Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

Yunchuan Kong	Date

Network-Based Machine Learning Methods for Omics Data

By

Yunchuan Kong

Doctor of Philosophy

Biostatistics

Tianwei Yu, Ph.D.
Advisor

Hao Wu, Ph.D.
Committee Member

Zhaohui Qin, Ph.D.
Committee Member

Glen Satten, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Network-Based Machine Learning Methods for Omics Data

By

Yunchuan Kong
M.S., Emory University, 2019
B.Sc., The Chinese University of Hong Kong, 2015

Advisor: Tianwei Yu, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2020

Abstract

Network-Based Machine Learning Methods for Omics Data

By

Yunchuan Kong

In the field of bioinformatics, large-scale biological networks play an essential role for studying transcriptomic data. As networks can bring useful relational information in solving problems, tasks involving biological networks range from system biology, statistical modeling, to machine learning. In this dissertation, focusing on different roles of biological networks, we explore both the construction of networks and their integration with statistical methods. On the one hand, we have found hypergraphs, an extension of traditional networks, to be an excellent tool to represent higher-order interactive relationships among biological units and analyze complex systems; on the other hand, we have discovered that incorporating known biological networks and constructing biological feature networks can be helpful in improving certain supervised machine learning algorithms.

The first topic of this dissertation is about the highly dynamic biological regulatory system. It is shown that correlations between certain functionally related genes change over different biological conditions, which are often unobserved in the data. At the gene level, the dynamic correlations result in three-way gene interactions involving a pair of genes that change correlation, and a third gene that reflects the underlying cellular conditions. This type of ternary relation can be quantified by the Liquid Association statistic. Studying these three-way interactions at the gene triplet level have revealed important regulatory mechanisms in the biological system. Currently, due to the extremely large amount of possible combinations of triplets within a high-throughput gene expression dataset, no method is available to examine the ternary relationship at the biological system level. Hence, in Chapter 2, we propose a new method, Hypergraph for Dynamic Correlation (HDC), to construct module-level three-way interaction networks. The method is able to present integrative uniform hypergraphs to reflect the global dynamic correlation pattern in the biological system, providing guidance to downstream gene triplet-level analyses. To validate the method's ability, we conducted two real data experiments using a melanoma RNA-seq dataset from The Cancer Genome Atlas (TCGA) and a yeast cell cycle dataset. The resulting hypergraphs are clearly biologically plausible, and suggest novel relations relevant to the biological conditions in the data. We believe the new approach provides a valuable alternative method to analyze omics data that can extract higher order structures.

In the second topic of this dissertation, we aim at solving a unique challenge in predictive modeling for gene expression data, which usually bear small samples (n) compared to the huge amount of features (p). This "$n \ll p$" property has hampered application of deep learning techniques for disease outcome classification. Recently,

literature shows that sparse learning by incorporating external gene network information could be a potential solution to this issue. To build a robust classification model, we propose the Graph-Embedded Deep Feedforward Networks (GEDFN) in Chapter 3, to integrate external relational information of features into the deep neural network architecture. The method is able to achieve sparse connection between network layers to prevent overfitting. To validate the method's capability, we conducted both simulation experiments and real data analysis using a Breast Invasive Carcinoma (BRCA) RNA-seq dataset and a Kidney Renal Clear Cell Carcinoma (KIRC) RNA-seq dataset from The Cancer Genome Atlas (TCGA). The resulting high classification accuracy and easily interpretable feature selection results suggest the method is a useful addition to the current graph-guided classification models and feature selection procedures.

The third topic of this dissertation is an extension of the second topic. Faced with the "$n \ll p$" challenge in predictive modeling, the GEDFN model with sparse learning by incorporating known functional relations between the biological units, has been proved a solution to this issue in Chapter 3. However, such methods require an existing feature graph, and potential mis-specification of the feature graph can be harmful on classification and feature selection. To address this limitation and develop a robust classification model without relying on external knowledge, we propose a forest graph-embedded deep feedforward network (forgeNet) model in Chapter 4, to integrate the GEDFN architecture with a forest feature graph extractor, so that the feature graph can be learned in a supervised manner and specifically constructed for a given prediction task. Similar as in Chapter 3, to validate the method's capability, we experimented the forgeNet model again with both synthetic and real datasets. The resulting high classification accuracy suggests that the method is a valuable addition to sparse deep learning models for omics data.

In the future work, possible directions are to continue exploring the integration of biological networks and statistical modeling. Certain research area has already been established such as the Graph Convolution Network (GCN). Also, following our construction of hypergraphs in the first topic, it is also tempting to study further applications beyond the scientific findings themselves.

