomposed using Network-regularized NMF originally proposed by [47] by minimizing the following objective function:

$$\min_{W,H \geq 0} ||A_{\infty} - HW^T||^2 + \text{trace}(WLW^t)$$

Matrices $H_{n \times K}$ and $W_{m \times K}$ form a decomposition of the network smoothed matrix $A_{\infty}^{n \times m}$, where $K$ is the number of desired clusters or subtypes, columns of $W$ are the basis vectors or “metagenes”, and the rows
of $H$ are the weighted cluster assignments to each one of $K$ clusters where the largest value on each row indicates the assigned cluster to the corresponding patient tumor. The matrix $L$ is the Laplacian matrix computed based on the adjacency graph $M$. By minimizing the term $\text{trace}(WLW^t)$, the basis vectors in $W$ are constrained to respect local network neighborhoods, i.e., connected genes in the network tend to have similar corresponding values in basis vectors. Similar to NMF, following multiplicative updating rules derived for Network-regularized NMF [47]:

$$
H_{ik} \leftarrow H_{ik} \frac{(XW)_{ik}}{(HW^tW)_{ik}}
$$

$$
W_{ik} \leftarrow W_{ik} \frac{(X^tH+MW)_{ik}}{(WH^tH+DW)_{ik}}
$$

in which, $D$ is the diagonal matrix of the degrees of nodes in the network corresponding to $M$. The above multiplicative updating rules are proved to converge to local minimum of the above optimization problem [47].

### 2.3 Bi-clustering Methods for Simultaneous Tumor Stratification and Biomarker Identification

One traditional approach for simultaneous subtype and biomarker identification is bi-clustering [24, 25, 26] where the goal is to simultaneously cluster both rows (tumors) and columns (genes) of a data matrix by finding sub-matrices (or so called bi-clusters) in which the subset of tumors behave similarly over the subset of genes. Based on the application of interest several definitions exist for bi-clusters. For example, constant bi-clusters are the sub-matrices of data with almost the same values. Other bi-clustering approaches may look for sub-matrices with constant values on rows or on columns. Several surveys on different types of bi-clusters and the algorithms to find them are provided in [24, 48, 49]. In traditional bi-clustering approaches, usually a measure of quality for bi-clusters, e.g. mean squared residue (MSR) or scalling mean squared residue (SMSR), are defined and heuristic algorithms are used to find bi-clusters with high quality [50, 51]. Here we review recent more sophisticated bi-clustering approaches used for analyzing biomedical data [52, 53, 54, 55].

Several probabilistic modeling approaches are used for bi-clustering [52, 53]. In [52] they proposed a probabilistic modeling approach called Plaid Models (PM) for multivariate analysis of gene expression data. They assume that the data matrix is an image generated by different layers representing the underlying bi-clusters with different colors. Based on this, they assume following model for the gene expression values
in the gene expression data matrix $Y$:

$$Y_{ij} = \mu_0 + \sum_{k=0}^{K} \mu_k \delta_{ik} \kappa_{jk}$$

in which $K$ is the number of desired bi-clusters, $\mu_0$ is the background color, $\mu_k$ is the main effect (color) of the $k$th bi-cluster, $\delta_{ik}$ and $\kappa_{jk}$ are binary values indicating whether the $i$th gene and $j$th sample belong to cluster. This model can be used to find constant bi-clusters. To further improve the model, [53] proposed a Bayesian approach, called the Bayesian Biclustering model (BBC), that allows multiple bi-clusters without overlapping samples or overlapping genes. They assume the following model:

$$Y_{ij} = \sum_{k=1}^{K} ((\mu_k + \alpha_{ik} + \beta_{jk} + \epsilon_{ijk}) \delta_{ik} \kappa_{jk}) + e_{ij}(1 - \sum_{k=1}^{K} \delta_{ik} \kappa_{jk}),$$

where $\alpha_{ik}$ and $\beta_{jk}$ are the effect of sample $i$ and gene $j$ which are added to allow finding bi-clusters with constant rows and constant columns, $\epsilon_{ijk}$ is the noise term for cluster $k$, and $e_{ij}$ is used to model the data points that do not belong to any clusters. The priors of the indicators $\kappa$ and $\delta$ are set as follows to avoid overlapping samples in bi-clusters:

$$\kappa_{ij} \sim Bernoulli(q_k)$$

$$P(\delta_{ij} = 1, \delta_{il} = 0, l \neq k) = p_k$$

$$P(\delta_{il} = 0, l = 1, 2, ..., K) = p_0 = 1 - \sum_{k=1}^{K} p_k$$

where $p_k$ and $q_k$ are constant. The following a priori assumptions are made:

$$\mu_k \sim N(0, \sigma_{\mu_k}^2)$$

$$\alpha_{ik} | \delta_{ik} = 1 \sim N(0, \sigma_{\alpha_k}^2)$$

$$\beta_{jk} | \kappa_{jk} = 1 \sim N(0, \sigma_{\beta_k}^2)$$

$$\epsilon_{ijk} \sim N(0, \sigma_{\epsilon_k}^2)$$

$$e_{ij} \sim N(0, \sigma_e^2)$$

(2.2)
where all the $\sigma^2$ are assumed to follow an inverse Gamma distribution. To learn the model they use a Gibbs sampling method. Further improvement over the above method is provided in [55]. They provide a Bayesian Framework for bi-clustering called the Penalized Plaid Model (PPM), in which they propose a method to find the optimal number of bi-clusters and address overlapping. In this model, the hard-EM algorithm has been used for parameter estimation. Another probabilistic bi-clustering approach is cMonkey [54]. Recently, [25] used cMonkey for stratification of breast cancer tumors, which shows promising results in identifying phenotypically heterogeneous subtypes together with their corresponding biomarker or feature set.

Other approaches for bi-clustering propose the use of Matrix Factorization methods to find the bi-clusters. One famous algorithm called Non-smooth Non-negative Matrix Factorization (nsNMF) [56] is used in [57] to analyze gene expression profiles. This algorithm is motivated by the NMF clustering algorithm explained previously in this section but provides very sparse matrices and therefore enables gene selection for identified subtypes. They propose to approximate data matrix $X_{n,m}$ by $WSH$ where $W$ and $H$ are the same as in the original NMF method. The positive symmetric matrix $S_{K \times K}$, is called the smoothing matrix defined as

$$S = (1 - \theta)I + \frac{\theta}{K}11^T,$$

where $I_{K \times K}$ is the identity matrix and $1$ is a vector of length $K$ with all the elements equal to 1. The parameter $\theta$ is the ”smoothness” parameter that controls the sparseness of the of the final solution. To find the solution, they propose an algorithm to minimize the divergence of $X$ from $WSH$ defined as follows:

$$D(X, WSH) = \sum_{i=1}^{n} \sum_{j=1}^{m} \left( X_{ij} \ln \frac{X_{ij}}{(WSH)_{ij}} - X_{ij} + (WSH)_{ij} \right).$$

They further provide an iterative algorithm based on the following rules to find $W$ and $H$ that minimizes the divergence.

$$H_{ik} \leftarrow H_{ik} \frac{\sum_{j=1}^{n} ((WS)_{jk}(X)_{ij}) / \sum_{q=1}^{K} ((WS)_{jq}(H)_{qj})}{\sum_{j=1}^{n} (WS)_{jk}}$$

$$W_{ik} \leftarrow W_{ik} \frac{\sum_{j=1}^{m} ((HS)_{kj}(X)_{ij}) / \sum_{q=1}^{K} ((W)_{jq}(HS)_{qj})}{\sum_{j=1}^{n} (HS)_{jk}}$$

$$W_{ik} \leftarrow \frac{W_{ik}}{\sum_{j=1}^{n} (W)_{jk}}$$

They further prove that the above algorithm converges to a local minimum of $D(X, WSH)$. 16
All of the above discussed approaches for bi-clustering are designed for analysis of gene expression profiles which are real valued vectors. As discussed in the introduction, the tumor stratification problem using somatic mutation data can be better represented as finding densely connected sub-graphs in a bipartite graph, which enables both subtype identification and personalized biomarker identification for each subtype. More formally, given the somatic mutation data matrix $A_{n \times p}$, which contains the smoothed somatic mutation profiles of $p$ genes for $n$ tumors, we can construct its corresponding bipartite graph denoted by $G = (U, V, E)$, where the node set $U = \{u_1, \ldots, u_n\}$ is in one part of the graph representing the tumor samples, and the node set $V = \{v_1, \ldots, v_m\}$ is in the other part of the graph representing the genes. Node $v_i$ is connected to $u_j$ with weight $A_{i,j}$ if $A_{i,j} > 0$.

Community detection and graph partitioning approaches have been studied for finding densely connected subgraphs [58]. In community detection or graph partitioning, the nodes in a given graph are clustered in an arbitrary number of partitions, modules, or communities such that the clusters are highly connected internally while there are a small number of connections across different clusters. Although these methods can be used to find clusters in any graph, recently, researchers have proposed alternative methods that perform better for bipartite graphs [59, 60, 61]. In the following, we discuss two methods for bipartite graph clustering proposed by [61, 59] that can be used for tumor stratification.

One approach for finding densely connected subgraphs is a graph partitioning approach in which the aim is to divide the graph nodes into two groups such that the number of edges between groups or the cut is minimum [60, 61]. In [61] they propose a method for partitioning the bipartite graph by minimizing the ratio-cut between the partitions. For a graph $G = (V, E)(|V| = n)$, the ratio-cut between two partitions $V_1$ and $V_2$ ($V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$) is defined as:

$$\text{Ratio-cut} = \frac{\text{cut}(V_1, V_2)}{|V_1|} + \frac{\text{cut}(V_1, V_2)}{|V_2|}.$$

They show that solving the following optimization problem finds the partitions that minimize the Ratio-cut:

$$\min_{q \neq 0} \frac{q^T L q}{q^T D q} \text{ subject to } q^T D e = 0$$

where $q$ of size $n$ is a $\{-1, 1\}$ assignment vector ($q_i = 1$ means $i$th node in the graph is assigned to first partition and $q_i = -1$ means $i$th node in the graph is assigned to second partition), $D$ is a diagonal matrix of node degrees, $L$ is the Laplacian matrix of $G$, and $e = [1, \ldots, 1]^T$. This problem is also NP-complete so
they propose a spectral graph algorithm in which they relax the values of the elements in \( q \) to take any real number. They prove that a solution \( q^* \) of the relaxed problem is equal to the eigenvector corresponding to the second smallest eigenvalue of the following generalized eigen system problem:

\[
Lz = \lambda Dz
\]

To find larger numbers of partitions they suggest either calling the spectral partitioning problem recursively or using other eigenvectors of the above problem to form a matrix \( Z \) and perform k-means clustering algorithm. For the case of a bipartite graph with adjacency matrix \( A_{n \times m} \), they define \( A_n = D_1^{\frac{3}{2}} A D_2^{\frac{1}{2}} \) in which \( D_1 \) and \( D_2 \) are diagonal matrices of degrees for the left part of the bipartite graph and right part of the bipartite graph respectively. They then prove that the solution for the bipartite partitioning problem are the left and right singular vectors corresponding to the the second smallest singular values of \( A_n \).

Another approach for community detection was proposed by [62], in which they used the leading eigenvectors of a Modularity matrix of the graph to find the communities. This method is further extended for bipartite graphs in [59], in which they define a modularity matrix suitable for bipartite graphs and use the left and right singular vectors of the new modularity matrix to find the clusters. Given the adjacency matrix \( A_{n \times m} \) of a bipartite graph, they define the modularity matrix \( B \) as \( B = A - P \) where \( P_{ij} = \frac{k_i d_j}{m} \), \( k_i \) is the degree of the \( i \)th node in \( U \), and \( d_j \) is the degree of the \( j \)th node in \( V \). The module or community assignment of the nodes in \( U \) and \( V \) can be represented by assignment matrices \( S_{n, K} \) and \( G_{m, K} \) where \( K \) is the number of communities or modules, \( S_{i,k} = 1 \) if the \( i \)th node in \( U \) belongs to \( k \)th module, and \( G_{j,k} = 1 \) if the \( j \)th node in \( V \) belongs to \( k \)th module. The modularity \( Q \) of a given assignments \( S \) and \( G \) is defined as

\[
Q = \frac{1}{m} \text{trace}(S^T B G)
\]

Now the problem is to find the assignment matrices \( S^* \) and \( G^* \) that maximize the modularity \( Q \) which is computationally intractable. Instead, they use singular vector decomposition to find the left and right leading singular vectors of \( B \) to obtain approximate solutions for the corresponding relaxed problems.

Neither of the above graph partitioning and community detection methods have been tested for the stratification problem, but it has shown promising results in finding community groups in bipartite social networks [59] and document clustering [61]. However, these methods may not be appropriate for stratification
problems as they attempt to cluster all the genes, while in stratification we are interested in finding a few important genes or biomarkers for each identified subtype and the rest of the genes could be put aside.

The most dense subgraphs in bipartite graphs are bi-cliques. The problem of finding and enumerating all edge maximal bi-cliques has been studied in literature [63, 64, 65, 66] among which [64] proposed a novel approach to efficiently find large bi-cliques. Let $A_{n \times m}$ be the adjacency matrix of the bipartite graph $G$, a maximal bi-clique can be computed by finding solution of the following optimization problem:

$$\max_{s, g} s^T A g$$

s.t. $s_i > 0$ for $i \in \{1..n\}$

$g_i > 0$ for $i \in \{1..m\}$

$$\sum_{i=1}^{n} s_i^\alpha = 1$$

$$\sum_{i=1}^{m} g_i^\beta = 1$$

Given a solution $(s^*, g^*)$ of the above optimization problem, the nodes in left part and right part of the maximal bi-clique are indicated by non-zero elements in $s^*$ and $g^*$ respectively. The parameters $1 < \alpha \leq 2$ and $1 < \beta \leq 2$ favor different shapes of cliques: If for example $\alpha > \beta$ then it tends to find the cliques with a larger number of nodes from the left side than from the right side. Also, setting $\alpha, \beta$ to a value close to 1 (e.g. 1.05 or 1.1) helps find very tight (exact) cliques and setting $\alpha, \beta$ to larger values (e.g. 1.2) could find loose cliques with some missing edges. The latter is more useful in most of the applications such as tumor stratification since densely connected subgraphs are more realistic. To solve the above optimization problem, they use following multiplicative update rules:

$$s_i^{(t+1)} = (s_i^{(t)} \frac{(Ag^{(t)})_i}{s^{(t)T}Ag^{(t)}})^{1/\alpha}$$

$$g_i^{(t+1)} = (g_i^{(t)} \frac{(A^TS^{(t)})_i}{s^{(t)T}Ag^{(t)}})^{1/\beta}$$

Starting from a random feasible $s_0$ and $g_0$ they prove that the multiplicative update rules converge to a point satisfying KKT conditions for local maxima. To find multiple bi-cliques they suggest deleting the first bi-clique from the matrix $A$ and performing the above algorithm to find the next bi-clique. This can be repeated until we find a desired number of bi-cliques. Unlike graph partitioning or community detection approaches, bi-clique finding is able to set aside irrelevant genes as it does not attempt to cluster all the
genes. It tends to find highly densely connected subgraphs which usually contain small numbers of genes. In Chapter 6, we will show how this method can be extended to incorporate knowledge from gene-gene interaction networks for better simultaneous tumor stratification and biomarker identification.
Chapter 3
Detecting Pairwise Interactive Effects For Continuous Random Variables

In this chapter, we first review the previously proposed methods for estimating the interactive effects between features focusing on the methods dealing with continuous random variables (In this chapter we assume that the features are continuous random variables and we will refer to them as variables). We then propose new methods for better estimation of interactive effects especially for the small sample size scenario. We represent biomedical data by $X$ which is a $n$-by-$p$ matrix, where $p$ is the number of candidate variables and $n$ is the number of observations or samples. Typically in biomedical data we have: $p \gg n$. Throughout this chapter we investigate methods for measuring the interactive effect between two continuous random variables $x_i$ and $x_j$ with respect to the outcome variable $y$, which, for example, can be binary denoting whether the corresponding samples come from “healthy” or “diseased” subjects. We review the previously proposed methods, especially focusing on the ones studying continuous random variables; and then further propose a set of new methods to better detect interactive effects. We have categorized existing methods together with our newly proposed methods into three categories: (1) Information theoretic synergy based methods [32, 22, 23], in which marginal and conditional probability distributions of $x_i$, $x_j$, and $y$ are estimated from data to compute the information theoretic synergy as interactive effect. In this category, the reliability of interactive effect estimation depends on how well the involved probability distributions are estimated; (2) Classification based methods [34, 5, 4], in which first a classification model is learned from the data and then the interactive effect is measured by some parameters in the model, such as coefficients in Logistic Regression, relative values of variables in Relative Expression Analysis (REA), or estimated synergistic predictive power of $x_i$ and $x_j$ to $y$ using classification accuracy; and (3) Association based methods, where the interactive effect is measured by the increase in association between $x_i$ and $x_j$ after observing the outcome $y$. In Figure 3, we have laid out a hierarchy of the approaches studied in this chapter that include the existing methods as well as our newly proposed methods.

We evaluate the performance of all the methods based on simulations and compare their power in detecting underlying interactions among simulated random variables with data of different sample sizes. The
Figure 3. All the methods studied in this chapter with their relationships illustrated in a tree graph.

Simulation results show that our proposed methods outperform many existing methods for interactive effect estimation in general.

3.1 Existing Methods

Here we review the existing methods in the literature for measuring interactive effects between two continuous random variables $x_i$ and $x_j$ on a categorical outcome $y$.