Network-Based Machine Learning Methods for Omics Data

By

Yunchuan Kong
M.S., Emory University, 2019
B.Sc., The Chinese University of Hong Kong, 2015

Advisor: Tianwei Yu, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2020

Acknowledgement

Words cannot express my deepest thanks to dear advisor, Prof. Tianwei Yu, who not only directed me to the final destination of this incredible doctoral journey, but also shaped me into a scholar equipped with established knowledge, critical thinking, and eagerness of exploring the world. During past years, Tianwei has been supportive to me in every regard. His sharp insight illuminated the future direction at my "dark time", when I knew nowhere to go. His smart ideas helped me out when I was trapped into problems that seemed unsolvable. His generosity allowed me to embrace abundant resources for my research, driving our ideas to come true. Last but not least, his unique personality made him always a nice person to communicate with-discussing and chatting in his office has been one of the most enjoyable memories throughout my academic career. It was my fortune to have him as my Ph.D. advisor and as a lifetime friend in the future.

My sincere appreciation also extend to Dr. Hao Wu, Dr. Zhaohui Qin and Dr. Glen Satten, for being my dissertation committee members. Their guidance has been beyond just for this dissertation, and beneficial for me since the early stage of my Ph.D. study. I thank Prof. Yijian Huang for being my academic advisor in my junior years and constantly caring about my growth. Although it is not possible to list every name here, I appreciate the support from all of the faculty members, staff and friends in the Department of Biostatistics and Bioinformatics, which I am grateful for being part of. In this department, people are always ready to help.

The very selfless persons who have supported me without any reservation are my parents. They respected my choice of studying abroad, despite the bitterness of letting the family be apart. It was tough, and I owe them invaluable love that I might never have a chance to repay. Now, as my student career comes to an end, there is an

even bigger world out there waiting for me to explore. I will be stepping onto the next stage with my parents' best wishes, again. No matter how far I go, the inspiration from them is always planted in the deepest place of my heart. I love you, mum and dad. I also dedicate this dissertation to my grandfather, who passed away last fall and was not able to see the completion of my Ph.D.. I believe he would be proud of me as always.

On the road of my Ph.D. journey, I was not alone. It was lucky to meet my sweet girlfriend, Yutong, here in the BIOS department. I thank her for accompanying and supporting me all the way. I look forward to continuing our life journey together and wish all the best to her academic career as well.

Contents

1 Introduction **1**

1.1 Overview 2

1.2 Biological networks for dynamic correlation 2

1.3 Supervised learning with known biological feature graphs 5

1.4 Supervised learning with constructed biological feature graphs 6

2 HDC: Hypergraph for Dynamic Correlation **8**

2.1 Introduction 9

2.2 Methodology 10

2.2.1 Quantifying the ternary relationship 10

2.2.2 Selecting significant triplets using permutation and mixture models 12

2.2.3 Selecting gene modules using supervised and unsupervised approaches 13

2.2.4 Constructing the module-level hypergraph 15

2.3 Results 16

2.3.1 Human cutaneous melanoma dataset 16

2.3.2 Yeast cell cycle dataset 26

2.4 Discussion 31

2.5 Conclusion 33

3 GEDFN: Graph-Embedded Deep Feedforward Network 34

3.1 Introduction . . . 35

3.2 Methodology . . . 36

3.2.1 Deep feedforward networks . . . 36

3.2.2 Graph-embedded deep feedforward networks . . . 38

3.2.3 Evaluation of feature importance . . . 40

3.2.4 Detailed model settings . . . 41

3.3 Simulations . . . 42

3.3.1 Synthetic data generation . . . 43

3.3.2 Simulation results and discussion . . . 45

3.4 Data Analysis . . . 50

3.4.1 Breast invasive carcinoma data . . . 50

3.4.2 Kidney renal clear cell carcinoma data . . . 55

3.5 Conclusion . . . 58

4 forgeNet: Forest Graph-Embedded Deep Feedforward Network 59

4.1 Introduction . . . 60

4.2 Methodology . . . 61

4.2.1 The forgeNet model . . . 61

4.2.2 Evaluation of feature importance . . . 63

4.2.3 Implementation . . . 64

4.3 Simulations . . . 65

4.3.1 Synthetic data generation . . . 66

4.3.2 Evaluation of simulation experiments . . . 67

4.3.3 Simulation results . . . 69

4.3.4 Analysis of estimated feature graphs . . . 71

4.4 Data Analysis . . . 73

4.4.1 Breast invasive carcinoma RNAseq data . . . 73

4.4.2 Breast invasive carcinoma microRNA data 76
4.4.3 Healthy human metabolomics dataset 79
4.5 Conclusion 81

A Appendix for Chapter 3 **83**
A.1 Hyper-parameter tuning of GEDFN 83
A.1.1 Overview 83
A.1.2 Architecture 84
A.1.3 Regularization 84
A.1.4 Training 84
A.1.5 Note 85
A.2 Correlation distributions of synthetic data 86
A.3 Mis-specification of feature graphs 87

B Appendix for Chapter 4 **88**
B.1 Correlation distributions of synthetic data 88
B.2 Sensitivity analysis 88
B.2.1 Initial values 88
B.2.2 Hyper-parameters 90
B.3 Computational cost 91
B.4 Simulation experiments for datasets with no signal 92

Bibliography **95**

List of Figures

2.1 The flow chart of the analysis. 10

2.2 Visualization of the hypergraph for the TCGA melanoma dataset with supervised grouping. (a) The plot of the entire network, where hyperedges were reduced to binary edges for visualization; (b) Detailed plot of the top 15 most connected vertices; (c) Sub-hypergraph centered at the module "DNA damage response, signal transduction by p53 class mediator"; (d) Sub-hypergraph centered at the module "DNA dependent DNA replication". Vertex colors reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices. 17

2.3 Visualization of the gene-level hypergraph of the triplet "DNA damage response, signal transduction by p53 class mediator", "DNA damage response, signal transduction by p53 class mediator", and "sphingolipid metabolic process". Vertex sizes reflect the degree, with more connected nodes larger. All gene-level hyperedges are type 2 edges. . 18

2.4 Visualization of the gene-level hypergraph of the triplet "visual perception", "DNA-dependent DNA replication", and "vacuole organization". Vertex sizes reflect the degree, with more connected nodes larger. All gene-level hyperedges are type 3 edges. 19

2.5 Visualization of the hypergraph for the TCGA melanoma dataset with unsupervised grouping. (a) The plot of the entire network. (b) Detailed plot of the top 15 most connected vertices. Vertex colors reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices. 20

2.6 Hypergraph of the yeast cell cycle dataset with supervised grouping. (a) The plot of the entire network; (b) Detailed plot of the top 15 most connected vertices; (c) Sub-hypergraph centered at the module Single organism membrane budding; (d) Gene level hypergraph for the module triplet "single-organism membrane budding", "G2M transition of mitotic cell cycle", and "pyruvate metabolism". For the module-level hypergraph, vertex sizes reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices. For the gene-level hypergraph, vertex sizes reflect the degree, with more connected nodes larger. 27

2.7 An example triplet of yeast genes. The 2-D scatter plot of expression values from PFK1 and VRS4 is given, where PIN4 serves as the "scouting gene" z. The points are divided into three groups according to the expression level of z, with low level the first 1/3 percentile, medium level the middle 1/3 percentile, and high level the last 1/3 percentile. Three lines (red solid, green dotted, and blue dashed) denote the corresponding linear regression lines of PFK1 over VRS4, given the three levels of PIN4. 29

2.8 Visualization of the hypergraph for the yeast dataset with unsupervised grouping. (a) The plot of the entire network; (b) Detailed plot of the top 15 most connected vertices. Vertex sizes reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices. . . 29

3.1 Network architecture of the GEDFN model for experiments in Section 3.3 and Section 3.4. 42

3.2 Plots of the classification and feature selection comparison for the case with the sigmoid inverse link function. Singleton proportions: left column 0%, middle column 50%, right column 100%. First row: AUC of ROC for classification; second row: AUC of precision-recall for feature selection; third row: recall plots given fixed precision from LRL. Error bars represent the estimated mean quantity plus/minus the standard error. 47

3.3 Plots of the classification and feature selection comparison for the case with the weighted tanh plus quadratic inverse link function. Singleton proportions: left column 0%, middle column 50%, right column 100%. First row: AUC of ROC for classification; second row: AUC of precision-recall for feature selection; third row: recall plots given fixed precision from LRL. Error bars represent the estimated mean quantity plus/minus the standard error. 49
3.4 Feature sub-graph selected by GEDFN for BRCA data. 53
3.5 Feature sub-graph selected by GEDFN for KIRC data. 57

4.1 Illustration of the forgeNet model. Notations are consistent with those in the text. 63
4.2 Comparison of classification and feature selection for the simulation study. (a) AUC of ROC for classification; (b) AUC of precision-recall for feature selection; (c) recall plots given fixed precision from LRL. Error bars represent the estimated mean quantities plus/minus the estimated standard errors. 68

A.1 Histograms of the pairwise feature correlation distributions for randomly selected simulation datasets. 86
A.2 Comparison of classification and feature selection between GEDFN using informative graphs and GEDFN with misspecified graphs. Left column: sigmoid inverse link; right column: tanh plus quadratic inverse link. First row: AUC of ROC for classification; second row: AUC of precision-recall for feature selection. Error bars represent the estimated mean quantity plus/minus the standard error. 87

B.1 Histograms of the pairwise feature correlation distributions for randomly selected simulation datasets. 89

List of Tables

2.1 Enrichment analysis of the human dataset for the top 15 most connected clusters. For each cluster, the enriched term that include the most number of genes in the cluster is shown. 21

2.2 Enrichment analysis of the yeast dataset for the top 15 most connected clusters. For each cluster, the enriched term that include the most number of genes in the cluster is shown. 30

3.1 Classification results for BRCA data 51

3.2 Selected feature sub-graph analysis for BRCA data 52

3.3 Top GO biological processes for the sub-graph selected by GEDFN (BRCA data). Manual pruning of partially overlapping GO terms was conducted. 54

3.4 GO enrichment analysis for features selected by GEDFN only (BRCA data). Manual pruning of partially overlapping GO terms was conducted. 54