3.1.1 Information Theoretic Measures

One of information theoretic measures is based on the definition of synergy $\text{Syn}(x_i, x_j; y)$ by Anastassiou et al. [23], which measures the portion of the information gain from the pair of covariates $(x_i, x_j)$ subtracting their individual information gain [67]:

$$\text{Syn}(x_i, x_j; y) = I(x_i, x_j; y) - I(x_i; y) - I(x_j; y),$$

(3.1)

where $I(x; y) = H(y) - H(y|x) = H(x) - H(x|y)$ can be considered as the amount of the information gain measured in the reduction of the entropy of $y$ after knowing the value of either individual or multiple variables. Hence, it is essential to accurately estimate $I(x; y)$, which can be challenging under the small sample size scenario with $p \gg n$. Further expanding the formula in (3.1), we can rewrite the synergy based
on conditional entropies: \( \text{Syn}(x_i, x_j; y) = H(x_i, x_j) - H(x_i, x_j|y) + H(x_i|y) - H(x_i) + H(x_j|y) - H(x_j) = H(y|x_i) + H(y|x_j) - H(y,x_i,x_j) - H(y) \). For quantized random variables \( x \) and \( y \), we can have relatively reliable maximum likelihood estimates of conditional entropies \( H(y|x) = \sum_\mu P(x = \mu) \sum_\nu -P(y = \nu|x = \mu) \log(P(y = \nu|x = \mu)) \) if the sample size is reasonably large \([23, 35, 33, 32]\).

When \( x \) is continuous and \( y \) is discrete, we can compute conditional entropies \( H(x|y) = \sum_\nu P(y = \nu) \int_{-\infty}^{+\infty} [-P(x = \mu|y = \nu) \log(P(x = \mu|y = \nu))] d\mu; \) and \( H(y|x) = \int_{-\infty}^{+\infty} P(x = \mu) [\sum_\nu -P(y = \nu|x = \mu) \log(P(y = \nu|x = \mu))] d\mu. \) Clearly, in this case, it is more difficult to have reliable conditional entropy estimates, especially with the limited sample size, which is often the case in biomedical research.

We now first review the existing methods that estimate conditional entropies for continuous input variables with binary outcome.

### 3.1.1.1 Quantization Methods

The first intuitive approach for continuous random variables is to quantize the data and estimate the conditional entropies as quantized random variables. Generally, the observed values of continuous random variables can be categorized into several groups for quantization. The obtained quantized variable \( x_c \) will be used to compute the conditional entropy \( H(y|x_c) \) as an approximate estimate for \( H(y|x) \) and consequently to estimate \( \text{Syn}(x_i, x_j; y) \).

One simple quantization scheme is to take the corresponding sample mean estimates \( m_i = \frac{1}{n} \sum_{\ell=1}^{n} x_{i\ell} \) as the threshold to dichotomize the original measurements as Bernoulli random variables. There are other quantization schema. For example, instead of using mean values, we can quantize by quantiles. Also, to estimate \( H(y|x_i, x_j) \), the quantization of the pair of variables can be done simultaneously. For example, one can categorize the observed values of \( x_i \) and \( x_j \) into two groups by using the line \( x_i - x_j = 0 \) as a separating line which categorizes the data points in the \((x_i,x_j)\) plane into two groups: The sample points with \( x_i \geq x_j \) and the points with \( x_i < x_j \), which indeed is the essence of Relative Expression Analysis (REA) \([5]\).

The main problem with those quantization methods is that the appropriate thresholds or separating criteria for quantization are generally unknown. More complicated clustering methods have been proposed for automatic quantization. For example, Anastassiou et al. \([22]\) have proposed an UPGMA hierarchical clustering method to quantize continuous variables. By taking clustering results at different levels of the hierarchy, the average estimates of conditional entropies with different numbers of clusters are used as the
robust interactive effect estimates [22]. Other clustering methods (e.g. in [68]) can be used for more complicated quantization of continuous variables. We will test the performance of the following specific methods in this category:

- **SynDichoMean**: dichotomize by sample mean estimates.

- **SynDichoUPGMA**: quantize by the UPGMA clustering [22]. Implementation details can be found in Section 1 of Appendix A.

### 3.1.1.2 Simplified Gaussian Assumptions

Instead of quantization, we can also estimate conditional entropies by assuming that input variables follow simple distributions as done in [32]. For example, assuming that \( x \) follows a Gaussian distribution (a multivariate Gaussian distribution if \( x \) denotes multiple variables), we can obtain the corresponding maximum likelihood estimates (MLE) of the distribution parameters, the expectation and standard deviation in this case. Given the standard deviation \( \sigma \) (or the covariance matrix \( \Sigma \)), the entropy \( H(x) \) can be directly computed as \( \frac{1}{2} \ln(2\pi e\sigma^2) \) (or \( \frac{1}{2} \ln |2\pi e\Sigma| \) for multivariate cases). Based on this, we can then estimate the synergy \( \text{Syn}(x_i, x_j; y) \). In our simulation experiments, we will test its performance and refer to this method as \( \text{SynGaussX} \).

### 3.1.2 Classification-based Methods

#### 3.1.2.1 Logistic Regression

The most commonly used statistical model for relating input variables \( x \) and a given binary outcome \( y \) is arguably the logistic regression model [69], in which the effect of a variable on the outcome can be measured by its corresponding model coefficient after fitting with the given data. Based on this, the interactive effect of two variables \( x_i \) and \( x_j \) on the outcome \( y \) can be computed similar to individual effect by \( -\log(p) \) where \( p \) is the p-value of the coefficient \( \beta \) after fitting the following logistic regression model \( \log(g/(1 - g)) = \alpha_0 + \alpha_1 x_i + \alpha_2 x_j + \beta x_i x_j \), in which \( g = p(y = 1|x_i, x_j) \) [34]. We later refer to this method as \( \text{LogRegCoeff} \) in our simulation experiments and we note that \( \text{LogRegCoeff} \) is a quadratic method due to the integration of the interaction terms.
3.1.2.2 Relative Expression Analysis

This method has been proposed by [5] to find interactive pairs of genes based on their relative expression. Given two input variables $x_i$ and $x_j$, the interactive effect associated with the outcome $y$ can be measured by $IE = |P(x_i < x_j|y = 0) - P(x_i < x_j|y = 1)|$. The estimate for $P(x_i < x_j|y = 0)$ is the frequency of the observations with $x_i < x_j$ and $y = 0$ and similarly $P(x_i < x_j|y = 1)$ is the frequency of observations with $x_i < x_j$ and $y = 1$ by MLE. Looking deep into their method reveals that they have used the linear function $x_i = x_j$ to quantize data points in the two-dimensional plane $(x_i, x_j)$ and assumed that this line is a proper quantization boundary. Based on this assumption, they expect that for highly interactive pairs, the line $x_i = x_j$ better discriminates samples with larger IE. We also evaluate its power of detecting interactions in our experiments and refer to this method as REA.

3.1.2.3 Linear Classifiers

For highly interactive pairs of variables, it is intuitive to expect that we can train simple classifiers that possess certain discriminating power. Based on this, a procedure based on permutation tests has been proposed in [4] to test each pair of input variables $x_i, x_j$ to find significantly interacting pairs by estimated classification accuracies based on Linear Discriminant Analysis (LDA). They perform two empirical significance tests: the first to rank variable pairs by estimated accuracies and the second to check whether this accuracy is only due to interactive effects of variables by permuting one variable in each pair. Based on these tests, highly ranked pairs were considered significantly interacting pairs. However, the procedure depends on the relative difference of estimated accuracies and does not provide a measure for evaluating interactive effects. Motivated by this procedure, we propose new interactive effect estimates based on the classification performance in the following sections.

3.2 Our Proposed Methods

3.2.1 New Information Theoretic Measures

In this section, we first propose two new methods to estimate information theoretic synergy for continuous random variables based on supervised modeling.
3.2.1.1 Supervised Quantization

Previous information theoretic measures are done in an unsupervised fashion in the sense that either quantization or distribution assumptions of $x$ are made without considering $y$. Hence, it may lose critical predictive information as the approximation by quantization may destroy the underlying structure of data. We explore new supervised quantization for better estimates of interactive effects by taking into account the information given in $y$. More specifically, we propose new supervised quantization approaches based on existing classification methods at different levels of complexity, in which we search for the best possible separating boundaries detected by a classification algorithm for quantization. In other words by learning a classifier, all the possible separating boundaries based on the type of classifiers e.g. line (linear classifier), hyperbola (quadratic classifier), and Voronoi diagram (KNN classifier) can be explored. The separating boundary which minimizes the error is assumed to be the most appropriate boundary for quantizing $x$. To accomplish this, we first learn a classifier to predict $y$ from $x$, and then for each observation $x^{(\ell)}$, $1 \leq \ell \leq n$ we predict its corresponding output $y^{(\ell)} \in \{0, 1\}$ by the learned classifier, which is considered as the new quantized value of $x^{(\ell)}$. Similar to previous information theoretic measures using quantization, we can now estimate the probabilities and compute conditional entropies using quantized variable $x$. Note that depending on the classifier used, the separating boundaries may change. For example, simple modeling methods like LDA will find the best possible linear boundaries to categorize the data $x$ while KNN (K Nearest Neighbor classifiers) can find more complex boundaries with increasing $K$ but may overfit the data. An illustrative example of how different methods may quantize $(x_i, x_j)$ is shown in Figure 4. We expect that the integration of the outcome information and exploring non-linear interactions among variables can capture more general interactive relationships among variables than unsupervised linear ones like the previously discussed methods [22, 5, 4]. In order to test our hypothesis, we evaluate the performance of the following supervised quantization methods:

![Figure 4. Illustration for how different quantization methods work. A: Linear separating boundary. B: Quadratic separating boundary. C: UPGMA clustering based separating boundary. D: KNN based separating boundary.](image-url)
• **SynDichoLDA**: LDA classifier (MATLAB implementation) [70] is used to find a linear separating boundary to quantize \( x \).

• **SynDichoQDA**: Quadratic Discriminant Analysis (QDA) classifier (MATLAB implementation) [70] is used to find a quadratic separating boundary to quantize \( x \).

• **SynDichoKNN**: KNN classifier is used to find a non-linear separating boundary to quantize \( x \). In order to avoid overfitting, we implement a method similar to **SynDichoUPGMA** and estimate interactive effects by averaging over multiple KNN classifiers with different \( K \). Implementation details can be found in Section 2 of Appendix A.

### 3.2.1.2 Estimating \( P(y|x) \) by Supervised Learning

In this section, we propose another new way to estimate conditional entropies \( H(y|x) \) by directly estimating the \( P(y|x) \) without quantization. Usually, classification methods are used to learn the relation between input variables \( x \) and the outcome \( y \). After learning the classifier, for any observation \( x^{(\ell)} \), \( 1 \leq \ell \leq n \), the classifier can predict its corresponding most probable output \( y^{(\ell)} \in \{0, 1\} \). In addition, many classification methods, such as Naive Bayes, Logistic Regression, LDA, and QDA, can provide estimates of the posterior probabilities \( P(y|x = x^{(\ell)}) \). For example, in LDA, to estimate the posterior probability, first the probabilities \( P(y), P(x), \) and \( P(x|y) \) are estimated by assuming Bionomial and Gaussian distributions respectively, and the posterior probability \( P(y|x) \) is then computed using the Bayes rule: 

\[
P(y|x) = \frac{P(x|y)P(y)}{P(x)}.
\]

In other classifiers including Decision Trees, Support Vector Machines (SVM), and KNN, the posterior probability \( P(y = y^{(\ell)}|x = x^{(\ell)}) \) for a given input \( x^{(\ell)} \) can be considered as a value describing how confident the classifier is on \( y^{(\ell)} \). For example, in KNN, if from the \( K \) nearest neighbors of \( x^{(\ell)} \), \( t \) of them have class label 1, then we can approximately estimate that the posterior probability \( P(y = 1|x = x^{(\ell)}) = \frac{t}{K} \) and the posterior probability \( P(y = 0|x = x^{(\ell)}) = \frac{K-t}{K} \). With these estimates of \( P(y|x) \), we can always estimate the conditional entropies \( H(y|x) \) using following formula:

\[
H(y|x) = \int_{-\infty}^{+\infty} P(x = x) \left[ \sum_{y \in \{0,1\}} -P(y = y|x = x) \log(P(y = y|x = x)) \right] dx.
\]  

(3.2)

But the problem with the above formula is that we need to estimate \( P(x) \), for example, by making a Gaussian assumption or by quantization as discussed earlier. Such assumptions and approximations may lead to
inaccurate estimates of the desired conditional entropies. To overcome this problem, the empirical distribution of \( x \) based on the given \( n \) observations can be used: \( P(x = x) = \frac{1}{n} \sum_{\ell=1}^{n} \delta(x - x^{(\ell)}) \), in which \( \delta(.) \) is the Dirac Delta function. Substituting this empirical distribution of \( P(x) \) into the equation (3.2) we get:

\[
H(y|x) = \int_{-\infty}^{+\infty} \frac{1}{n} \sum_{\ell=1}^{n} \delta(x - x^{(\ell)}) \left[ \sum_{y \in \{0,1\}} -P(y = y|x = x) \log(P(y = y|x = x)) \right] dx. \tag{3.3}
\]

Taking the integral into summation we get

\[
H(y|x) = \frac{1}{n} \sum_{\ell=1}^{n} \int_{-\infty}^{+\infty} \delta(x - x^{(\ell)}) \left[ \sum_{y \in \{0,1\}} -P(y = y|x = x) \log(P(y = y|x = x)) \right] dx. \tag{3.4}
\]

Also, based on the definition of the Dirac Delta function, we have \( \int_{-\infty}^{+\infty} \delta(x - \alpha) f(x) dx = f(\alpha) \), which implies that:

\[
H(y|x) = \frac{1}{n} \sum_{\ell=1}^{n} \sum_{y \in \{0,1\}} -P(y = y|x = x^{(\ell)}) \log(P(y = y|x = x^{(\ell)})).
\]

Based on the above formula we can efficiently estimate \( H(y|x) \). In our simulation experiments, we test the performance of the following specific methods for measuring interactive effects based on the direct estimation of \( P(y|x) \):

- **SynPosteriorLDA**: The posterior probability estimated by LDA (MATLAB implementation) [70].

- **SynPosteriorQDA**: The posterior probability by QDA (MATLAB implementation) [70].

- **SynPosteriorKNN**: The posterior probability estimated by KNN. For more detailed information about the implementation details to avoid overfitting, please refer to Section 2 of the *Appendix A*.

### 3.2.2 Methods Based on Classification Accuracy

We further introduce new methods for estimating interactive effects directly based on classification accuracy. For two continuous variables \( x_i \) and \( x_j \) and the outcome \( y \), we measure the interactive effect as

\[
IE(x_i, x_j; y) = \text{Acc}(x_i, x_j; y) - \text{Acc}(x_i; y) - \text{Acc}(x_j; y) \tag{3.5}
\]
in which \( \text{Acc}(x; y) \) is the training accuracy of a chosen classifier with input \( x \) and output \( y \). In this new measure, similar to the definition of information theoretic synergy we try to capture the interactive effect by subtracting the individual effect of individual variables. In addition, to obtain more reliable accuracy estimates, we use the posterior probability to weigh the correct and incorrect predictions as follows:

Consider a vector \( p = [p_1, p_2, ..., p_n] \) as the posterior probability vector obtained by the classifier for inputs \( x = [x^1, x^2, ..., x^n] \), where \( p_\ell = P(y = 1|x = x^\ell) \) and \( y = [y^1, y^2, ..., y^n] \) is the actual outcome vector with \( y^{(\ell)} \in \{-1, 1\} \) (instead of \( \{0, 1\} \)). The accuracy measure used in (3.5) is defined as

\[
\text{Acc} = \frac{(p - 0.5) \cdot 1^T}{n},
\]

in which \( 1 \) is an \( n \)-dimensional vector with all elements equal to 1. Based on the above formula, \( \text{Acc} \in [-0.5, 0.5] \), where -0.5 corresponds to the cases with all inverse predictions, 0.5 for all correct predictions, and 0.0 for random predictions. Consequently, this new measure IE takes a value between -1.5 and 1.5. We will test the following methods for measuring the interactive effect based on classification accuracy:

- **\( \text{AccLDA} \):** The training accuracy obtained by LDA [70] is used to estimate interactive effect.

- **\( \text{AccQDA} \):** The training accuracy by QDA [70] is used.

- **\( \text{AccKNN} \):** The training accuracy obtained by KNN is used. For more detailed information about the implementation details to avoid overfitting, please refer to Section 2 of Appendix A.

**3.2.3 Methods Based on Association Analysis**

We propose to measure the interactive effect between variables \( x_i \) and \( x_j \) with respect to the outcome by comparing the statistical association between \( x_i \) and \( x_j \) given \( y \), denoted by \( \text{ASC}(x_i; x_j|y) \), and the statistical association between \( x_i \) and \( x_j \) without knowing the value of \( y \), denoted by \( \text{ASC}(x_i; x_j) \). We expect that for highly interactive pairs, \( \text{ASC}(x_i; x_j|y) \) will be larger than \( \text{ASC}(x_i; x_j) \) as two variables are more statistically associated after observing the outcome. Based on this, we can measure the interactive effect as the difference between \( \text{ASC}(x_i; x_j|y) \) and \( \text{ASC}(x_i; x_j) \):

\[
\text{IE}(x_i, x_j; y) = \text{ASC}(x_i; x_j|y) - \text{ASC}(x_i; x_j).
\]  

(3.6)

We can use any of the existing measures to test the association or dependence between two variables in the above formula including Pearson correlation or mutual information. Recently, a new dependence measure
called Maximal Information Coefficient (MIC) has been proposed [71]. Unlike the usual association measures which mostly look for specific functional associations (such as linear, exponential, periodic, etc.), MIC can find a wide range of relationships among variables. Using MIC as the dependence measure, we estimate the interactive effect as:

\[
IE(x_i, x_j; y) = \text{MIC}(x_i; x_j | y) - \text{MIC}(x_i; x_j).
\] (3.7)

We refer to this method for measuring the interactive effect as \textit{MICIE} in our simulation experiments. The implementation details are given in Section 3 of Appendix A. It is notable to say that if we use the mutual information \( I(x_i; x_j) \) as the measure for association, the corresponding interactive effect measure in (3.6) is in fact equal to the synergy \( \text{Syn}(x_i, x_j; y) \) by straightforward algebraic manipulation.