3.5 Classification results for KIRC data 55

3.6 Top GO biological processes for the sub-graph selected by GEDFN (KIRC data). Manual pruning of partially overlapping GO terms was conducted. 58

4.1 Analysis of feature graphs constructed by RF and GBM. Proportions are averaged across the 50 datasets in each simulation case. 72

4.2 Classification results for BRCA data 73
4.3 Top 3 GO biological processes for each method, after manual removal of redundant GO terms. 74
4.4 Classification results for BRCA microRNA data 77
4.5 Top 5 pathways selected by each method using mirPath V.3. 78
4.6 Classification results for healthy human metabolomics data 79
4.7 Top 5 pathways selected by each method using Mummichog. 80

B.1 Testing ROC-AUC of repeated experiments for fixed datasets. Left: forgeNet-RF. Right: forgeNet-GBM. 90
B.2 An example of hyper-parameter tuning for forgeNet (RF). The column names denote **hidden layers** and their corresponding numbers of hidden neurons. For example, "p+64+16" stands for a neural network architecture with a p-dimensional input layer, a p-dimensional graph-embedded layer, a 64-dimensional fully connected hidden layer, a 16-dimensional fully connected hidden layer and a two-dimensional output layer. 91
B.3 Computational time for forgeNet-RF and the corresponding RF model alone. The time used by RF and by forgeNet-RF are separated by "/". For example, "4.5/14.3" means the time of running RF is 4.5 seconds while running the entire forgeNet takes 14.3 seconds. 92

B.4 Memory usage for forgeNet-RF and the corresponding RF model alone. Since we used the GPU version of the Tensorflow deep learning library, forgeNet merely induced additional space cost in terms of GPU memory only, and the RAM usage remained the same as the corresponding tree-ensemble method. The extra (GPU) memory usage of forgeNet-RF is shown in a bracket. For example, "141.2 (73.1)" means the RF extractor used 141.2 MB RAM memory, and to train the entire forgeNet, 73.1 MB extra GPU memory were also used. 93

B.5 Computational time for forgeNet-GBM and the corresponding GBM model alone. The time used by GBM and by forgeNet-GBM are separated by "/". For example, "5.5/11.0" means the time of running GBM is 5.5 seconds while running the entire forgeNet takes 11.0 seconds. 93

B.6 Memory usage for forgeNet-GBM and the corresponding GBM model alone. Since we used the GPU version of the Tensorflow deep learning library, forgeNet merely induced additional space cost in terms of GPU memory only, and the RAM usage remained the same as the corresponding tree-ensemble method. The extra (GPU) memory usage of forgeNet-GBM is shown in a bracket. For example, "138.4 (2.4)" means the GBM extractor used 138.4 MB RAM memory, and to train the entire forgeNet, 2.4 MB extra GPU memory were also used. . . . 93

B.7 Classification results for "null" datasets 94

Chapter 1

Introduction

1.1 Overview

In the study of transcriptomics, metabolomics, and proteomics data, biological networks, including gene regulation networks, metabolomics pathways and protein-protein interaction networks, play an essential role. In this dissertation, we are interested in exploring insights brought by biological networks in a broad sense, ranging from the usefulness of bio-networks in machine learning to the construction for high-dimensional network representation.

Typically, a network, or equivalently a graph $\mathbf{G} = (V, E)$ consists of two sets of elements. V represents the collection of all nodes (or vertices) in the network, and E denotes the collection of all edges connecting nodes. In biological networks, biological units can be represented by network vertices, the connections or interactions are then the edges. For example, a gene network represents a biological system with interested biological units genes, hence vertices denote different genes and edges can be regulatory relationships among genes.

In Chapter 2, We first investigate the power of biological networks in understanding dynamic biological systems, by proposing a hypergraph construction method for dynamic correlation. The newly constructed module level hyper-networks along with the corresponding visualizations can reveal helpful information at the transcriptomic scale. Given the evidence of the effectiveness of biological networks, we next try to extend the application of biological networks in classification for omics data. In Chapter 3 and Chapter 4, we develop two deep learning-based classification methods utilizing biological feature networks.

1.2 Biological networks for dynamic correlation

In the quantitative analysis of high-throughput omics experiments, the gene transcript-, protein- or metabolite-levels of abundance are profiled simultaneously. Examples

include high-throughput sequencing of mRNA (RNA-seq) and high throughput mass spectrometry for quantitative analysis of specific cellular proteome or metabolites. The abundance levels of the biological units are the outcome of complex biological regulatory networks, in which the links between the units may be turned on and off in response to certain biological conditions (Barzel and Barabási, 2013; Ideker and Krogan, 2012; Luscombe et al., 2004; Ocone et al., 2013). As a result, many correlations are dynamic, shifting between positive, negative and no-correlation states, triggered by certain biological conditions. Such conditions may not be phenotype changes, e.g. disease/non-disease in case-control studies, but they may be more subtle and often unobservable (Li, 2002; Li, Liu, Sun, Yuan and Yu, 2004).