### 3.2.4 Performance Evaluation by Simulated Data

We evaluate and compare the performance of all the previously presented methods based on simulated datasets. We simulate random datasets by extending a case-control disease model adopted in [23]. In order to generate datasets with interactive effects, we first randomly assign an outcome variable \( y \) uniformly from \{0, 1\}. For each sample, based on the outcome value, we generate a sample of input variables as follows: We first randomly generate 50 input variables \( x_1, x_2, \ldots, x_{50} \). Among these input variables, we randomly select six of them to simulate the individual effects and another six distinct pairs of other variables from all 1,225 possible pairs of variables to have significant interactive effects on disease outcome \( y \). Extending the method in [23] to capture more general interacting relationships among variables, we simulate three different types of interactions between two interactive variables \( x_i, x_j \) and the outcome \( y \): “simple”, “complex”, and “very complex”, which are depicted in Figure 5. These simulated interacting pairs show high interactive effects with respect to \( y \) while weak individual effects. Among six pairs of interacting variables, we simulate the data by including two pairs with “simple” interactive effects; two “complex” interactive effects, and the remaining two “very complex” interactive effects. If the outcome variable is 1, we randomly assign a value drawn from an equally weighted Mixture-of-Gaussian (MoG) with two Gaussian components with the means \((-0.5, -0.5)\) and \((0.5, 0.5)\) for “simple” interacting pairs \((x_i, x_j)\); assign a random value drawn from an equally weighted MoG with five components with the means \((-1, -1), (1, -1), (0, 0), (-1, 1)\), and \((1, 1)\) for “complex” interacting pairs; and finally randomly assign a value drawn from an equally weighted MoG with eight components with the respective means \((-1.5, -1.5), (-1.5, 0.5), (-0.5, -0.5), (-0.5, 1.5), (0.5, -1.5), (0.5, 0.5), (1.5, -0.5), and (1.5, 1.5)\) for “very complex” interacting pairs. Other-
Figure 5. Scatter plots illustrating the relationships between two interacting variables associated with the outcome $y$ in our simulated datasets (Exaggerated) coded by different shapes. A: Simple interactive effect. B: Complex interactive effect. C: Very complex interactive effect.

wise when $y^{(\ell)} = 0$, we assign a random value drawn from an equally weighted MoG with two components with the means $(-0.5, 0.5)$ and $(0.5, -0.5)$ for “simple” interacting pairs; assign a value drawn from an equally weighted MoG with four components with the means $(0, -1), (-1, 0), (1, 0)$, and $(0, 1)$ for “complex” interacting pairs; and assign a value randomly drawn from an equally weighted MoG with eight components with their means equal to $(-1.5, -0.5), (-1.5, 1.5), (-0.5, -1.5), (-0.5, 0.5), (0.5, -0.5),$ $(0.5, 1.5), (1.5, -1.5)$, and $(1.5, 0.5)$ for “very complex” interacting pairs. Furthermore, for the first three variables with individual effects, we randomly assign a value drawn from a Gaussian distribution with the mean 0.5 when $y^{(\ell)} = 1$; otherwise the value is drawn from a Gaussian with the mean $-0.5$. For other three variables, we assign a random value drawn from a MoG with two components whose respective means are 0.5 and 1.5 when $y^{(\ell)} = 1$ and otherwise the value is drawn from a MoG with the means $-0.5$ and $-1.5$. Finally, for the rest of 50 variables, the corresponding values are either drawn from a two-component MoG with the means at $-0.5$ and 0.5 or from a four-component MoG with their corresponding means equal to $-1.5, -0.5, 0.5$, and 1.5 regardless of $y$. We have also introduced 10% noise to the outcome by flipping the simulated outcome with 0.1 probability after generating each sample. Finally, for all the Gaussian distributions in the above simulations, the standard deviation is fixed at 0.5. The above steps for simulating a single sample are summarized in Algorithm 1 in Section 4 of the Appendix A. From this case-control disease model, we have simulated 1,000 datasets for each one of the following sample sizes: 20, 40, 60, 80, 100, 120, and 140. With the corresponding datasets, we first estimate interactive effects for all 1,225 pairs of variables based on different methods. Based on the Receiver Operating Characteristic (ROC) analysis of estimated interactions, we evaluate these different interaction detection methods by the AUC (area under ROC
curve) values comparing to the ground truth in simulated datasets. In other words, the variable pairs are first sorted based on their estimated interactive effects. Assuming that the top $i$ pairs are the identified interacting pairs for a given method, true positive and false positive rates are calculated for all $i$ ($i = 1, 2, \ldots, 1225$) by comparing the top-ranked pairs with the actual interacting pairs with simulated interactive effects. Then the ROC curve is built using those true positive and false positive rates. Finally, the AUC is calculated as the performance of the method. The average AUC values obtained by different methods from different categories are shown in Figure 6 (A-D). Furthermore, to better illustrate how different methods compare to each other, we have put together all the curves (except the obvious low performing REA) in Figure 6(E). In order to better illustrate the performance differences under small sample size and large sample size scenar-

![Figure 6](image)

**Figure 6.** Simulation dataset results. A: Existing information theoretic methods. B: Existing classification based methods. C: Our proposed information theoretic methods. D: Our proposed classification based and association based methods. E: Comparison between all methods.
Figure 7. Simulation dataset results with 20 and 140 samples. A: The simulation results based on 50 variables B: The simulation results based on 200 variables.

As the results show, the unsupervised clustering based quantization, \textit{SynDichoUPGMA}, is able to outperform the simple mean quantization, \textit{SynDichoMean}, when we have enough samples. In contrast, our supervised quantization method based on KNN (\textit{SynDichoKNN}) provides more accurate estimate of the interactive effects in both small and large sample size scenarios. We note that we have averaged over a wide range of $K$ to avoid overfitting for KNN-based methods (please refer to Section 2 of the Appendix A for more details), which helps improve the performance for the small sample size scenario. Compare the corresponding quantization methods with the methods based on posterior probability estimates, it is clear that further modeling the posterior probability helps improve the detection accuracy. The results also show that our proposed methods, \textit{SynPosteriorQDA} and \textit{SynPosteriorKNN}, have higher AUCS compared to all other information theoretic based methods, in all sample size for 50 variables, which shows that modeling the posterior probabilities can help further improve the performance in cases with a range of different sample sizes. However, with 200 variables and 20 samples (Figure 7 B) the simple Gaussian assumption shows better performance. We believe that this high performance is due to the bias towards Gaussian assumption.
as the variables in the datasets are generated from Gaussian distributions. Finally, as the results clearly show, for smallest sample sizes (20 and 40) our proposed AccQDA have the highest accuracy while for the larger sample sizes (≥ 60) our more complex AccKNN performs better. As we expected, when we have a limited number of samples, AccQDA achieves higher AUC values than AccKNN, probably due to its controlled model complexity compared to AccKNN. If we classify the methods based on their complexity into three classes: linear, quadratic, and complex non-linear modeling, we can see that our new classification accuracy based measures always outperform the other methods in each class, which suggests the effectiveness of this new set of measures regardless of the complexity of the modeling. Based on the same classification, we can see that complex non-linear modeling methods mostly perform better for larger sample sizes while simpler quadratic modelings are better for smaller sample sizes as they have a lower chance of overfitting the data. Furthermore, as the results clearly show, all the methods based on linear models SynDichoLDA, SynPosteriorLDA, and AccLDA fail to capture interactive effects when the given datasets contain more general interacting relationships. Nevertheless, our proposed AccLDA is the best performing method among all the linear modeling methods. The performance of the REA method is the worst among all the methods that we have tested as it uses a fixed linear separating boundary.

3.3 Run Time Analysis

We have compared the methods based on the running time that it takes for each method to calculate all the interactive effects in a set of simulated datasets. We have performed this experiment based on simulated datasets with 50 variables and for the following same sample sizes: 20, 40, 60, 90, 100, 120, 140. We have generated 10 datasets for each sample size. The average time each method takes to calculate all the interactive effects of all the datasets is reported in the Figure 8. As we expected, the simple linear methods are significantly faster than the quadratic method which are significantly faster than the other more complex methods. However, in real-world applications, the calculation of interactive effects is usually only done once for a dataset and can be implemented in parallel. The accuracy of the method is much more important than the time it takes to estimate interactive effects. Moreover, the results show that our quadratic method AccQDA provides a reasonable balance between the run time and the accuracy for the small sample size scenario (as shown in Figure 6 of the main manuscript). We further provided the time it takes for calculating the interactive effects for different sample sizes in Figure 9. As the results show, the running time of the complex methods grow very rapidly with the number of samples, while other methods do not show a
significant increase in running time. Also, the logistic regression based method shows higher running time in the small sample size, which is due to the fact that with small sample size the optimization algorithm for finding the parameters of the model does not converge.

Figure 8. Run time comparison of all methods using simulated datasets.

The methods discussed in this study analyze each variable pair respectively to estimate their potential statistical associations with respect to the outcome. Due to multiple hypothesis testing, these methods may lead to large false positives. Another way to investigate the problem is by penalized feature selection [3, 72]. For example, we can test interactive effects by simultaneously modeling all the variables in the system using LASSO regression [73]. To study interactive effects by LASSO, \( p(p - 1)/2 \) pairwise interaction terms \((x_i; x_j)\) in addition to \( p \) original variables can be included in the regression model. As the number of terms in LASSO grows quadratically with respect to the number of variables, it can quickly become computationally and statistically prohibitive, especially when we have only a limited number of samples in many biomedical applications. In order to make sure that the studied pairwise methods have reasonable performance with small sample size compared to LASSO-based methods (generalized linear models with \( L_1 \) penalty), we also evaluated the performance by the LASSO model and compare it with all the previous methods based on the same simulated datasets. In addition to 50 original variables in the LASSO model, we
Figure 9. Run time comparison of all the methods in this chapter for different sample sizes using simulated datasets.

also added 1,225 interaction terms to the model. We fit the LASSO model with the $L_1$-norm regularization coefficient being selected by a standard 10-fold cross-validation procedure (MATLAB implementation [74]). The absolute values of the fitted coefficients are ranked for calculating the AUC based on the ground truth in simulated datasets. Similar to other methods, the performance of LASSO has been evaluated for different sample sizes using the same 1000 simulated datasets. The results are shown in Figure 6(E) together with all the other methods. As the results show, the performance of LASSO is worse than all other non-linear methods when we have relatively small numbers of samples. Based on these results, our proposed pairwise association estimates can provide effective screening for feature selection with the consideration of potential interactions among features, following which more thorough modeling can be implemented with reduced model complexity.
3.4 Statistical Significance

All the measurements discussed in this chapter are based on the statistics estimated from data. Knowing the distribution of the corresponding null hypothesis for different statistics is very important to avoid false discovery by making sure that the variable pairs with high interactive effects can be shown to have statistically significant interactions. For discrete random variables, [32] studied the mutual information and derived a distribution for its corresponding null hypothesis. Also, [33] studied the statistical significance for interactions among quantized variables and provided the null hypothesis distribution. However as they discussed, the performance depends on the sample size. For all the other methods, the significance analysis has always been performed by empirically estimating the null hypothesis distribution using permutations as done in [22, 4], which is computationally demanding. These permutation methods can be similarly applied to all of our proposed methods as well.
Chapter 4
Network-based Feature Ranking

1In this chapter we propose a novel feature ranking approach based on “synergy network” which takes into account the interactive effects among features in addition to their individual effects to rank the features. Based on the measured interactive effects and individual effects of all the features, we construct a synergy network $G(V, E)$, in which $V$ is the set of nodes representing the features. Each node $v_i \in V$ is weighted by $f(v_i)$, where $f(v_i)$ is equal to the measured individual power of the $i$th feature on the outcome and $E$ is the set of edges $(v_i, v_j)$ with edge weight $s(v_i, v_j)$ where $s(v_i, v_j)$ is equal to the measured interactive effect between $i$th and $j$th features. We then propose a novel algorithm based on this synergy network to identify biomarkers for predicting the disease outcome.

4.1 Finding Subnetworks for Biomarker Identification

As explained, the synergy network integrates both individual and synergistic (interactive) power of features in a single graph structure. Similar to the traditional problem of feature selection, here we are looking for subsets of features or sub-networks in the synergy network, with the highest possible discriminative power regarding disease outcome $y$. To simplify the problem, we approximate the discriminative power of sub-networks by the summation of the node weights and edge weights induced in them. We note that this approximation is expected to perform better than traditional feature selection approaches based on only individual effects [3] due to the integration of synergistic effects in our synergy network. The biomarker identification problem is then reduced to solving the following optimization problem:

$$\max_{C \subseteq G} \sum_{v_i \in C} f(v_i) + \lambda \sum_{v_i, v_j \in C} s(v_i, v_j),$$

(4.1)

1This chapter was published in EURASIP Journal on Bioinformatics and System Biology, Vol. 2013, pp. 12, 2013. Permission is included in Appendix B
where $C$ denotes potential sub-networks and $0 \leq \lambda \leq 1$ is a weighting coefficient between individual and synergistic effects. As both $f(v_i)$ and $s(v_i, v_j)$ are non-negative, the previous optimization problem has the degenerate solution which includes all features in $C$. To overcome this problem, we further impose another constraint to restrict the size of selected sub-networks to have $|C| \leq K$. This formulation is in fact equivalent to the problem of finding a maximum weighted clique (MWCP) [75] which is a generalization of the classical maximum clique problem (MCP). As MCP is NP-hard [76], it can be easily shown that MWCP is NP-hard as well. Thus our biomarker identification problem formulated in (4.1) is also an NP-hard problem. Several approaches have been previously proposed to find the exact optimal solution of the problem by employing branch-and-bound techniques but it is probable that exhaustive search over all possible sub-networks is needed [75]. In this chapter, we propose a fast approximate algorithm for MWCP which also provides a ranked list of features based on both their individual and interactive effects.

4.2 Feature Ranking by a Graph Spectral Algorithm

We first rewrite the optimization problem given in (4.1), as a quadratic integer programming problem as follows: For each node $v_i$ in $G$, we consider an integer variable $x_i$ which is equal to 1 if the node $v_i$ is selected in the sub-network $C$ and is 0 otherwise. Using these variables we can rewrite (4.1) as

$$\max_{\mathbf{x}} \sum_{i=1}^{n} f(v_i) x_i^2 + \lambda \sum_{i,j=1}^{n} s(v_i, v_j) x_i x_j,$$

where $n$ is the number of feature nodes in $G$. We further define the matrix $M_{n \times n}$ with diagonal entries $M_{i,i}$ equal to the individual power $f(v_i)$, and off-diagonal entries $M_{i,j}$ equal to the synergistic power $\lambda \times s(v_i, v_j)$. Using this matrix, we can write the optimization problem for biomarker identification in the following matrix format:

$$\max_{\mathbf{x}} \mathbf{x}^T M \mathbf{x} \quad (4.2)$$

s.t. $\mathbf{x}^T \mathbf{x} \leq K$;

$x_i \in \{0, 1\},$

in which $\mathbf{x} = [x_1, \cdots, x_n]^T$ is a binary integer vector. In fact, the size constraint is equivalent to putting in a sparse penalty on $\mathbf{x}$ to select the smallest number of features that have high predictive power. In order to solve this constrained quadratic integer programming problem, we develop a spectral approximate algorithm. We first relax the integer variable $x_i \in \{0, 1\}$ to $x_i \in \mathbb{R}$. Then using Lagrangian relaxation we can transform the original optimization problem given in (4.2) to the following optimization problem:
\[
\max_x \ x^T M x + \alpha(K - x^T x),
\]
(4.3)

where \( \alpha \) is the Lagrangian multiplier. Based on the Karush-Kuhn-Tucker (KKT) condition \cite{77}, the optimal solution of this relaxed quadratic problem has to (necessarily) satisfy the condition that the derivative of the relaxed objective function equals 0:

\[
\frac{\partial}{\partial x} \left[ x^T M x + \alpha(K - x^T x) \right] = 0.
\]
(4.4)

By straightforward algebraic manipulations, we can show that the potential solution \( x^* \) has to satisfy \( Mx^* = \alpha x^* \). Therefore, the relaxed solution \( x^* \) to the MWCP is an eigenvector of matrix \( M \). Furthermore, we want the objective function \( x^T M x^* = \alpha x^T x^* = \alpha K \) to have the maximum value with \( x^* \), which means that we want \( \alpha \) to be as large as possible. Hence, the solution \( x^* \) will be the eigenvector of \( M \) with the largest corresponding eigenvalue. Also given the relaxed solution \( x^* \), for any \( K \), the approximate solution to the original integer programming optimization problem is to take top \( K \) nodes with largest corresponding magnitudes in \( x^* \). This also shows that the candidate biomarkers with larger magnitudes in \( x^* \) are more desirable to be selected in the final subset of biomarkers as potential prognostic biomarkers. Thus, we can use the absolute values in \( x^* \) as a score to rank the biomarkers. We note that \( K \) can be an arbitrary number without loss of generality, which will not affect our final ranking as the \( x^* \) only depends on the matrix \( M \).

As one can see, the proposed method combines both individual power and synergistic power among all candidate biomarkers into one single score that can be used to rank them.

### 4.3 Performance Comparison Based on the Simulated Datasets

We simulate a case-control disease model, in which the outcome \( y \) (disease) follows a Bernoulli distribution with the success parameter equal to \( p(y = 1|V) \) given the input \( V \). We first simulate 30 random variables as input features \( V = v_1, v_2, ..., v_{30} \). From all 435 potential pairs of these randomly simulated features, ten of them are randomly selected to have interactive effects with respect to the outcome. Based on this, we follow the following logistic model to simulate the disease outcome \( y \):

\[
\log \left( \frac{p(y = 1|V)}{1 - p(y = 1|V)} \right) = \alpha_0 + \sum_{i=1}^{30} \alpha_i v_i + \sum_{i \neq j} \beta_{ij} v_i v_j.
\]
(4.5)
In this logistic model, the magnitude of each individual coefficient $\alpha_i$ determines the individual effect of the corresponding feature $v_i$ on outcome $y$ and the magnitude of the interaction coefficient $\beta_{ij}$ determines the amount of interactive effect of two features $v_i$ and $v_j$ on the outcome. To obtain the previously described case-control data, we simulate 30 random features with each feature $v_i$ following a mixture-of-Gaussian distribution with equally weighted (mixture parameters equal to 0.5) Gaussian distributions with the same variance of 1.0 and the means equal to $-1.0$ and $1.0$ respectively. For 435 interaction coefficients $\beta_{ij}$s, we randomly set 425 of them to zero and the values of the other ten are drawn from the standard normal distribution (mean 0.0 and variance 1.0). We also set all the individual coefficients $\alpha_i$s to zero which means that there is no feature with significant individual effect. To simulate the outcome $y$, we first compute the probability $p(y = 1 | V)$ based on the previous logistic model (4.5). Then, we generate the value for $y$ from a Bernoulli distribution with the success parameter equal to $p(y = 1 | V)$. We have generated 20 such case-control datasets with 200 data samples in each set for the performance evaluation of our method. In order to make sure that our performance comparison results are independent of how we set the values of these coefficients, each of these 20 datasets are simulated with different random values for coefficients $\beta_{ij}$s. For the simulated datasets, we use the logistic regression coefficient explained in Chapter 3 to measure both individual effect and interactive effects because the simulated dataset is generated using a logistic model.

To identify the biomarkers using network-based feature rankings, we construct a synergy network based on the given dataset and rank the features using our network-based feature ranking algorithm. After ranking the features using the network-based feature ranking, we perform a forward feature selection [3] to find the biomarkers. To make sure that we do not overestimate the performance of our biomarker identification approach, we perform the following “embedded” cross-validation procedure: Similar to the regular ten-fold cross-validation, we first randomly divide the dataset into ten folds, within which, one fold is used as the testing set to test the performance and the remaining nine folds are used as the training set to select biomarkers and learn the classifier. In order to select biomarkers based on the training set, we first use all the data points in the training set to construct a synergy network and perform our network-based feature ranking algorithm to obtain the ranked list. Then using the ranked list, we perform a forward feature selection method to select the best performing set of biomarkers. In the forward feature selection method, we sequentially add candidate factors to the current feature set (starting with an empty set), if it improves the classification performance; otherwise we move to the next factor in the ranked list. To evaluate the performance of a set of potential risk factors during forward feature selection, we use another standard ten-fold cross-validation.
in which we further divide the *training set* into ten folds, nine of which are used to train the classifier and the remaining is used to test the performance. After performing forward feature selection and identifying the biomarkers, we learn a classifier based on the *training dataset* using those selected features and compute the performance based on the *testing set*. During our performance evaluation procedure, we adopt the MATLAB implementation of Quadratic Discriminant Analysis (QDA) as the classifier [70] to make sure that the pairwise interaction among risk factors is taken into account by the classifier. We use AUC (area under ROC curve) to measure the performance of any classifier in our performance evaluation procedure.

To demonstrate the advantage of our network-based feature ranking, we compare the performance of our ranking with individual-based feature ranking. We use our “embedded” cross-validation procedure to evaluate the performance of both network-based ranking and individual-based ranking. We repeat the “embedded” cross-validation 100 times for both individual and network-based rankings, and calculate the average AUC for both methods. The performance comparison for our 20 simulated datasets is shown in Figure 10. The average AUC of our network-based method among 20 datasets is 0.6518 respectively, compared to 0.5577 obtained by individual-based ranking (Figure 10). As expected, the performance of our network-based ranking is significantly higher than individual-based ranking. This clearly shows that filtering methods based on individual ranking are unable to capture those risk factors with synergistic effects but weak individual effects, which are critical biomarkers for better prediction.

In order to further show that our network-based method does not only bias toward risk factors with only synergistic effects, we further check the performance of our network-based ranking when there are risk factors with significant individual effects in the case-control disease model. We use the same Logistic Regression model in (4.5) where in addition to 10 non-zero interaction coefficients $\beta_{ij}$, we also have 5 random non-zero individual coefficients $\alpha_i$ ($\alpha_0$ is set to zero as well). The values for those non-zero $\alpha_i$s are also drawn from a standard normal distribution. We have also generated 20 datasets of this new model each with 200 samples. Similar to the previous 20 datasets, each of these 20 datasets are simulated with different random values for coefficients $\alpha_i$s and $\beta_{ij}$s. The average AUC obtained by our network-based method among these 20 new datasets is 0.6536 which is significantly higher than 0.6040 obtained by individual-based ranking (Figure 11). This shows that our network-based ranking consistently performs better than individual ranking even when there are features with significant individual effects.
Figure 10. Performance comparison between individual-based and network-based ranking for 20 simulated datasets in which we have weak individual effects and significant synergistic effects.

Figure 11. Performance comparison between individual-based and network-based ranking for 20 simulated datasets in which we have both significant individual effects and significant synergistic effects.

4.4 Biomarker Identification for Breast Cancer Metastasis

In this set of experiments, we test the performance of our spectral network-based ranking to identify bio-makers for predicting breast cancer metastasis. Two micro-array datasets for Breast Cancer metastasis are used in this study. The first one is referred to as “USA” dataset [21] and contains the gene expression profiles of 286 samples, in which 107 eventually developed metastasis. The other dataset, referred to as “Netherlands” dataset [20], contains 295 samples, in which 79 eventually developed metastasis. There are a total of 6,168 genes whose expression were measured in both datasets. We only use these genes as features...
for both datasets. For the genes that are measured by more than one probe in a dataset, we use the average of the corresponding expression values. We used AccQDA for measuring the interactive effects due to the small sample size in both “USA” and “Netherlands” datasets. We also estimate the individual effect of each feature \( x_i \) similarly by Acc\((x_i; y)\).

We evaluate the performance using a standard 10-fold cross-validation evaluation method. Based on nine folds of the data (training set), we construct the “synergy network” and rank the features by the network-based ranking method [7]. Then the top \( T \) features in the ranking list are used to learn an SVM classifier (with complexity parameter \( C = 100 \) and RBF kernel with \( \sigma = 1 \)) based on the same nine folds of training data. The performance of the classifier is then evaluated based on the remaining one fold (test set) by AUC. This procedure is repeated ten times for each test set. The average AUC is then reported as the estimated performance of the network-based biomarker identification. In order to show that the high performance obtained is due to integrating interactive effects, we further use the same 10-fold cross validation procedure described above to measure the performance of individual-based ranking, in which we rank the features only based on their individual effects. The results for both network-based ranking and individual-based ranking for \( T = 1, 5, 10, 15, \ldots, 50 \) are shown in Figure 12(A).

As shown, the performance of network-based ranking that takes into account interactive effects is substantially better than individual-based ranking. We performed a two-sample t-test based on the two set of AUCs (one from network-based ranking and one from individual-based ranking). The p-values obtained for all \( T \)s (\( T \in 1, 5, 10, 15, \ldots, 50 \)) are 0.22, 0.19, 0.04, 0.09, 0.001, 0.005, 0.005, 0.05, 0.03, 0.06, and 0.15 respectively. The trend of p-values further verifies that network-based ranking outperforms the individual-based ranking quite significantly in the range of 15 to 30 top genes, which demonstrates the importance of interactive effects in identifying more accurate biomarkers and better phenotype prediction. Also, as we expected, the first few genes in network-based ranking may not show any improvement over individual-based ranking, which is due to the fact that there might not be significant interactive effects within those first few genes; but the performance of network-based ranking improves when we include more genes that have high interactive effects with the first few genes. As we have performed biomarker identification 10 times during our cross-validation procedure, for each of the top \( T \) features, we have ten possibly different sets of biomarkers based on different training sets. To have a reliable single set of biomarkers, for a given \( T \) we selected the biomarkers which at least appeared in two of the ten biomarker sets and report them as the final biomarker set. In order to make sure that those identified biomarkers are predictive of breast cancer metastasis, we have
tested their performance on the independent “Netherlands” dataset [20, 19]. We trained an SVM classifier (with the same complexity parameter $C = 100$ and RBF kernel with $\sigma = 1$) for the “Netherlands” dataset using the identified biomarkers from the “USA” dataset for each $T$. To evaluate metastasis prediction performance, we have performed 100 repeated 10-fold cross-validations. The results for the final biomarker sets identified by both individual-based and network-based ranking for all $T$s ($T \in 1, 5, 10, 15, \ldots, 50$) are shown in Figure 12(B). As one can see, the network-based ranking not only performs better than individual-based ranking within the dataset, but also performs substantially better than individual-based ranking across the datasets, which again verifies that considering interactions among biomarkers is crucial in identifying more accurate biomarkers that perform stably across datasets. We have also performed the two-sample t-test and the p-values obtained for $T = 1$ and $T = 3$ are 0.24 and 0.02 respectively, where the p-values for all the other $T$s are less than $1 \cdot 10^{-5}$. This further shows that the improvements obtained by considering the interactive effects among features are statistically significant, even when the biomarkers are tested using an independent dataset.

We have further provided the final set of biomarkers for $T = 20$ in Table A.9 in the Appendix A for

![Figure 12](image-url)

**Figure 12.** The AUC for predicting breast cancer metastasis based on network-based ranking and individual-based ranking. A: Results based on the “USA” dataset. B: Results based on the “Netherlands” dataset (An independent dataset) using the biomarkers identified from “USA” dataset.
Table 1 Gene set enrichment analysis for identified biomarkers by network-based ranking.

<table>
<thead>
<tr>
<th>gene symbols</th>
<th>pathway</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFL1</td>
<td>BIOCARTA_RHO_PATHWAY &quot;a&quot;</td>
<td>1.98e-06</td>
</tr>
<tr>
<td>GSN</td>
<td>REACTOME_CELL_JUNCTION_ORGANIZATION</td>
<td>2.96e-05</td>
</tr>
<tr>
<td>ACTR2</td>
<td>REACTOME_CELL_CELL_COMMUNICATION</td>
<td>1.07e-04</td>
</tr>
<tr>
<td>CLDN11</td>
<td>KEGG_OXIDATIVE_PHOSPHORYLATION</td>
<td>1.51e-04</td>
</tr>
<tr>
<td>CLDN5</td>
<td>REACTOME_TIGHT_JUNCTION_INTERACTIONS</td>
<td>2.26e-04</td>
</tr>
<tr>
<td>TESK1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UQCRC2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATP5J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COX15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

"a"Rho cell motility signaling pathway

both the network-based and individual-based rankings. We have performed gene set enrichment analysis using GSEA software [78], in which we have compared the final set of biomarkers with the curated KEGG pathways, BioCarta pathways, and the REACTOME gene set collection. In GSEA, a set of genes that is provided by the user is compared with known gene-sets (the known gene sets can be pathways, functional modules, or any other set of genes that are proven to work together to regulate a biological process). The gene-sets with significant intersection (for details on significance analysis please refer to the section 7 of the Appendix A) with the user provided set of genes will be returned. So, for example, if GSEA returns a pathway that is related to cell motility, it means that the input gene set is highly likely affecting the cell motility and therefore can contribute to metastasis. The results by network-based ranking are provided in Table 1. Interestingly, the final biomarker set obtained by the network-based method has statistically significant overlap with cell junction and communication pathways and the pathways related to cell motility which have been shown to be related to cancer metastasis [79, 80, 81, 82]. In contrast, when we performed the same enrichment analysis using the biomarkers identified by individual-based ranking method (please refer to Section 7 in Appendix A), none of those important pathways was enriched. This clearly shows that the network-based analysis, by considering the interactive effects among genes, has the potential to identify genes that may be involved in biological functions strongly related to certain disease development. As the results show, considering interactive effects can help both in providing biomarkers with higher predictive power and in revealing the underlying cellular mechanisms for disease development.

4.5 Biomarker Identification for Type 1 Diabetes

In this section, we use our network-based feature ranking to identify biomarkers for type 1 diabetes (T1D) based on clinical features. As opposed to the gene expression dataset used for Breast Cancer, the T1D
dataset we use had only 19 features which allows comparison with the exhaustive search feature selection method.

DPT-1 was a study designed to determine if T1D can be prevented or delayed by preclinical intervention with an insulin supplement. It focuses on first and second degree non-diabetic relatives of patients with T1D before the age of 45, since they have more than ten-fold risk of developing T1D compared to the general population [83]. DPT-1 screened 103,391 subjects and categorized them into four risk groups based on genetic susceptibility, age, the presence of autoantibodies (including islet cell autoantibodies (ICA), insulin autoantibodies (IAA), glutamic acid decarboxylase (GAD), and insulinoma-associated protein 2 (ICA512)), and the change of metabolic markers during OGTT and IVGTT. The 3,483 subjects positive for ICA were staged to quantify the projected five-year risk of diabetes [84]. Our analysis focuses on the study for the “high risk” and “intermediate risk” groups [84, 85, 86], which contain 339 and 372 subjects respectively. The subjects of each group were randomly divided into two roughly equal sub-groups: one received a parenteral insulin supplement while the other was assigned to the placebo arm of the study. In this experiment, we focus on the subjects of the Placebo group. We consider the placebo subgroups of both “high risk” and “intermediate risk” groups as a dataset for our analysis. The dataset contains the following 19 features from baseline characteristics in DPT-1, focusing on immunologic and metabolic markers. We have taken the available titer values for different autoantibodies, including ICA, IAA, GAD, ICA512, and MIAA (micro-insulin autoantibodies). For metabolic indices, we have fasting glucose, glycated hemoglobin (HbA1c), fasting insulin, and first-phase insulin response (FPIR) from IVGTTs. Homeostasis model assessment of insulin resistance (HOMA-IR) and FPIR-to-HOMA-IR ratio are also computed as in [86]. From OGTTs, in addition to 2-hour glucose and fasting glucose, we have collected blood samples for C-peptide measurements in the fasting state and then 30, 60, 90, and 120 minutes after oral glucose, from which we have computed peak C-peptide as the maximum point of all measurements and AUC (area under curve) C-peptide using the trapezoid rule. Furthermore, as age and Body Mass Index (BMI) have been conjectured to be important confounding factors, we also include them in our set of features. We are interested in identifying the most predictive group of features as biomarkers from the above described candidates to predict the outcome which is the development of T1D at the end of the DPT-1 study. The dataset contains 356 subjects within which 133 subjects developed T1D at the end of the study.

To check the performance of our network-based biomarker identification for DPT-1, similar to simulated datasets, we repeated the “embedded” cross-validation 100 times and used the average performance. In or-
der to show the advantage of our network-based feature ranking method, we also compute the performance of individual-based feature ranking. The results are given in Table 2. As one can see, the AUC obtained by our network-based ranking is significantly higher than individual-based ranking with p-value of $7.17 \times 10^{-11}$.

The results obtained based on the DPT-1 dataset clearly show that our spectral network-based feature ranking provides biomarkers with significantly better predictive power than individual-based feature ranking. This also verifies our expectation that the integration of synergistic interaction among features provides biomarkers with higher prediction accuracies.

Table 2 Comparing the performance of the network-based spectral feature ranking with individual-based feature ranking based on the DPT-1 dataset.

<table>
<thead>
<tr>
<th>Individual ranking</th>
<th>Network-based ranking</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6524</td>
<td>0.6724</td>
<td>$7.17 \times 10^{-11}$</td>
</tr>
</tbody>
</table>

In each run of the “embedded” ten-fold cross-validation procedure, we in fact have ten possibly different sets of selected features as we perform feature selection for each fold based on a different subset of training samples for each run of the cross-validation procedure. By repeating this procedure 100 times, we obtain 1000 ($100 \times 10$) different subsets of biomarkers. In order to report a single reliable set of biomarkers, we first compute the frequency of the appearance of each feature and then select the features that at least appeared in 40% of the 1000 (i.e. 400) selected subsets. The single set of biomarkers based on both individual and network-based rankings are provided in Table 3. We have also evaluated the performance of those final biomarkers by 100 repeated ten-fold cross validation and reported the results in Table 3.

Table 3 Final sets of bio-markers and their corresponding performance for the DPT-1 dataset using individual ranking, network-based ranking, and exhaustive search methods.

<table>
<thead>
<tr>
<th>Individual ranking</th>
<th>Network-based ranking</th>
<th>Exhaustive search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age; 2-h glucose; IAA; ICA512; Peak C-Peptide; AUC C-Peptide;</td>
<td>2-h glucose; IAA; FPIR; Fasting glucose (IVGTT); ICA512; Peak C-Peptide; AUC C-Peptide; FPIR-to-HOMA-IR ratio;</td>
<td>2-h glucose; Age; FPIR-to-HOMA-IR ratio; Fasting glucose (IVGTT); IAA; Peak C-Peptide; Weight;</td>
</tr>
<tr>
<td>0.6779</td>
<td>0.7154</td>
<td>0.7227</td>
</tr>
</tbody>
</table>
Due to the relatively small number of features in this study, it is feasible to perform an exhaustive search over all possible subsets of features to find the biomarker set with the best performance. We computed the AUC of all \(2^{19} - 1\) possible subsets based on 100 repeated ten-fold cross-validation. The best performing subset together with its measured performance is also given in Table 3. The results in Table 3 clearly show that network-based feature ranking method provides more predictive biomarkers than the individual-based feature ranking that are closer to the best performing biomarkers by exhaustive search. We further provide in Figure 13 the Venn diagram of selected biomarkers which shows the intersection of biomarkers selected by different methods. As one can see, the intersection between biomarkers selected by our network-based ranking and best possible performing biomarkers is larger than the intersection between biomarkers selected by individual-based ranking and the best possible performing biomarkers.