Given the profiling data are essentially snapshots of the system, it is challenging to extract higher order relations from the data, such as conditional correlations and changes in variability. To explore patterns in high-throughput expression data, methods that include clustering, dimension reduction, sparse factorization have been proposed. These methods are mostly based on static pairwise relations between the biological units, and do not capture dynamic relations (Xu and Wunsch, 2010; Ma and Dai, 2011).

According to Li (2002); Boscolo et al. (2008); Chen et al. (2011), the expression levels of certain genes can be treated as indicators of cellular states, and correlation changes conditioned on such genes are computed to measure dynamic correlations. The involvement of such genes as dynamic correlation condition results in three-way gene interactions, and quantitative measures for the three-way interaction have been developed to quantify the ternary relationship, such as the Liquid Association (LA) statistic proposed by Li (2002), the Modified Liquid Association (MLA) developed by Ho et al. (2011), and the z-statistic in Zhang et al. (2007). These ideas have been demonstrated successful in practice showing interpretable biological findings at the gene level. Biologically, it is plausible that a single gene may not be a good

proxy measure of the underlying condition for the dynamic correlation. However measures involving more than one gene as the conditioning variable is difficult to design, and costly in computation. To address this issue, a method treating the LA relation as latent factor model has been developed, where in stead of using genes as proxy measures, the conditioning variable is estimated from the data (Yu, 2018). However such an approach can only find dominating signals that control the dynamcic correlations of large numbers of gene pairs. Some critical dynamic correlation may happen among a small group of genes, yet play important biological roles. Hence an unbiased examination of all gene triplets is valuable.

Currently, the existing methods suffer from computational scalability when examining the entire biological system since it is difficult to examine gene-level three-way interactions triplet-by-triplet as the amount of possible combinations is extremely large. Efforts have been made to focus on a smaller number of subsets, by considering consistent LA relations across multiple datasets (Wang et al., 2017), or focusing on subnetwork-level LA relations (Yan et al., 2017).

Meanwhile, it is desirable to view the complex interactions of individual triplets jointly as a whole, since otherwise it is hard to grasp the dynamic correlation behaviors at the system level. Therefore, an aggregated representation is in need for ternary gene relationships, analogous to the gene co-expression network for the pairwise static correlation relationship. The gap resulted from this problem motivated the work in Chapter 2, where we develop a hypergraph-based approach constructing module-level three-way interaction networks for ternary gene relationship study.

1.3 Supervised learning with known biological feature graphs

In recent years, more and more studies attempt to link clinical outcomes, such as cancer and other diseases, with gene expression or other types of profiling data. It is of great interest to develop new computational methods to predict disease outcomes based on profiling datasets that contain tens of thousands of variables. The major challenges in these data lie in the heterogeneity of the samples, and the sample size being much smaller than the number of predictors (genes), i.e. the $n \ll p$ issue, as well as the complex correlation structure between the predictors. Thus the prediction task has been formulated as a classification problem combined with selection of predictors, solved by modern machine learning algorithms such as regression based methods (Liang et al., 2013; Algamal and Lee, 2015), support vector machines (Vanitha et al., 2015), random forests (Kursa, 2014; Cai et al., 2015) and neural networks (Chen et al., 2014). While these methods are aimed at achieving accurate classification performance, major efforts have also been put on selecting significant genes that effectively contribute to the prediction (Kursa, 2014; Cai et al., 2015). However, feature selection is based on fitted predictive models and is conducted after parameter estimation, which causes the selection to rely on the classification methods rather than the structure of the feature space itself. Beside building robust predictive models, the feature selection also serves another important purpose-the functionality of the selected features (genes) can help unravel the underlying biological mechanisms of the disease outcome.

Given the nature of the data, i.e. functionally associated genes tend to be statistically dependent and contribute to the biological outcome in a synergistic manner, a branch of gene expression classification research has been focused on integrating the relations between genes with classification methods, which helps in terms of both

classification performance as well as learning the structure of feature space. A critical data source to achieve this goal has been gene networks. A gene network is a graph-structured dataset with genes as the graph vertices and their functional relations as graph edges. The functional relations are largely curated from existing biological knowledge (Chowdhury and Sarkar, 2015; Szklarczyk and Jensen, 2015). Each vertex in the network corresponds to a predictor in the classification model. Thus, it is expected that the gene network can provide useful information for a learning process where genes serve as predictors. Motivated by this fact, certain procedures have been developed where gene networks are employed to conduct feature selection prior to classification (Chuang et al., 2007; Wei and Pan, 2007; Wang et al., 2007; Li and Li, 2008). Moreover, methods that integrate gene network information directly into classifiers have also been developed. For example, Dutkowski and Ideker (2011) proposes the random forest-based method, where the feature sub-sampling is guided by graph search on gene networks when constructing decision trees. Zhu et al. (2009); Lavi et al. (2012) modify the objective function of the support vector machine with penalty terms defined according to pairwise distances between genes in the network. Similarly, Kim et al. (2013) develops logistic regression based classifier using regularization, where again a relational penalty term is introduced in the loss function. The authors of these methods have demonstrated that embedding expression data into gene network results in both better classification performance and more interpretable selected feature sets.

1.4 Supervised learning with constructed biological feature graphs

In Chapter 3, the Graph-Embedded Deep Feedforward Network (GEDFN) is proposed with a known biological network embedded as a hidden layer in deep neural

networks, in order to achieve an informative sparse structure. In GEDFN, the graph-embedded layer helps achieve two effects. One is model sparsity, and the other is the informative flow of information for prediction and feature evaluation. These two effects allow GEDFN to outperform other methods in profiling data classification given an appropriately specified feature graph. However, methods utilizing known biological network information, such as GEDFN, bear a common limitation, which is the potential mis-specification of the required biological network. In practice, profiling data are used for various clinical outcomes, and the mechanistic relations between biological units and different clinical outcomes can be quite different. Hence, there does not exist a single known network that uniformly fits all classification problems. Thus, biological networks used in graph-embedded methods can only be "useful" but not "true". Consequently, how to decide if a known biological network is useful in predicting a certain clinical outcome with a certain gene expression dataset remains an unsolved problem, causing difficulties in applying graph-embedded methods in practice. In Chapter 3, we discuss the feature graph mis-specification issue of the GEDFN model and show that the method is robust with mis-specified biological networks. Nevertheless, it is unrealistic to guarantee that the robustness applies in a broad sense, as feature graph structures can be extremely diverse such that simulation would not be able to cover all scenarios.

To address these issues, in Chapter 4, we aim at developing a method that doesn't rely on a given feature network, yet can still benefit from the idea of building a model with sparse and informative flow of information. Instead of using known feature graphs, we try to construct a feature graph within the feature space.

Chapter 2

HDC: Hypergraph for Dynamic Correlation

2.1 Introduction

The main difficulty to analyze three-way interaction for an entire system is the extremely large amount of possible triplets at the gene level. For example, for a gene-expression dataset with 20,000 genes, the number of possible combinations would be around 1.33×10^{12}. Thus, one can do little when trying to describe the entire system while focusing on gene-level interactions. To resolve the dilemma, we consider a bottom-up approach to bring the ternary relationship to the module level, while preserving partial information of gene-level three-way interactions. This idea allows us to shrink the scale of the system and thus facilitate the aggregated representation. For this purpose, it is natural to use a hypergraph to present the ternary relations.

Similar to traditional graphs, a hypergraph $\mathbf{G} = (V, E)$ with V the set of vertices and E the set of edges, is a generalization of a graph in the sense that an edge can connect any number of vertices rather than just two (Berge and Minieka, 1973). A special case when all the edges in E connect a certain number of vertices k, the hypergraph is called k-uniform hypergraph. Therefore, a traditional graph or network is just a two-uniform hypergraph, and it is obvious that in our case the triplets compose a three-uniform hypergraph.

We utilize Liquid Association (LA) (Li, 2002), which is the most computationally tractable among the methods for ternary relations, for the initial gene-level ternary relationship quantification. Screening procedures using mixture models are conducted to ensure the LA accurately detects significant ternary correlation, according to Ho et al. (2011). Two approaches of grouping genes are then introduced, one of which involves a new clustering procedure based on ternary relations. Using these approaches, three-way interaction hypergraphs are constructed. The workflow of our analysis is demonstrated in Fig. 2.1. We applied our methods to two real datasets, the TCGA human cutaneous melanoma dataset (Weinstein et al., 2013) and the the yeast cell cycle dataset (Spellman et al., 1998). For both datasets, module-level three-

way interaction networks were obtained, exhibiting relations that conform to existing knowledge, as well as point to new and plausible dynamic correlations.

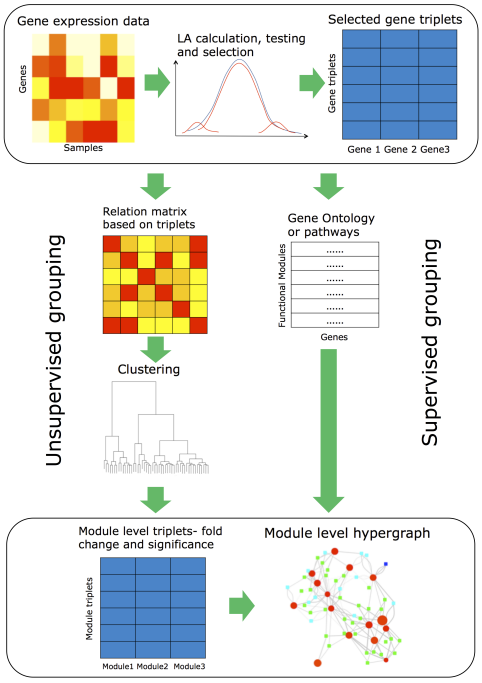


Figure 2.1: The flow chart of the analysis.