![Venn diagram illustrating the identified biomarkers using different methods.](image-url)
Chapter 5

Clustering Somatic Mutation Profiles

In this chapter we propose methods to increase the accuracy of tumor stratification based on somatic mutation profiles, which ultimately helps us in finding more accurate biomarkers by selecting the genes that are significantly mutated in each subtype. The somatic mutation data is provided as binary matrix $A_{n \times p}$, in which $A_{i,j} = 1$ indicates that the $j$th gene in $i$th tumor sample is significantly mutated compared to germ line cells. In other words, somatic mutations are mutations that are related to disease as opposed to mutations that regularly occur among the normal population. These mutations are found by enrichment analysis of tumor cell population compared to the normal population [87]. Somatic mutation profiles are a rich source of biomedical data for computational methods including computational tumor stratification as they do not suffer from the typical problem in gene expression data including RNA sample quality and lack of reproducibility between biological replicates [88, 6]. Furthermore, as the somatic mutations are presumed to cause cancer, similarity and differences among the mutation profiles provides valuable information for computational tumor stratification. However computational methods face two main issues in dealing with somatic mutation profiles:

1) Most of the clustering methods (e.g. kmeans, hierarchical clustering) [36, 37, 38, 40] depend on a distance (or similarity) measure between the samples, which is measured by, for example, Euclidean or Hamming distance. However, as discussed in the introduction, somatic mutation profiles are highly heterogeneous, i.e., it is very common that two clinically similar tumors share only a few mutations. For example the genes mutated in two tumors might not be the same but they may affect the same molecular pathways or functional processes, which hence cause similar functional aberration.

2) Somatic mutation profiles are very sparse. Among the tens of thousands of genes at most a few hundred are mutated for each tumor sample. When characterizing distances between tumor samples using traditional Euclidean or Hamming distances, the tumors with a larger number of 1s in their profile, usually have larger distances to all other tumors. In other words, there is a positive correlation between the number of 1s in the profile and its distance to other tumor samples. In Figure 14(A) we have illustrated this dependency by
Figure 14. The pairwise distances between 504 Breast Cancer tumors are depicted with a gray scale map (the distance values are quantile normalized to a normal distribution to achieve better contrast for visualization purposes). The rows and columns are sorted (in decreasing order) based on the number of 1s in the mutation profiles. A: The pairwise Hamming distance between mutation profiles. As the figure shows, there is a very clear positive correlation between the number of 1s in the tumors and its distance to all other tumors. Tumors with larger number of 1s have larger distances to all other tumors which is due to sparseness of the mutation profiles. B: The distance matrix after normalization. As expected, the dependency of distance on the number of ones is eliminated: The distances in both top-left and bottom right corners are small as they show distance between two vectors with very small number of 1s and two vectors with very large number of 1’s respectively (these happen in very extreme cases, but as we used quantile normalization for visualization proposes, these effects are exaggerated). In both cases distance must be small. The distances in two other corners are large because they show the distance between a vector with large number of 1s and a vector with small number of 1s, which must be large.

drawing the distance matrix between the mutation profiles based on the Breast Cancer dataset from TCGA (for more information about the dataset please refer to Section 5.2.1 in this chapter). Even after addressing the non-linearity and heterogeneity using prior biological knowledge (as explained in the following), the gene-set space representation of the mutation profiles still suffers from such a bias.

In the rest of this chapter we show gene-set mutation profile representations based on prior biology knowledge can help better characterize the distance between somatic mutation profiles. We also propose a simple, yet effective method to address the sparseness problem and eliminate the bias due to the dependency of the defined distance on the number of mutated gene-sets in the gene-set representation of the somatic mutation profiles.
5.1 Calculating Distances Between Somatic Mutation Profiles Using Gene-set Representations with Prior Knowledge

There are valuable sources of biological knowledge annotating functional relationships among biomolecules such as pathway databases [89] and also in Gene Ontology (GO) annotations [90, 91]. The archived pathways are groups of genes that are proved to be functionally related in cell biological processes and are usually compiled and curated by domain experts. We unify the format of biological knowledge in both pathways and GO annotations datasets by representing them as gene-sets. We consider a gene-set as a set of genes which are proved to be functionally related based on the knowledge provided in such biological knowledge datasets. For each pathway in the pathway dataset, we consider a gene-set that contains all the genes in that pathway. For each GO term, we consider a gene-set in which we include all the genes annotated to that GO term. Given a set of \( m \) gene-sets, we represent each tumor by a binary vector of length \( m \) where we set the \( i \)th element equal to 1, if the \( i \)th gene-set is affected by the mutation (in other words, at least one of the genes in the gene-set is mutated in the tumor). This is based on the assumption that, if one of the genes in a pathway or functional module fails, the module is not able to function anymore. Although biological systems usually show high levels of fault tolerance, we believe that somatic mutations are the ones affecting critical genes in the modules as they are enriched in the disease population. Then the distance between tumors is calculated as the Hamming distance between the gene-set profiles. Furthermore, due to the sparseness of the somatic mutation profiles, the gene-set representation of the tumors also suffer from the problems discussed in the previous section. We further eliminate the dependency of the distance on the number of 1s by normalizing the Hamming distance as follows. We denote the Hamming distance between \( i \)th and \( j \)th tumor by \( d_{ij} \) and calculate the normalized distance denoted by \( d_{ij}^{\text{norm}} \) by following formula:

\[
 d_{ij}^{\text{norm}} = \frac{d_{ij}}{\sum_{p=1}^{n} d_{ip} \times \sum_{p=1}^{n} d_{pj}} \tag{5.1}
\]

After normalizing the distances, we use a simple Hierarchical Clustering (Matlab implementation of UPGMA) to find a pre-specified number of clusters. Our normalized distances between the original mutation profiles are depicted in Figure 14(B). As expected, the dependency of distance on the number of 1s is eliminated after normalization, as discussed in the figure caption.
5.2 Results and Discussion

5.2.1 Breast Cancer Dataset

We use somatic mutation profiles of breast cancer provided on the TCGA website [1]. The dataset we used contains somatic mutations of 16,711 genes for 504 Breast Cancer tumor samples. To evaluate the performance of a stratification method we compare the results with the actual Breast Cancer subtypes provided in TCGA. Breast cancer subtypes are characterized by measuring the status of Estrogen Receptor (ER), Progesterone Receptor (PR), and HER2/neu receptor (HER2) genes in the cell [92]. Based on these molecular markers, the Breast Cancer tumors are usually classified into four important subtypes: Luminal A (ER+ and/or PR+, HER2-), Luminal B (ER+ and/or PR+, HER2+), Triple negative/basal-like (ER-, PR-, HER2-), and HER2 (ER-, PR-, HER2+). For some of the tumors in the dataset the test for the status of HER2 returned equivocal. We denote the equivocal outcome by HER2* and consider 4 other subtypes based on that: (ER-, PR-, HER2*), (ER+, PR-, HER2*), (ER-, PR+, HER2*), and (ER+, PR+, HER2*).

5.2.2 Pathways and GO Annotation Data

For our pathway analysis, we downloaded the curated gene sets collection (C2) from the Molecular Signatures Database (MSigDB) [89] that contains a total of 4,725 gene-sets that are collected from various on-line pathways datasets. For our GO Annotation analysis, we downloaded the GO Annotations for Homo sapiens from the Gene Ontology Consortium website. We used the available functions in the Matlab Bioinformatics toolbox to extract the gene annotation information for both Biological Processes (10,646 gene-sets) and Molecular Functions (3,785 gene-sets) aspect of the GO. In our experiment we test the performance based on the gene-sets obtained from both aspects.

5.2.3 Performance Evaluation Measure

We computed the Normalized Mutual Information (NMI) between tumor stratification results and the actual subtypes of Breast Cancer. We can take two vectors of positive integers with the same size $n$, $c_1$ and $c_2$, to denote assigned subtypes for given samples. For example, $c_1$ is the outcome from our gene-sets based method with $n_1$ identified subtype groups and $c_2$ gives the ground truth stratification based on the actual Breast Cancer subtypes with $n_2$ curated labels ($n_2 = 8$ in our Breast Cancer dataset as explained previously). To measure the performance of tumor stratification we calculate the NMI between two vectors
c_1 and c_2 using following formula: $R = \frac{I(c_1;c_2)}{H(c_1)+H(c_2)}$ in which, $H(x)$ is equal to the information theoretic entropy of a random variable $x$ and $I(x,y) = H(x) + H(y) - H(x,y)$ is the mutual information between $x$ and $y$. This normalized mutual information takes the value of zero if two vectors are independent and the maximum of $R_{\text{max}} = \frac{\min(H(c_1),H(c_2))}{H(c_1)+H(c_2)}$ where the values are completely dependent. We divide the NMI by its maximum to obtain a value between 0 and 1 and report this number as the final performance measure. The higher is $\frac{R}{R_{\text{max}}}$, the better correlated are $c_1$ and $c_2$.

5.2.4 Experimental Results

We tested the performance of our gene-sets based representation methods including pathway based, GO Annotation based on biological Processes aspect (GO Biological Processes), and GO Annotation based on Molecular Functions aspect (GO Molecular Functions) and compared them with the performance of the original somatic mutation profile (similar to our approaches we use normalized hamming distance and Hierarchical clustering to stratify based on the original somatic mutation profiles) and the state-of-the-art NBS approach. For all stratification methods, we looked for 4 subtypes which are the number of actual subtypes in Breast Cancer. We performed a sub-sampling 20 times during which we randomly selected 80% of the tumors and 80% of the genes. For each sampled dataset, we found the subtypes and calculate the performance using all the methods under investigation. The results for each sampled dataset are shown in Table 4. As the results clearly show our GO Biological Processes representation achieves an average NMI of 0.1126 which is higher than than the rest of methods and outperforms the state-of-the-art NBS method that achieves average NMI of 0.0807, with p-value $< 2.77e - 05$ (based on ttest). As the results shows the methods based on gene-sets are performing better than the original somatic mutation profiles and the state-of-the-art NBS methods in identifying Breast Cancer Subtypes. Among them the gene-sets representation based on GO Biological Processes perform best. Furthermore, to show the importance of normalizing distance when dealing with sparse binary vectors, we performed Hierarchical clustering on GO Biological Processes representation (the best performing method) without normalizing the pairwise distances, which results in a very poor performance. The result clearly shows that the normalization is necessary for such sparse binary vectors to help alleviate the dependency of the distance on the number of 1s in the vectors.

5.2.5 Biomarkers Identified for Breast Cancer Subtypes

The 4 identified subtypes by the GO Biological Processes method contain 75, 144, 49, and 236 tumors. We calculated the frequency of a gene in a subtype as the number of tumors in the subtype with mutation in
Table 4 The NMI calculated for subtypes identified by each method compared to the actual subtypes of Breast Cancer, based on 20 different sub-sampled datasets.

<table>
<thead>
<tr>
<th>#</th>
<th>pathway</th>
<th>GO BP</th>
<th>GO BF</th>
<th>Original mutation profiles</th>
<th>NBS</th>
<th>GO BP without normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.115</td>
<td>0.119</td>
<td>0.115</td>
<td>0.103</td>
<td>0.077</td>
<td>0.099</td>
</tr>
<tr>
<td>2</td>
<td>0.103</td>
<td>0.108</td>
<td>0.123</td>
<td>0.070</td>
<td>0.093</td>
<td>0.082</td>
</tr>
<tr>
<td>3</td>
<td>0.089</td>
<td>0.108</td>
<td>0.095</td>
<td>0.072</td>
<td>0.082</td>
<td>0.078</td>
</tr>
<tr>
<td>4</td>
<td>0.070</td>
<td>0.119</td>
<td>0.128</td>
<td>0.065</td>
<td>0.084</td>
<td>0.075</td>
</tr>
<tr>
<td>5</td>
<td>0.099</td>
<td>0.118</td>
<td>0.089</td>
<td>0.072</td>
<td>0.074</td>
<td>0.089</td>
</tr>
<tr>
<td>6</td>
<td>0.093</td>
<td>0.103</td>
<td>0.127</td>
<td>0.095</td>
<td>0.081</td>
<td>0.083</td>
</tr>
<tr>
<td>7</td>
<td>0.076</td>
<td>0.080</td>
<td>0.071</td>
<td>0.095</td>
<td>0.032</td>
<td>0.062</td>
</tr>
<tr>
<td>8</td>
<td>0.084</td>
<td>0.111</td>
<td>0.091</td>
<td>0.074</td>
<td>0.074</td>
<td>0.067</td>
</tr>
<tr>
<td>9</td>
<td>0.098</td>
<td>0.104</td>
<td>0.112</td>
<td>0.070</td>
<td>0.089</td>
<td>0.065</td>
</tr>
<tr>
<td>10</td>
<td>0.114</td>
<td>0.106</td>
<td>0.113</td>
<td>0.088</td>
<td>0.129</td>
<td>0.073</td>
</tr>
<tr>
<td>11</td>
<td>0.101</td>
<td>0.103</td>
<td>0.111</td>
<td>0.065</td>
<td>0.072</td>
<td>0.085</td>
</tr>
<tr>
<td>12</td>
<td>0.124</td>
<td>0.138</td>
<td>0.122</td>
<td>0.081</td>
<td>0.072</td>
<td>0.088</td>
</tr>
<tr>
<td>13</td>
<td>0.108</td>
<td>0.134</td>
<td>0.128</td>
<td>0.090</td>
<td>0.086</td>
<td>0.113</td>
</tr>
<tr>
<td>14</td>
<td>0.102</td>
<td>0.114</td>
<td>0.117</td>
<td>0.078</td>
<td>0.059</td>
<td>0.102</td>
</tr>
<tr>
<td>15</td>
<td>0.114</td>
<td>0.125</td>
<td>0.128</td>
<td>0.098</td>
<td>0.065</td>
<td>0.092</td>
</tr>
<tr>
<td>16</td>
<td>0.088</td>
<td>0.142</td>
<td>0.137</td>
<td>0.077</td>
<td>0.066</td>
<td>0.096</td>
</tr>
<tr>
<td>17</td>
<td>0.093</td>
<td>0.111</td>
<td>0.121</td>
<td>0.097</td>
<td>0.108</td>
<td>0.083</td>
</tr>
<tr>
<td>18</td>
<td>0.067</td>
<td>0.053</td>
<td>0.073</td>
<td>0.081</td>
<td>0.065</td>
<td>0.115</td>
</tr>
<tr>
<td>19</td>
<td>0.094</td>
<td>0.120</td>
<td>0.112</td>
<td>0.074</td>
<td>0.117</td>
<td>0.071</td>
</tr>
<tr>
<td>20</td>
<td>0.111</td>
<td>0.136</td>
<td>0.112</td>
<td>0.088</td>
<td>0.089</td>
<td>0.084</td>
</tr>
<tr>
<td>Ave</td>
<td>0.0971</td>
<td><strong>0.1126</strong></td>
<td>0.1112</td>
<td>0.0817</td>
<td>0.0807</td>
<td>0.0852</td>
</tr>
</tbody>
</table>

that gene. For each subtype, we selected the top 10 highly frequently mutated genes as the set of biomarkers associated with the subtype. The set of biomarkers for each of those subtypes is reported in Table 5. To show the biological interpretability of the biomarkers identified for each subtype, we discuss here the relation between biomarkers identified for subtypes with clinical information about the tumors assigned to the identified subtypes. Genes TP53, PIK3CAA are very important tumor suppressor genes [42]. Mutation in these genes results in poor prognosis in most of the cancers. As the results show, PIK3CAA is selected as a biomarker in all subtypes. But TP53 appears in the list of biomarkers for the second and third subtypes only, suggesting that the second and third subtypes have poorer prognosis compared to other subtypes as they have more tumor suppressor genes mutated. The average survival time for the four identified subtypes are 1947, 1027, 915, and 2056 respectively, which verifies the poor prognosis of the second and the third subtypes as suggested by identified biomarkers. Furthermore, almost all the Triple Negative, HER2, and (ER-, PR-, HER2*) breast cancer tumors, which usually have lower survival time compared to other subtypes [92, 42],
Table 5  Biomarkers associated with Breast Cancer subtypes identified by our GO Biological Processes representation method.

<table>
<thead>
<tr>
<th>Subtype #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genes</td>
<td>PIK3CA,</td>
<td>TP53, PIK3CA,</td>
<td>TP53, PIK3CA,</td>
<td>PIK3CA,</td>
</tr>
<tr>
<td></td>
<td>TTN, MUC16,</td>
<td>TTN, MUC16,</td>
<td>TTN, MUC16,</td>
<td>GATA3,</td>
</tr>
<tr>
<td></td>
<td>USH2A, CDH1,</td>
<td>USH2A, FLG,</td>
<td>USH2A, FLG,</td>
<td>MAP3K1,</td>
</tr>
<tr>
<td></td>
<td>FBN3, RELN,</td>
<td>MUC12, FAT3,</td>
<td>MUC12, FAT3,</td>
<td>TTN,</td>
</tr>
<tr>
<td></td>
<td>SPTA1, FAT3,</td>
<td>MUC4, DNAH17</td>
<td>MUC4, DNAH17</td>
<td>CDH1, MUC16,</td>
</tr>
<tr>
<td></td>
<td>GATA3</td>
<td></td>
<td></td>
<td>CROCCP2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RUNX1,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CSMD1, MUC4</td>
</tr>
</tbody>
</table>

are assigned to in the second and the third identified subtypes, which further verifies the poor prognosis of second and third identified subtypes as suggested by the identified biomarkers.
Chapter 6

Novel Graph-regularized Bi-clique Finding Algorithm For Simultaneous Tumor Stratification and Biomarker Identification

1 We develop a novel graph-regularized bi-clique finding algorithm for tumor stratification in this chapter. Our new tumor stratification algorithm is specifically designed to address the challenges and limitations of existing methods: (1) incorporating prior biology knowledge in gene-gene interaction relationships through a graph regularization term to alleviate the small sample size problem, noise, and extreme heterogeneity in somatic mutation profiles; and (2) simultaneously clustering tumor samples into subtype groups and identifying biomarkers associated with each subtype. We expect by identifying bi-cliques constrained by prior biology knowledge may yield more accurate tumor subtypes as well as robust and personalized biomarkers for corresponding subtypes. Our experimental results using the Uterine and Ovarian cancer data from TCGA [1, 6] demonstrates that our method is superior to the performance of the NBS approach, which we consider as the state-of-the-art method for tumor stratification.