2.2 Methodology

2.2.1 Quantifying the ternary relationship

The input of our analysis is an ordinary $n \times m$ gene expression data matrix, with rows representing genes and columns representing specific samples. The ternary relationship is quantified by the statistic Liquid Association (LA) proposed in Li (2002). The LA statistic measures the extent to which the correlation of a pair of variables (X,Y) depends on the value of a third variable Z. Thus, the pairwise correlation is dynamic in the sense that it is affected dynamically according to the third variable. Based on this property, the LA statistic is therefore a suitable tool to quantify the ternary relationship for triplets of variables.

Specifically, according to Li (2002), suppose we are interested in measuring the ternary correlation among X, Y and Z. Without loss of generality, we can regard the ternary correlation as the dynamic pairwise correlation between X and Y given the third variable Z. The LA statistic is three-way symmetric regardless which variable is treated as the conditioning variable, or "scouting gene". Now let $g(Z)$ be the conditional expectation of the correlation between X and Y, namely, $g(Z) = E_{X,Y}(XY|Z)$. Then, the LA statistic is defined as the expected changes of the correlation between X and Y: $LA(X, Y|Z) = E_Z(g'(Z))$. When the variables are normalized with mean zeros, it is proved in Li (2002) that $LA(X, Y|Z) = E(XYZ)$, which means the LA statistic of X and Y given Z is just the expectation of the product XYZ. Therefore, the LA statistic can be estimated simply by calculating the sample mean of the product XYZ:

$$E(XYZ) = \frac{1}{m}\Sigma_{i=1}^{m} X_i Y_i Z_i, \tag{2.1}$$

where m is the dimension of the variables. Note that following this definition, LA is invariant of which variable (X, Y, or Z) we treat as the dynamic correlation condition, hence gives a measure of the ternary correlation.

Straightforward as the LA is, Ho et al. (2011) points out that for the quantity $E(XYZ)$ to reflect the true dynamic correlation of X and Y given Z, certain conditions must be met. They therefore developed the Modified Liquid Association (MLA) statistic to detect the dynamic correlation more accurately, which incurs a much higher computing cost. Also, in Ho et al. (2011), the authors proved that the MLA of X and Y given Z (denoted as $MLA(X, Y|Z)$) is equivalent to $E(XYZ)$ as well when certain conditions are satisfied (Theorem 1, (Ho et al., 2011)). These conditions include the normality of the "third variable" Z and the distributions of $X|Z$ and $Y|Z$ have constant mean and variance.

In our analysis, the $n \times m$ gene expression data matrix is normalized using normal score transformation for every row following Li (2002), and we are interested in the

ternary correlation among three variables. In the initial phase of selecting related gene triplets, which specific variable serves as the dynamic condition is less important. Also, Li's original approach is computationally better suited for transcriptome-scale scans. Thus, the invariant property regarding the dynamic condition variable of $E(XYZ)$ is desirable. To preserve this property, we restrict the mutual pairwise correlations within a triplet to be small, creating a sufficient condition for Theorem 1 in Ho et al. (2011). To see how this is achieved, if X, Y and Z are marginally normally distributed, and all the three pairwise correlations, $corr(X, Y)$, $corr(X, Z)$ and $corr(Z, Y) \approx 0$, then X, Y and Z are three independent normal variables. Hence, if the sequence U_1, U_2, U_3 is any permutation of X, Y, Z, then $E(U_1|U_3) = E(U_1) = 0$, $Var(U_1|U_3) = Var(U_1) = 1$, $E(U_2|U_3) = E(U_2) = 0$, $Var(U_2|U_3) = Var(U_2) = 1$, and $U_3 \sim \mathcal{N}(0, 1)$ are satisfied. Hence, the ternary correlation of a triplet (X,Y,Z) satisfies the condition in Ho et al. (2011), and can be quantified by Equation 2.1. Notice this requirement of low pairwise correlations also satisfy Li's original setup of Liquid Association (Li, 2002).

2.2.2 Selecting significant triplets using permutation and mixture models

The ternary correlation is calculated gene by gene, namely for each gene Z, the sample product mean of Z and all possible gene pairs (X,Y) are calculated, for all the triplets satisfying the condition discussed in Section 2.2.1. We expect only a small portion of the triplets to have true ternary relationship. The ternary correlation of triplets with insignificant relationship approximately follow a normal distribution (Ho et al., 2011). We employ a permutation procedure to estimate the parameters of the distribution.

To simplify our illustration, we define $\lambda^{(Z)}_{(i,j)}, (i, j) \in \{all\, possible\, pairs\, for\, Z\}$ as the ternary correlation associated with the given gene Z with the other two genes X and Y varying , and $\hat{\lambda}$ the sample product mean. The permutation selection is

conducted as following: after calculating the ternary correlation $\hat{\lambda}^{(Z)}$ of all possible triplets for a gene Z, an empirical distribution of $\lambda^{(Z)}$ is obtained. We then randomly permute the sample labels of Z and calculate all ternary correlations with all the $\{X, Y\}$ pairs again, obtaining another empirical distribution of $\lambda^{(Z^*)}$, which is considered as the null distribution where X, Y and Z have no ternary relationship. We estimate the two densities of the distributions using the kernel density estimation technique (Duong, 2007). Then, the ratio between the estimated permutation empirical density and the estimated actual empirical density, at a given value of ternary correlation λ, serves as the false discovery rate, i.e. the posterior probability that a λ at this value belongs to the null distribution:

$$fdr^{(Z)}(\lambda) = \hat{f}_0^{(Z^*)}(\lambda)/\hat{f}^{(Z)}(\lambda). \tag{2.2}$$