Similar to Chapter 5, we represent somatic mutation profiles data by $A$, which is an $n \times p$-dimensional binary matrix containing profiles for $p$ genes in $n$ available tumors. As in NBS, $A$ can be further smoothed by using gene-gene interaction information to obtain a matrix of non-negative real values. With either this binary or smoothed somatic mutation matrix $A$, we are only interested in identifying bi-clusters (or sub-matrices) in $A$ that contain mostly 1s or large values, but not sub-matrices containing all zeros (or small values), which does not indicate a biologically meaningful tumor subtype. This is different from existing bi-clustering methods, often used to analyze microarray expression profiles, which look for bi-clusters with similar patterns no matter whether they have consistently large or consistently small values. Another major difference in our proposed bi-clique finding algorithm from existing bi-clustering methods is that we only look for a small number of genes that exhibit consistent patterns for each subtype, hoping for biologically more useful and robust biomarkers for consequent disease prognosis and treatment.

---

1This chapter was published in Computational biology and chemistry Vol. 57, pp. 3-11. Permission is included in Appendix B.
In this Chapter, we will take a graph representation of the given somatic mutation data matrix $A$ so that simultaneous tumor stratification and biomarker identification is transformed to a graph-regularized bi-clique finding problem of searching for densely connected sub-graphs in a bipartite graph constrained with prior biology knowledge on gene-gene interaction information (Figure 15A). We note that we may extend the method to analyze more general omics profiles including mRNA expression data by first deriving significant tests to obtain a binary matrix capturing the gene-tumor association relationships [1, 6]. Given somatic mutation data matrix $A$, we first construct a corresponding bipartite graph denoted by $G = (U, V, E)$, where the node set $U = \{u_1, \ldots, u_n\}$ is in one part of the bipartite graph corresponding to available tumor samples; and the node set $V = \{v_1, \ldots, v_m\}$ is in the other part of the graph representing profiled genes. Node $v_i$ is connected to $u_j$ with a weight $A_{i,j}$ (1 or graph-diffused value) if $A_{i,j} > 0$. To achieve tumor stratification, we are looking for a partition of tumor samples with each subset having a corresponding small number of genes that are consistently mutated or have strong association. This can be achieved by finding densely connected sub-bipartite-graphs (relaxed bi-cliques) in the given bipartite graph. Here we propose a novel approach by extending the bi-clique finding method proposed in [64] (discussed in Chapter 2) to incorporate
the gene-gene interaction network information. As we are extending their algorithm, before proposing our approach, we discuss their method in more detail.

6.1 Bi-clique Finding Algorithm [64]

The authors in [64] have proposed a bi-clique finding approach based on a relaxed integer programming formulation based on the Motzkin and Straus theorem [93] to efficiently find large bi-cliques. With the adjacency matrix $A$ of the bipartite graph $G$, a maximal bi-clique can be computed by finding the solution to the following optimization problem:

$$
\begin{align*}
\max_{s, g} & \quad s^T A g \\
\text{s.t.} & \quad s_i > 0 \text{ for } i \in \{1..n\}; \\
& \quad g_i > 0 \text{ for } i \in \{1..p\}; \\
& \quad \sum_{i=1}^n s_i^\alpha = 1; \\
& \quad \sum_{i=1}^m g_i^\beta = 1,
\end{align*}
$$

in which $s$ and $g$ are two relaxed binary vectors, reflecting whether the corresponding sample ($s$) or gene ($g$) is selected in the final bi-clique. Given a relaxed solution $(s^*, g^*)$ to the above optimization problem, the nodes in the left and right parts of the maximal bi-clique are indicated by non-zero elements in $s^*$ and $g^*$ respectively. The parameters $1 < \alpha \leq 2$ and $1 < \beta \leq 2$ favor different shapes of bi-cliques: when $\alpha > \beta$, the method tends to find bi-cliques with a larger number of nodes from the left side than from the right side; and vice versa. Also, setting $\alpha, \beta$ to a value close to 1 helps find very tight (exact) bi-cliques and setting $\alpha, \beta$ to larger values leads to more relaxed approximate bi-cliques with some missing edges. The latter is more practical in most of the bioinformatics applications such as tumor stratification since densely connected sub-bipartite-graphs are more realistic due to underlying cellular mechanisms as well as measurement noise. Note that the optimization problem is non-convex and multiplicative update rules, similar to popular solutions to NMF problems, have been derived to solve the problem to local maxima:

$$
\begin{align*}
s_i^{(t+1)} &= \left(s_i^{(t)} \frac{(Ag_i^{(t)})}{s^{(t)} A g^{(t)}}\right)^{1/\alpha} \\
g_i^{(t+1)} &= \left(g_i^{(t)} \frac{(A^T s_i^{(t)})}{s^{(t)} A g^{(t)}}\right)^{1/\beta}
\end{align*}
$$
Starting from a random feasible solution: \( s_0 \) and \( g_0 \), it has been proven that the derived multiplicative update rules converge to a point satisfying KKT conditions for local maxima [64]. To find multiple bi-cliques they suggest to iteratively delete the first bi-clique from the bi-partite graph \( G \) and perform the above algorithm to find the next bi-clique. The iterative sequential procedure can be repeated until we find a desired number of bi-cliques.

### 6.2 Graph-regularized Bi-clique Finding

Based on the previous bi-clique finding algorithm, we further incorporate prior knowledge on gene-gene interaction relationships to achieve more robust and better tumor stratification (Figure 15B). As previously discussed in Chapters 1 and 5 many complex diseases may arise from aberrant changes to a set of pathways or functional modules rather than a specific set of individual genes [94, 6, 95, 96]. Therefore, incorporating prior biology knowledge in gene-gene interaction through a graph-regularized formulation may further improve tumor stratification performance and can help overcome the heterogeneity in somatic mutation profiles. We represent the available gene-gene interaction relationships by an undirected graph \( H \) with the adjacency matrix, \( M_{p \times p} \) (Figure 15A). The corresponding Laplacian matrix \( L \) can be computed as \( L = D - M \) where \( D \) is the diagonal matrix with node degrees on diagonal entries \( (D_{i,i} = \sum_j M_{i,j}) \).

Recently, researchers have used a graph Laplacian to incorporate network information mostly for network-constrained regularization of model parameters [47, 6, 97] as the Laplacian matrix is positive semi-definite which makes it a suitable choice for optimization. The NBS approach that was discussed in Chapters 1 and 2 also uses the Laplacian matrix to incorporate the network information for NMF-based clustering to integrate information from local network neighborhoods. For our tumor stratification problem, instead of only looking for dense sub-bipartite-graphs containing tumor samples that have mutations in same set of genes, we would like to use the interaction information to find sub-bipartite-graphs that contain tumors that have mutations targeting neighboring genes in the network as well. In order to achieve that, we introduce a graph regularization term:

\[
g^T L g = \sum_{i=1}^{p} \sum_{j=1}^{p} M_{i,j} (g_i - g_j)^2,
\]

in which \( g \) again denotes the \( p \)-dimensional assignment vector for genes. Minimizing this regularization term explicitly enforces that the value of \( g^T L g \) gets small and the neighboring nodes in the gene interaction graph have similar corresponding values in \( g \).
To extend the bi-clique finding algorithm in [64], we add the above graph regularization term to have our graph-regularized bi-clique finding problem formulation:

\[
\max_{s, g} \quad s^T A g - g^T L g
\]

\[
\text{s.t.} \quad s_i > 0 \text{ for } i \in \{1..n\};
\]

\[
g_i > 0 \text{ for } i \in \{1..p\};
\]

\[
\sum_{i=1}^{n} s_i^\alpha = 1;
\]

\[
\sum_{i=1}^{m} g_i^\beta = 1,
\]

in which free parameters \(\alpha\) and \(\beta\) again control the norm of the final solution. Setting the value of \(\alpha\) or \(\beta\) equal to 1 (or close to 1) forces its corresponding solution vector \(s\) or \(g\) to be sparse with a large number of zero elements as discussed in [73]. On the other hand, for larger values of \(\alpha\) or \(\beta\), the corresponding solution vectors will be less sparse. By using different values of \(\alpha\) or \(\beta\), we have the control over the size of subtype groups and the number of biomarkers.

We have derived the following updating rules that have the convergence guarantee to the KKT points of the above optimization problem:

\[
s_i^{(t+1)} = \left( \frac{s_i(Ag)_i}{s^T Ag} \right)^{1/\beta};
\]

\[
g_i^{(t+1)} = \frac{g_i \left( (s^T A)_i + 2 (Mg)_i + g_i^{\alpha-1} (2g^T D g) \right)}{2 (g^T D)_i + g_i^{\alpha-1} (s^T Ag + 2g^T Mg)},
\]

where in the righthand side of the equations, \(s\) and \(g\) are the values at iteration \(t\) and \(L = D - M\). We have proven the following theorem which guarantees that the above updating rules converges to a local maximum of the optimization problem (6.1).

**Theorem 6.1** Starting from an arbitrary feasible point \(s^{(0)}\) and \(g^{(0)}\), the iterative updates based on the updating rules in equations (6.2) and (6.3) converge to a point that satisfies KKT conditions for the optimization problem (6.1).
Proof. We first write the KKT conditions for the optimization problem (6.1):

• Stationary:

\[
(s^T A)_i - 2(g^T L)_i - \lambda_1 \alpha g_i^{\alpha-1} + \mu_i = 0, \text{ for } i = 1..p; \quad (6.4)
\]
\[
(Ag)_i - \lambda_2 \beta s_i^{\beta-1} + \gamma_i = 0, \text{ for } i = 1..n; \quad (6.5)
\]

• Primal feasibility:

\[
s_i \geq 0, \text{ for } i = 1..n; \quad (6.6)
\]
\[
g_i \geq 0, \text{ for } i = 1..p; \quad (6.7)
\]
\[
\sum_{i=1..n} s_i^{\beta} = 1; \quad (6.8)
\]
\[
\sum_{i=1..p} g_i^{\alpha} = 1; \quad (6.9)
\]

• Dual feasibility:

\[
\mu_i \geq 0, \text{ for } i = 1..p; \quad (6.10)
\]

• Complementary slackness:

\[
\mu_i g_i = 0, \text{ for } i = 1..p; \quad (6.11)
\]
\[
\gamma_i s_i = 0, \text{ for } i = 1..n. \quad (6.12)
\]

Assuming that the updating rules in equations (6.2) and (6.3) converge to points \( s^* \) and \( g^* \), we have:

\[
s_i^* = \left( \frac{s_i^*(Ag^*)_i}{s^*Ag^*} \right)^{1/\beta}; \quad (6.13)
\]
\[
g_i^* = \frac{g_i^* \left( (s^*A)_i + 2(Mg^*)_i + g_i^{\alpha-1}(2g^*Dg^*) \right)}{2(g^*D)_i + g_i^{\alpha-1}(s^*Ag^* + 2g^*Mg^*)}. \quad (6.14)
\]

The equation (6.13) implies:

\[
s_i^* \beta s^* Ag^* - s_i^*(Ag^*)_i = 0. \quad (6.15)
\]
Therefore $\sum_{i=1..n} \left(s_i^* \beta s^* A g^* - s_i^* (A g^*)_i\right) = 0$. Hence we have:

$$\sum_{i=1..n} s_i^\beta = 1.$$ 

Starting with non-negative $s^{(0)}$, the updating rule always keeps $s^{(t)}$ non-negative, since all elements in $A$ are non-negative. Consequently we have primal feasibility for $s^*$.

Assuming $\lambda_2 = \frac{s^* A g^*}{\beta}$ and with the equation (6.15), we have

$$s_i^* \beta \lambda_2 - s_i^* (A g^*)_i = 0,$$

which implies that:

$$s_i^* \left(s_i^* - 1 \beta \lambda_2 - (A g^*)_i\right) = 0.$$  \hspace{1cm} (6.16)

Let $\gamma_i = s_i^* - 1 \beta \lambda_2 - (A g^*)_i$, we have both complementary slackness and stationary conditions satisfied. Thus all the KKT conditions are satisfied in $s^*$.

Furthermore, Equation (6.14) implies that

$$g_i^* \left(2(g^* D)_i + g_i^{\alpha-1} (s^* A g^* + 2g^* M g^*) - (s^* A)_i - 2(M g^*)_i - g_i^{\alpha-1} (2g^* D g^*)_i\right) = 0.$$ 

Hence,

$$g_i^* \left(2(g^* L)_i - (s^* A)_i + g_i^{\alpha-1} (s^* A g^* - 2g^* L g^*)_i\right) = 0.$$ \hspace{1cm} (6.17)

Summing over $i$ gives $\sum_{i=1..n} g_i^* \left(2(g^* L)_i - (s^* A)_i + g_i^{\alpha-1} (s^* A g^* - 2g^* L g^*)_i\right) = 0$, which implies that $\sum_{i=1..n} g_i^\alpha = 1$. Starting with non-negative $g^{(0)}$, the corresponding updating rule always keeps $g^{(t)}$ non-negative since all elements in $A$ and $D$ and $M$ are non-negative. Hence, we also have primal feasibility for $g^*$.

Finally, let $\lambda_1 = \frac{s^* A g^* - 2g^* L g^*}{\alpha}$. Based on (6.17), we have

$$g_i^* \left(2(g^* L)_i - (s^* A)_i + g_i^{\alpha-1} \alpha \lambda_1\right) = 0.$$ \hspace{1cm} (6.18)

With $\mu_i = 2(g^* L)_i - (s^* A)_i + g_i^{\alpha-1} \alpha \lambda_1$, we have both complementary slackness and stationary conditions
satisfied. Thus all the KKT conditions are satisfied in $g^*$ based on our derived multiplicative updating rules.

Solving the above optimization problem results in one densely connected sub-bipartite-graph. In order to get all the potential subtypes for tumor stratification, we iteratively repeat the procedure of applying the above method and sequentially remove the corresponding nodes in the identified sub-bipartite-graph from $G$ and $M$ until there is no edge or node in either side of the bipartite graph $G$.

6.3 Results and Discussion

We have evaluated the performance of our graph-regularized bi-clique finding algorithm and compared it to the state-of-the-art NBS method proposed in [6] on cancer datasets from TCGA [1, 6]. In order to have a fair comparison with NBS, we have used exactly the same profiled somatic mutation data for the Uterine and Ovarian cancer datasets and followed the data processing pipeline provided in [6]. Tumor stratification results by both our proposed method and NBS have been evaluated based on both histological tissue subtypes and available survival data.

6.3.1 Datasets and Pre-processing

The somatic mutation profiles for Uterine and Ovarian cancers from TCGA and the gene-gene interaction information based on the STRING protein interaction network were obtained from the on-line resources provided by [6]. The uterine cancer dataset contains the mutation profiles of 248 patient tumors for 17,968 genes. The somatic mutation data for Ovarian cancer were collected from 356 patients for 9,850 genes. As done in NBS, only genes with interaction(s) in the gene interaction network were used for tumor stratification. After removing genes that are not in the STRING database, we have 10,630 and 9,850 genes in total for Uterine and Ovarian cancers in our experiments, respectively. We follow the performance evaluation in [6] to validate the identified tumor subtypes by both methods based on the observed phenotypic and clinical information. The survival data for Ovarian cancer was downloaded from the website https://www.synapse.org/#!Synapse:syn300013, which was used to perform survival analysis for all the identified subtypes. However, as explained in [6], survival analysis is not possible for Uterine cancer due to low mortality rates in the dataset. For its performance evaluation, we have used curated histological tissue types and tumor grades as in [6]. The corresponding phenotypic information including histological types and tumor grades also were downloaded from the website for the paper [6]. To be specific, all the tumor
samples were originally categorized into two groups based on tumor grades: high grade (3) and low-grade (1,2); and three groups according to the following histological types: endometrioid type, serous adenocarcinoma, and others. By aggregating these information, samples for Uterine cancer have been categorized into 6 groups in total: endometrioid type-high grade, endometrioid type-low grade, serous adenocarcinoma-high grade, serous adenocarcinoma-low grade, other-high grade, other-low grade. In the current dataset, only five unique groups exists since there was no tumor samples in the serous adenocarcinoma-low grade category.

6.3.2 Graph-regularized Bi-clique Finding Pipeline

The original somatic mutation data is provided as a binary matrix $A_{n \times p}$ containing the somatic mutation profiles of $p$ genes for $n$ samples. The gene-gene interaction information is provided as another binary matrix $M_{p \times p}$ which contains the prior knowledge about the interactions among $p$ genes. We first pre-process these two matrices as done in [6]: the interaction matrix $M$ is first normalized by dividing each column by adding all the elements in that column; the somatic mutation matrix $A$ is smoothed over the interaction network by a network diffusion method introduced in [46] based on the following recursive formula $A_{t+1} = \gamma A_t M' + (1 - \gamma) A_0$, in which $A_0$ is the original somatic mutation matrix $A$ until convergence to obtain the desired diffused somatic mutation matrix $A_\infty$. The motivation is to diffuse the influence of mutation effects to the neighboring genes in the network and the tuning parameter $\gamma$ controls the depth of influence. In our experiments, the value of $\gamma$ is set to 0.7 and we iterate as long as $A_{t+1} - A_t < 1e - 6$ as done in NBS.

For our graph-regularized bi-clique finding, the parameters $\alpha$ and $\beta$ are set to 1 and 2, corresponding to $L_1$- and $L_2$-norm regularization, respectively. The $\alpha$ is set to 1 to enforce sparse solutions as the number of genes in the study is large and we want to select only a small number of them for each tumor subtype as typically required for biomarker identification. For tumor stratification, we have set $\beta$ to 2 without enforcing sparse solutions as we have a small number of tumor patients and hope for solutions with each patient belonging to one potential tumor subtype. We repeat the multiplicative updating rules until the difference between two resulting objective function values in (6.1) for two consecutive iterations becomes less than $1e - 6$ or up to 200 iterations. After convergence, we select tumor samples and genes into the corresponding subtypes as follows: given a solution $s^*$, we first sort the values and then select the top $l$ elements in the vector as tumor samples belonging to the subtype, where $l$ is the largest number for which $\sum_{i=1}^{l} (s_i^*)^2 < 0.99$ to cover 99% of the $L_2$-norm of $s^*$. We adopt the same strategy for biomarker identification of the corresponding subtype based on the converged values in $g^*$ to select the top $m$ elements.
that cover 99\% of the $L_1$-norm of $g^*$: $\sum_{i=1}^{m} (g^*_i) < 0.99$. Generally, we can adopt this strategy for any vector $x$ obtained by our algorithm with the corresponding $L_r$-norm. Depending on the norm of the solution vector $x$, we may have a small or large number of selected elements relative to the dimension of $x$. When $r$ is close to 1, the number is small as the solution is sparse while for larger $r$ close to 2, we have more selected elements. The number of genes selected by our algorithm depends on the threshold value. One can select a smaller value if a smaller number of genes is desired as biomarkers or vice versa. After identifying each subtype group and associated genes, we remove the corresponding samples and genes and repeat the procedure until no tumor remains. Furthermore, the values in the solution vector $g^*$ can be considered as an association measure of the corresponding genes in each subtype. Based on this association measure, one can further sort the genes and select a desired number of important genes as we later do for the analysis of selected genes in Table 3.

The algorithm starts by finding sub-bipartite-graphs with reasonable numbers of genes and samples. However, after a few iterations, the remaining bipartite graph will become extremely sparse with no significant subnetwork structure. In this situation, the algorithm tends to find small sub-graphs within which vertices are sparsely connected. We perform a post-processing step so that the corresponding tumor samples can be considered as outliers as they do not display significantly consistent mutation profiles. Currently, we ignore the clusters with only one sample in the post-processing step of our algorithm. Further post-processing of those small clusters (outliers) to either remove or combine them with similar clusters might help improve the interpretation of the resulting clusters. In the current study, we have intentionally avoided polishing results for the purpose of fair comparison with the competing computational method. The summarized pipeline of our method is illustrated in Figure 15C.

### 6.3.3 Performance Evaluation With Uterine Cancer

To measure the performance, we compute the Normalized Mutual Information (NMI) (as explained in Section 5.2.3 of Chapter 5) between tumor stratification results and curated tumor subtypes based on the known histological types and tumor grades for performance evaluation with the Uterine cancer dataset. As done in [6], we perform a random sub-sampling of the original cancer dataset with a rate of 0.8, in which we randomly select 80\% of the tumor samples and genes from the original somatic mutation data. For each sub-sampled dataset, we implemented our graph-regularized bi-clique finding algorithm for tumor stratification. For the NBS implementation, as we need to specify the desired number of subtype groups,
Table 6  Performance comparison based on curated subtypes for Uterine cancer. The table shows the results for each randomly sub-sampled data for 20 times (rows). The first column shows the Normalized Mutual Information (NMI) based on tumor stratification from our graph-regularized bi-clique finding algorithm. The final numbers of tumor subtype groups are provided in parentheses for each sub-sampled data. The second column shows the performance of the NBS method by specifying the number of subtypes for NBS based on what is discovered by our method for each subsampled data. Columns 3-5 shows the performance by NBS by specifying 2, 5, and 8 subtypes. The last column gives the best Normalized Mutual Information by NBS among the results from all different numbers of subtypes (columns 2-5).

<table>
<thead>
<tr>
<th>Our method’s performance (# of subtypes discovered)</th>
<th>NBS performance with different subtype number: # of subtypes discovered by our method</th>
<th>Best NBS performance</th>
<th>Average: 0.32</th>
<th>Average: 0.29</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.21 (7)</td>
<td>0.20</td>
<td>0.08 0.26</td>
<td>0.08 0.28 0.26</td>
<td>0.28</td>
</tr>
<tr>
<td>0.35 (13)</td>
<td>0.19</td>
<td>0.10 0.09 0.12</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>0.33 (13)</td>
<td>0.23</td>
<td>0.11 0.16 0.20</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>0.29 (11)</td>
<td>0.30</td>
<td>0.12 0.27 0.27</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>0.33 (10)</td>
<td>0.28</td>
<td>0.15 0.23 0.23</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>0.35 (14)</td>
<td>0.26</td>
<td>0.15 0.27 0.28</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>0.30 (11)</td>
<td>0.27</td>
<td>0.12 0.26 0.28</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>0.31 (12)</td>
<td>0.29</td>
<td>0.10 0.22 0.25</td>
<td>0.29</td>
<td></td>
</tr>
<tr>
<td>0.38 (15)</td>
<td>0.38</td>
<td>0.20 0.32 0.32</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>0.30 (7)</td>
<td>0.26</td>
<td>0.10 0.23 0.25</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>0.36 (12)</td>
<td>0.25</td>
<td>0.14 0.32 0.30</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>0.29 (15)</td>
<td>0.32</td>
<td>0.11 0.23 0.25</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>0.30 (11)</td>
<td>0.32</td>
<td>0.13 0.32 0.33</td>
<td>0.33</td>
<td></td>
</tr>
<tr>
<td>0.31 (11)</td>
<td>0.32</td>
<td>0.14 0.21 0.28</td>
<td>0.32</td>
<td></td>
</tr>
<tr>
<td>0.31 (12)</td>
<td>0.26</td>
<td>0.14 0.30 0.28</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>0.33 (11)</td>
<td>0.23</td>
<td>0.12 0.18 0.25</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>0.32 (13)</td>
<td>0.28</td>
<td>0.13 0.27 0.27</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>0.38 (12)</td>
<td>0.30</td>
<td>0.09 0.26 0.29</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>0.34 (11)</td>
<td>0.30</td>
<td>0.17 0.27 0.31</td>
<td>0.31</td>
<td></td>
</tr>
<tr>
<td>0.31 (12)</td>
<td>0.23</td>
<td>0.09 0.25 0.25</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>
we tried NBS with the total subtype number specified at 2, 5, 8, and the discovered subtype number by our algorithm. In order to evaluate and compare the performance by NMI with the estimated statistical significance, we repeated sub-sampling 100 times and computed the NMI for all cases with NBS running four times with different subtype numbers. We also report the maximum NMI-based score from four NBS runs as the NBS performance. The average of those 100 sub-sampled runs for each method is then reported as their final performance indices, respectively. The results for randomly selected 20 sub-sampled runs as well as their average values are provided in Table 6. As the results show, our method outperforms NBS in a larger fraction of sub-sampled runs. The results are statistically significant for 100 sample runs with p-value $< 6.35e - 08$ based on the two-sample t-test. The selected 20 runs in Table 6 are also tested and have p-value = 0.0313. We also performed the above procedure using two other performance measures: the $\chi^2$ statistic based on a contingency table [98] and the Rand Index [99], as done by NBS [6]. The results show that our method performs significantly better with p-values = 0.0441 and 0.0765 for 20 selected runs respectively. The obtained results are expected as we believe that simultaneous tumor stratification and personalized biomarker identification can help find tumor subtypes more accurately. Furthermore, to verify the robustness of our method, we checked the consistency of identified subtypes among all 20 different sub-sampled runs. We calculated the $\chi^2$ statistic based on a contingency table [98] to test the consistency of clustering for all 190 possible pairs of results. All the p-values obtained were less than 1.24e-20, which shows that the obtained results for 20 different sub-sampled runs of data are mutually consistent with each other verifying that the method is robust to subsampling.

**6.3.4 Performance Evaluation with Ovarian Cancer**

We further evaluated the performance of our algorithm based on actual clinical data. Specifically, we analyzed survival time available for Ovarian cancer data based on the identified tumor subtypes by both our bi-clique finding algorithm and NBS using the R package *survival* [100]. After computationally discovered subtypes, we took the corresponding survival data of Ovarian cancer samples in each subtype group to draw a corresponding Kaplan-Meier survival curve (K-M plot) [101]. Our graph-regularized bi-clique finding algorithm discovers 8 subtype groups in Ovarian cancer data, whose corresponding K-M plots are illustrated in Figure 16A with four of them marked in different colors. To compare with NBS, we also draw the corresponding K-M plots in Figure 16B based on the NBS implementation of finding 8 subtype groups. We compute log-rank statistics for both stratification results and the corresponding p-values are also given in
Figure 16. Performance comparison with Ovarian cancer by survival analysis. A. Kaplan-Meier survival plots for NBS subtypes; B. Kaplan-Meier survival plots for our method that gives subtypes with significantly different survival curves.

Based on the log-rank statistics [102], our new algorithm is able to find tumor subtypes with significantly different survival time while the NBS method fails to do so. We note that in the original NBS paper [6], an extra consensus clustering step was implemented to obtain the final tumor stratification results by sub-sampling the given data 1,000 times, which further improved the performance. Here, our aim is to compare our graph-regularized bi-clique finding algorithm and the NBS algorithm by running the stratification step only once. We believe that consensus clustering would also improve the performance of both methods. Moreover, as our algorithm automatically decides the number of subtype groups, we cannot directly adopt the same consensus clustering procedure in [6] for fair comparison. Hence, we have designed the following experiments, as already done with the Uterine cancer dataset, to verify the effectiveness of our method by comparing the performance for each individual sub-sampling of the dataset.
We performed a random sub-sampling procedure on the Ovarian cancer dataset 20 times by randomly selecting 80% of genes and samples based on the given somatic mutation data. We implemented both our algorithm and NBS with four runs at the specified subtype number 2, 5, 8, and the number of discovered subtypes by our algorithm. We performed survival analysis based on the corresponding stratification results and calculated the log-rank statistics and the corresponding p-values as the performance index. The smaller the log-rank statistics p-value, the more significantly different are the survival curves between discovered tumor subtypes. For each sub-sampled run, the minimum p-value for all four NBS runs is considered as the best NBS performance. Table 7 shows the results for 20 sub-sampled runs. Again, these results with Ovarian cancer further validate that our method consistently outperforms the state-of-the-art NBS method in tumor stratification with strong biomedical evidence based on survival analysis.

We further list the associated genes with each of the eight subtypes identified by our method on the original somatic mutation data as shown in Figure 16. Our method has identified eight subtypes containing 71, 101, 48, 29, 16, 11, 26, and 23 tumor samples with 181, 205, 204, 227, 372, 533, 798, and 3186 associated genes.

Table 7 Performance comparison with Ovarian cancer. The table shows the results for 20 different sub-samples of data (rows). The first column gives the p-values of log-rank statistics based on our method with the number of discovered subtypes in parentheses (different for different sub-samples). The second column shows the performance of the NBS method with the desired subtype number as that discovered by our method for each sub-sample. Columns 3-5 provide the performance of NBS with 2, 5, and 8 subtypes. The last column is the best NBS performance among NBS runs with all different numbers of subtypes (from columns 2 to 5).

<table>
<thead>
<tr>
<th>Our method’s performance (# of subtypes discovered)</th>
<th>NBS performance with different subtype numbers: # of subtypes discovered by our method</th>
<th>Best NBS performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.038102 (7)</td>
<td>0.487 (2)</td>
<td>0.03414</td>
</tr>
<tr>
<td>0.001163 (7)</td>
<td>0.682 (2)</td>
<td>0.280</td>
</tr>
<tr>
<td>3.242e-08 (6)</td>
<td>0.043 (2)</td>
<td>0.006</td>
</tr>
<tr>
<td>3.257e-07 (6)</td>
<td>0.042 (2)</td>
<td>0.042</td>
</tr>
<tr>
<td>0.035431 (6)</td>
<td>0.689 (2)</td>
<td>0.467</td>
</tr>
<tr>
<td>0.289015 (7)</td>
<td>0.039 (2)</td>
<td>0.003</td>
</tr>
<tr>
<td>0.228358 (7)</td>
<td>0.079 (2)</td>
<td>0.058</td>
</tr>
<tr>
<td>4.410e-09 (6)</td>
<td>0.133 (2)</td>
<td>0.107</td>
</tr>
<tr>
<td>2.220e-16 (6)</td>
<td>0.536 (2)</td>
<td>0.013</td>
</tr>
<tr>
<td>0.286360 (6)</td>
<td>0.183 (2)</td>
<td>0.161</td>
</tr>
<tr>
<td>0.000000 (6)</td>
<td>0.103 (2)</td>
<td>0.037</td>
</tr>
<tr>
<td>0.151490 (6)</td>
<td>0.151 (2)</td>
<td>0.102</td>
</tr>
<tr>
<td>2.344e-06 (7)</td>
<td>0.064 (2)</td>
<td>0.026</td>
</tr>
<tr>
<td>2.048e-09 (6)</td>
<td>0.125 (2)</td>
<td>0.093</td>
</tr>
<tr>
<td>1.288e-09 (7)</td>
<td>0.221 (2)</td>
<td>0.033</td>
</tr>
<tr>
<td>0.039827 (6)</td>
<td>0.035 (2)</td>
<td>0.029</td>
</tr>
<tr>
<td>0.000000 (6)</td>
<td>0.189 (2)</td>
<td>0.059</td>
</tr>
<tr>
<td>4.799e-07 (6)</td>
<td>0.152 (2)</td>
<td>0.046</td>
</tr>
<tr>
<td>9.947e-08 (6)</td>
<td>0.065 (2)</td>
<td>0.013</td>
</tr>
<tr>
<td>3.906e-08 (7)</td>
<td>0.006 (2)</td>
<td>4e-4</td>
</tr>
</tbody>
</table>
For each subtype, we rank its associated genes based on the solution vector $g^*$, as described earlier. The top 20 genes associated with each subtype are listed in Table 8. We have also performed gene set enrichment analysis using the GSEA software [78] to find important pathways related to each subtype based on those genes. Some important pathways significantly enriched for each subtype are also listed in Table 8. In the following, we briefly state their possible relationships with Ovarian cancer and its stratification. For the discovered subtype 1, genes involved in GPCR downstream signaling have been proven to contribute to cell proliferation [103]. Genes involved in Immune System in subtype 2 have been suggested to be involved in Ovarian cancer prognosis and therapy [104]. For this subtype, NF-κB Signaling Pathway has a specific role in the development and maintenance of Ovarian cancer [105]. The genes in the same pathway have also been found to have significantly different expressions between their discovered subtypes in NBS [6]. Regulation of actin cytoskeleton for subtype 3 is frequently activated in ovarian cancer [106]. Our discovered subtype 4 is enriched with the genes involved in Pyruvate metabolism and Citric Acid (TCA) cyclen. Pyruvate uptake recently has been shown to be increased in highly invasive ovarian cancer cells [107]. Further investigating the K-M plots in Figure 16B, the tumor samples in subtype 4 show worse overall survival compared to subtype 2, which is again consistent with the reported results in [6]. In Retinol metabolism for subtype 5: Vitamin A (retinoid) metabolism has been found to be impaired in human ovarian cancers [108]; The associated genes in subtype 6 are involved in Signaling by Rho GTPases. Up-regulation of small GTPases, RhoA and RhoC, has been shown to be associated with tumor progression in ovarian carcinoma [109]. Finally, for subtype 8, overexpression of VEGF by ovarian cancer cells has been conjectured to be a major mediator of angiogenesis in this tumor type [110]. In addition, Eukaryotic protein translation can be recently considered as a target for inhibition of ovarian cancer growth [111]. With more thorough follow-up analyses with other available high-throughput data from TCGA as similarly done in [6], we believe that more biology insights into ovarian cancer can be derived for effective personalized prognosis and treatment in the future.
Table 8  Important identified genes for each subtype based on Ovarian cancer dataset together with the enriched pathways.

<table>
<thead>
<tr>
<th>#</th>
<th>Genes</th>
<th>Pathways</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OR1D2, OR52N1, OR5AC2, OR51E2, OR2G3, OR52R1, OR4C46, OR56A4, OR14C36, OR7D2, OR8U1, OR4D5, OR10AD1, OR10R2, OR1J1, OR5K4</td>
<td>Genes involved in Olfactory Signaling Pathway</td>
</tr>
<tr>
<td>2</td>
<td>PSMB6, PSNMC3, RELA, RREB1, VPS52, TLR8, KIF18, ZBTB5, SPEG, RPL5, ZNAS1, SRFPA10, SSH1, JPH1, JPH2</td>
<td>Genes involved in GPCR downstream signaling</td>
</tr>
<tr>
<td>3</td>
<td>AVP, H1R, DIAPH2, FOXO a, CAM, PAK1, UCP3, MAP4K4, POM121, CDC42, EP1, RBP1, ZBTB43, MAP3K3, ANGEL1, NARF, RPL35, ZHX1, SERPINA10, SSH1, JPH1, JPH2</td>
<td>Genes involved in Immune System</td>
</tr>
<tr>
<td>4</td>
<td>CAG, CDC20, TMB3, HMB, CTOR, CLA, MKR, P2RY1, PCK1, FGTA, HEG3, LSC, SLC16A1, UNG, VN2, SLC1A3, SLC, LSC4A2, T1, T1A, PAK3, FHB, A3X1</td>
<td>Genes involved in Pynnuvate metabolism and Citric Acid (TCA) cyclen</td>
</tr>
<tr>
<td>5</td>
<td>RDR1, HDI, H1A3, LG3R5, PROM1, T1XN1, LST2, ATP5, YY1, APA1, KRO1, BCL2L4, WRD61, YFP5, U18BP1, ABCB5, TGM</td>
<td>Retinol metabolism</td>
</tr>
<tr>
<td>6</td>
<td>RHG6, RHOG, HCCHRM, LEC72, MTR, MPK3, GPHN1, PSO9, SRL, H1A4, ARHGA26, MCF2L, GMIFD, DIP4C1B, ARHGA22, ARA32, SYDE2, ARHGA19, ARHGA18, ARHGA33</td>
<td>Genes involved in Signaling by Rho GTPases</td>
</tr>
<tr>
<td>7</td>
<td>ALDH1B1, CDIP, CYP1L2, HST1, HBD2, NOPE, PNAX2, P4PSAR, RRGAGA, RAD17, D1K581, MICAL2, FARP2, SEMA3A, DPYSL, LCO3, PLXN4, DPYSL, PNPT1, PNAX4, PPARC1B</td>
<td>Genes involved in SEMA3A-Plexin repulsion signaling by inhibiting Integrin adhesion</td>
</tr>
<tr>
<td>8</td>
<td>ATN1, EIF1AX, EIF2B1, MAZ, PGLI, PKR81, SLC35A2, EIF2B3, EIF2B2, EIF2B5, EIF2C2, RALY, KDM8C, USF2, UGGT1, UGTP1, EIF2A, CDCP2</td>
<td>Genes involved in VEGF, Hypoxia, and Angiogenesis</td>
</tr>
</tbody>
</table>

* These pathways are obtained by performing GSEA based on all genes identified in their corresponding subtype instead of the top 20 genes.
Chapter 7
Conclusion

In this dissertation we proposed several novel computational approaches for biomarker identification for complex and heterogeneous diseases. The developed computational approaches are able to improve the performance of existing methods by addressing several problems in dealing with high-throughput biomedical data with specific focuses on dealing with several infamous analytic issues, such as small sample size, non-linearity, and disease heterogeneity. In this chapter, we conclude the dissertation by providing a summary of the computational methods proposed and discuss the possible directions for future research in this area.

7.1 Summary

In Chapter 3 of this dissertation, we have comprehensively reviewed a set of methods for estimating pairwise interactive effects between continuous random features for biomedical applications. To further improve the performance, we proposed new methods by exploring non-linear and/or supervised modeling. We have shown that we can effectively adopt existing supervised modeling approaches to improve the performance of information theoretic based methods by either using them to perform the supervised quantization of continuous features or by directly using posterior probability estimates for interactive effect estimation. We have tested all methods with different levels of complexity and for different sample sizes. The results show that more complex non-linear modeling approaches like KNN are more accurate in detecting the actual interactive effects. However, they require a reasonably large sample size to avoid overfitting. For small sample size scenarios, quadratic modeling such as QDA with controlled model complexity can perform better and more robustly. The results also show that our classification accuracy based methods empirically outperform the other methods at the same level of model complexity.

In Chapter 4, we proposed a novel network-based feature ranking method that significantly improves the traditional biomarker identification by considering the interactive effect between features. We formulated the problem as finding cliques in the synergy network, in which the information about individual effects
and interactive effects between features are integrated. We have proposed a novel graph spectral algorithm that leads to an effective feature ranking algorithm considering the interaction between features. The comprehensive results based on simulated datasets have shown that our network-based feature ranking can help identify more predictive biomarkers than traditional individual-based feature ranking. Furthermore our results based on the breast cancer and type 1 diabetes datasets show that considering interactive effects is actually useful in identifying more accurate biomarkers. We show that the biomarkers identified for breast cancer are biologically interpretable as they have significant intersections with important pathways related to breast cancer metastasis.

In Chapter 5, we have focused on clustering somatic mutation profiles which are binary vectors with some undesirable characteristics including extreme sparseness and heterogeneity. We have proposed to use valuable prior biological knowledge in gene-set datasets (pathways and GO annotations), to better characterize the distance between mutation profiles which leads to more accurate clustering and consequently better biomarker identification associated with the subtypes. We have comprehensively tested several methods based on different types of gene-sets and compared them with a state-of-the-art method for tumor stratification. The results show that the information in the Biological Processes aspect of GO annotations provides the best clustering performance in identifying the actual subtypes of Breast Cancer.

In Chapter 6, we focused on bi-clustering somatic mutation profiles. We developed a new graph-regularized bi-clique finding algorithm for tumor stratification, in which we simultaneously find tumor subtypes and their associated biomarkers, taking into account the prior biology knowledge provided in a gene-gene interaction network. Tumor stratification is computationally formulated as a graph optimization problem of finding densely connected sub-bipartite-graphs in a bi-partite graph representation of the given somatic mutation data. We solve this non-convex optimization problem to local optima by deriving novel multiplicative updating rules with a convergence guarantee. Our experimental results based on both the Uterine cancer and Ovarian cancer datasets from TCGA have shown that our method can achieve better performance compared to the state-of-the-art method for tumor stratification and is able to find biologically meaningful biomarkers associated with the subtypes.

7.2 Future Research Directions

The main focus of Chapter 3 was to develop methods for better estimation of the interactive effects of two continuous random variables as features on a binary outcome. Sometimes data sets contain more di-
verse data types, especially when we have a set of genetic variables which are mostly continuous or binary variables and another set of environmental variables that might be categorical or ordinal. So, one potential direction for future research in this area to develop methods for estimating interactive effects between two variables of different types. Another natural extension to the work done in Chapters 3 and 4 is to consider higher order association or interactive effects between variables. Along that direction, the different order of interactive effects can be integrated in hyper-graphs where more generalized network-based feature selections are required to deal with hyper-graphs. However, as discussed in the introduction, the sample size limitation makes the higher order estimations unreliable. Moreover, the lack of biological justification for such higher order interactions may make it less practical. However, with the recent advances in profiling techniques, and large scale studies such as TCGA that can systematically gather data for a large number of patients, the sample size problem might recede to a point where higher order associations would become plausible. We believe that the first important step would be to determine the required number of samples for reliable estimates for higher order interactions. This can be done empirically using simulations or by extending the existing theories [32] for pairwise interactions.

In Chapters 5 and 6 we focused on improving the performance of traditional data mining approaches by taking into account prior biological knowledge in gene-gene interaction networks, pathway databases, and GO annotations. We observed that different sources of prior biological knowledge provide relatively different results. We attribute this to two main issues: 1) Different aspects of cellular systems are studied in different biological knowledge databases; 2) The depth to which certain parts of the cellular systems are studied are different, e.g., certain biological processes are studied more carefully compared to others. One potential direction would be to explore computational methods for effectively integrating all sources of biological knowledge, which requires careful consideration of the nature of the knowledge to avoid antagonistic effects and improve synergy. Another direction, would be to integrate several types of biomedical data provided by large-scale studies such as in TCGA. Several sources of high-throughput biomedical data are available for the patient including somatic mutation profiles, gene-expression, mRNA expression, and protein expression. These types of data can be integrated with the imaging data and clinical data for better analysis of complex disease. We believe that data-mining approaches that can simultaneously take advantage of such a vast sources of biomedical data and prior biological knowledge are essential for future research in computational biomarker identification for complex diseases such as cancer.
References


[86] P. Xu, Y. Wu, Y. Zhu, G. Dagne, G. Johnson, D. Cuthbertson, J. Krischer, J. Sosenko, J. Skyler, and
the DPT-1 Study Group, “Prognostic performance of metabolic indexes in predicting onset of Type 1

[87] E. R. Mardis and R. K. Wilson, “Cancer genome sequencing: a review,” Human Molecular Genetics,


[89] 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, and J. P. Mesirov, “Gene set enrichment analysis: A
knowledge-based approach for interpreting genome-wide expression profiles,” Proceedings of the

J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson,
M. Ringwald, G. Rubin, and G. Sherlock, “Gene ontology: tool for the unification of biology. the


and her2 expression: comparison of clinicopathologic features and survival,” Clinical medicine &


[94] J. Li, K. Bennett, A. Stukalov, B. Fang, G. Zhang, T. Yoshida, I. Okamoto, J.-Y. Kim, L. Song, Y. Bai,
X. Qian, B. Rawal, M. Schell, F. Grebien, G. Winter, U. Rix, S. Eschrich, J. Colinge, J. Koomen,
G. Superti-Furga, and E. B. Haura, “Perturbation of the mutated EGFR interactome identifies vulner-


Appendix A
More Experiments and Details on Interactive Effects

A.1 Estimating $H(y|x)$ Using UPGMA

The Chebychev distance is used to compute the distances between the given data points. The distances between the clusters are computed as the average of the distances between all pairs of points across two clusters. The UPGMA algorithm starts with $n$ clusters (each data point in a separate cluster) where $n$ is the number of data points. Then, in each clustering step, two clusters with the minimum distance between them are merged together and form a single cluster. This procedure is repeated as long as the minimum distance between clusters is less than a certain threshold (the threshold is set to 2.0 as stated in the original paper [22]). In order to obtain a more robust estimate of the entropy, as proposed in [22], rather than computing the entropy once only based on the final clustering result, we compute the entropy in each step of the UPGMA algorithm. The final entropy is computed as the average of the entropy estimates computed in all steps.

A.2 KNN Classifier Implementation Detail

To compute the posterior probability of the outcome given a data point $x^{(l)}$, first its top $K$ nearest neighbors are found (based on the Euclidean distance). Considering set $K$ is the set of all $k$ nearest neighbors and the set $T$ is the subset of those $k$ neighbors with class label 1, the posterior probability $P(y = 1|x = x^{(l)})$ is estimated as

$$\frac{\sum_{x^{(l')} \in T} w_{l'}}{\sum_{x^{(l')} \in K} w_{l'}}$$  \hspace{1cm} (A.1)

in which $w_{l'} = e^{-d_{l'}}$ and $d_{l'}$ is the Euclidean distance between the data points $x^{(l')}$ and $x^{(l)}$. Also, as previously mentioned, for those methods that use the KNN classifier (SynDichoKNN, SynPosteriorKNN, and AccKNN), the interactive effect is computed for all of the 10 different values of $K$ ($K = 1, 2, ..., 10$) and the average of them is considered as the final estimated value of the interactive effect.
A.3 Measuring the Association by MIC

In order to compute the Maximum Information Coefficient (MIC) between two variables $x_i$ and $x_j$, we have used the software package provided in [71] with default parameters. In order to compute the association after observing outcome, $\text{MIC}(x_i; x_j|y)$, we compute the MIC between the two variables once only based on observations with $y^{(l)} = 1$ ($\text{MIC}^1$) and once only based on the observations with $y^{(l)} = 0$ ($\text{MIC}^0$). The final statistical association after observing outcome is estimated by

$$\text{MIC}(x_i; x_j|y) = \frac{n_1 \times \text{MIC}^1 + n_2 \times \text{MIC}^0}{n},$$

(A.2)

where $n$ is the total number of observations, $n_1$ is the number of observations with $y = 1$, and $n_2$ is the number of observations with $y = 0$.

A.4 Simulated Datasets

The steps for simulating a single sample are summarized in Algorithm 1.

A.5 Simulation Results for 200 Variables

In order to verify the consistency of the results for better assessment of the selected methods when there are a larger number of variables, we have performed another experiment based on simulated datasets with 200 variables and for the following same sample sizes: 20, 40, 60, 80, 100, 120, 140. We have generated 100 datasets for each sample size to evaluate the average detection accuracy of simulated interactions. The results are shown in Figure A.17. As one can see, the performance of different methods based on this new set of simulated datasets is consistent with our original experiment in the manuscript.

A.6 Final Sets of Biomarkers for Both Individual-based and Network-based Ranking in Breast Cancer Datasets

Table A.9 provides the final sets of identified biomarkers by individual-based and network-based ranking methods, respectively.
Output: \((x_1, x_2, \ldots, x_n; y)\) \# one sample
\(y \leftarrow \text{uniformly random from } \{(0, 1)\}\) \# control or case;
if \(y = 1\) then
  for \((i, j)\in\text{Pairs with simple interactive effect}\) do
    \(\mu \leftarrow (\mu_1, \mu_2) \leftarrow \text{random from } \{(-0.5, -0.5), (0.5, 0.5)\}\);
    \((x_i, x_j) \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
  end
  for \((i, j)\in\text{Pairs with complex interactive effect}\) do
    \(\mu \leftarrow (\mu_1, \mu_2) \leftarrow \text{random from } \{(-1, -1), (1, -1), (0, 0), (-1, 1), (1, 1)\}\);
    \((x_i, x_j) \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
  end
  for \((i, j)\in\text{Pairs with very complex interactive effect}\) do
    \(\mu \leftarrow (\mu_1, \mu_2) \leftarrow \text{random from } \{(-1.5, -1.5), (-1.5, 0.5), (-0.5, -0.5), (-0.5, 0.5), (0.5, -1.5), (0.5, 0.5), (1.5, -1.5), (1.5, 1.5)\}\);
    \((x_i, x_j) \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
  end
  for \(k\in\text{First three variables with individual effect}\) do
    \(x_k \leftarrow \mathcal{N}(\mu = 0.5, \sigma = 0.5)\);
  end
  for \(k\in\text{Other three variables with individual effect}\) do
    \(\mu \leftarrow \text{random from } (0.5, 1.5)\);
    \(x_k \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\)
  end
else
  // \(y = 0\)
  for \((i, j)\in\text{Pairs with simple interactive effect}\) do
    \(\mu \leftarrow (\mu_1, \mu_2) \leftarrow \text{random from } \{(-0.5, 0.5), (0.5, -0.5)\}\);
    \((x_i, x_j) \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
  end
  for \((i, j)\in\text{Pairs with complex interactive effect}\) do
    \(\mu \leftarrow (\mu_1, \mu_2) \leftarrow \text{random from } \{(0, -1), (-1, 0), (1, 0)\}\);
    \((x_i, x_j) \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
  end
  for \((i, j)\in\text{Pairs with very complex interactive effect}\) do
    \(\mu \leftarrow (\mu_1, \mu_2) \leftarrow \text{random from } \{(-1.5, -0.5), (-1.5, 1.5), (-0.5, -1.5), (-0.5, 1.5), (0.5, -0.5), (0.5, 0.5), (1.5, -1.5), (1.5, 1.5)\}\);
    \((x_i, x_j) \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
  end
  for \(k\in\text{First three variables with individual effect}\) do
    \(x_k \leftarrow \mathcal{N}(\mu = -0.5, \sigma = 0.5)\);
  end
  for \(k\in\text{Other three variables with individual effect}\) do
    \(\mu \leftarrow \text{random from } (-0.5, -1.5)\);
    \(x_k \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\)
  end
end
for \(k\in\text{remaining variables with no effect following MoG with two means}\) do
  \(\mu \leftarrow \text{random from } \{-0.5, 0.5\}\);
  \(x_k \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
end
for \(k\in\text{remaining variables with no effect following MoG with four means}\) do
  \(\mu \leftarrow \text{random from } \{-1.5, -0.5, 0.5, 1.5\}\);
  \(x_k \leftarrow \mathcal{N}(\mu, \sigma = 0.5)\);
end
// add 10% noise
\(r \leftarrow \text{random from } [0, 1]\);
if \(r < 0.1\) then
  \(y \leftarrow \text{yes}\).
end

Algorithm 1: Algorithm for Simulating One Sample.
Figure A.17. Simulation dataset results with 200 randomly simulated variables in each sample. A: Existing parametric methods. B: Existing non-parametric methods. C: Our proposed parametric methods. D: Our proposed non-parametric methods. E: Comparison between all methods.

A.7 Gene Set Enrichment Analysis

To find the significance of the overlap between a set of user provided genes $S_1$ with size $n$ and a curated gene-set $S_2$ with size $K$, GSEA software first finds the size of their intersection $k$. Certainly, if the set $S_1$ is very large it will have intersection with so many curated gene-sets. On the other hand if set $S_2$ is large, it can have large intersection with any user provided set. So the size of the intersection ($k$), the size of set $S_1$ ($n$), the size of set $S_2$ ($K$), and also the number of all known genes ($N$) must be taken into account. To do
Table A.9 Final sets of biomarkers for both network-based ranking and individual-based ranking.

<table>
<thead>
<tr>
<th>network-based ranking</th>
<th>individual-based ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTR2, ATP5J, CA7, CFL1, CIT, CLDN11, CLDN5, COX15, CUL5, DLC1, ERCC1, GOLGA2, GSN, HTR1B, ITGB3BP, KIR3DL1, MAP3K4, MFAP3, MKI67, OR10C1, P2RY6, PFDDN5, POLI, PTMA, RANBP3, RPS9, SIGLEC7, SMARCA1, STIP1, TESK1, TMEM5, TROAP, TYMS, UQCRC2, VILL</td>
<td>ALPP, CETP, CHGA, CPS1, CYP2A7, EEF1A2, FCER2, GAD2, GPC5, GPR22, HPD, INSM1, ITH2, KCNA5, KCNJ8, MSMB, MYF6, NTS, NTSR2, PAH, PCSK1, SCN4A, SLC7A11, SULT1C2, SYCP1, TRH</td>
</tr>
</tbody>
</table>

Table A.10 Gene set enrichment analysis for biomarkers identified by the network-based ranking method.

<table>
<thead>
<tr>
<th>pathway</th>
<th>gene symbols</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>REACTOME_METABOLISM_OF_AMINO_ACIDS_AND_DERIVATIVES</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>4.83e-06</td>
</tr>
<tr>
<td>KEGG_PHENYLALANINE_METABOLISM</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>4.68e-05</td>
</tr>
<tr>
<td>KEGG_ALANINE_ASPARTATE_AND_GLUTAMATE_METABOLISM</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>1.51e-04</td>
</tr>
<tr>
<td>REACTOME_G_ALPHA_Q_SIGNALLING_EVENTS</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>1.53e-04</td>
</tr>
<tr>
<td>REACTOME_PEPTIDE_LIGAND_BINDING_RECEPTORS</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>1.63e-04</td>
</tr>
<tr>
<td>REACTOME_GASTRIN_CREB_SIGNALLING_PATHWAY_VIA_PKC_AND_MAPK</td>
<td>✓ ✓ ✓ ✓ ✓</td>
<td>2.11e-04</td>
</tr>
</tbody>
</table>

that, GSEA software uses hypergeometric distribution with parameters \((k - 1, K, N - K, n)\) which gives the significance of the overlap between \(S_1\) and \(S_2\).

Gene set enrichment analysis (GSEA) gives top enriched pathways for the identified biomarkers by the individual-based ranking without considering interactive effects among genes (Table 10). Compared to these biomarkers, more enriched pathways that are related to metastasis have been shown in GSEA for the identified biomarkers from network-based ranking.
Appendix B
Copyright Permissions

Below is permission for Chapter 4.

Permissions Request

This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Springer and BioMed Central offer a reprint service for those who require professionally produced copies of articles published under Creative Commons Attribution (CC BY) licenses. To obtain a quotation, please email reprints@springeropen.com with the article details, quantity(ies) and delivery destination. Minimum order 25 copies.