Setting a small number of false discovery rate, say 0.1, we are able to obtain the corresponding threshold on the value of λ. Triplets with a false discovery rate lower than the threshold are selected. The calculation and selection procedure is repeated for every gene in the dataset. Finally we obtain the entire list of triplets with significant ternary correlation. We note that the fdr estimate doesn't inflate in theory due to the large number of Z's being considered, because the null density doesn't change shape with more null λ values being calculated.

2.2.3 Selecting gene modules using supervised and unsupervised approaches

As mentioned in Section 4.1, given the large amount of gene-level triplets, it is impractical to present the three-way interactions of the system as a whole. Therefore, it is necessary to "build up" the system structure to a gene module level by dividing genes in a dataset into different modules. To achieve this, two options are available.

The first choice, which we refer to as supervised grouping, is to borrow external biological information such as gene functional modules from gene ontology (GO) terms (Consortium et al., 2015). We follow a procedure of selecting a subset of informative GO terms (Yu et al., 2005). While some genes in the dataset may not be included in the functional modules, other genes may appear in more than one modules. In the first case, the genes are ignored since they do not contribute to module-level information according to the external information. In contrast, in the latter case, the duplication of a certain set of genes across multiple modules is preserved as the set of genes contribute to multiple module level information.

The alternative way of grouping genes is clustering based on the gene level hypergraph structure, which is correspondingly an unsupervised grouping approach. In this study we base our clustering on the marginal relations between pairs of genes. To utilize the information of ternary relationship provided by the triplets selected in Section 2.2.2, we first construct an $n \times n$ matrix A recording the number of involvement of pairs in triplets, where n is the total number of genes in the dataset. Specifically, for example, if a triplet of genes (i,j,k) is selected after the procedure described in Sections 2.2.1 and 2.2.2, then according to the existence of this ternary correlation, the elements $A_{i,j}, A_{j,i}, A_{i,k}, A_{k,i}, A_{j,k}, A_{k,j}$ are all added by one to receive a "count". This counting procedure is repeated for the entire triplet list. Finally, the A matrix contains the amount of connections between any pair of genes when they jointly appear in a triplet. The diagonal elements of A are all set to zero since it is meaningless to consider self-connection here, and it is easy to see A is symmetric.

Given the matrix A, one can calculate the correlation matrix C for A, as it measures the similarity of the involvement in triplets among genes. Thus, using either the similarity matrix C or the corresponding distance matrix $1_{n\times n} - C$, where $1_{n\times n}$ is an $n \times n$ matrix with all elements equal to one, traditional distance-based clustering methods such as hierarchical clustering can be applied to cluster genes in the

dataset. Essentially, the unsupervised grouping approach clusters genes according to their similarities of involvement in triplets.

2.2.4 Constructing the module-level hypergraph

Using either supervised or unsupervised approach, the module memberships of genes are obtained. The next step is to replace each gene in the triplet list by its module label. In the case that some genes may have multiple module labels due to multi functionality in supervised grouping, the involved triplets are duplicated in order to preserve the multi-functional information as discussed in Section 2.2.3.

At this stage, the module-level triplet list forms an edge list for a 3-uniform hypergraph, in which modules are vertices and triplets are hyperedges. The three-uniform hypergraph is undirected but weighted, as there can be many gene triplets establishing the connections between three modules. Consequently, three types of edges - those connecting three different modules, two different modules or only one module, exist in the hypergraph. These correspond to cases that the original three genes in a triplet are divided into three modules, two modules or a single module. Therefore, the 3-uniform hypergraph allows self-loops. Summing up all identical module-level triplets, the counts for each unique module-level triplet can serve as the weight of the corresponding hyperedge. Given the size difference of the modules, we transform the edge weights from simple counts to fold changes over the expected number of links if all edges are placed randomly. We then threshold the fold change to get a sparsely connected network.

2.3 Results

2.3.1 Human cutaneous melanoma dataset

We applied our methods, which require an $n \times m$ matrix as input, to the Cutaneous Melanoma RNA-seq dataset from The Cancer Genome Atlas (TCGA) (Weinstein et al., 2013). The original dataset contains 20,530 genes and 474 samples (m=474). After excluding genes with more than ten percent zero values, 15,274 genes (n=15,274) were retained for testing our method.

Each gene was first normalized using the normal score transformation as recommended in Li (2002). Before calculating LA using Equation 2.1 in Section 2.2, to satisfy the sufficient condition described in Section 2.2.1, we calculated the variance covariance matrix of all genes, obtaining a bell-shaped unimodal empirical distribution of pairwise correlations with mean $\mu \approx 0$ and standard deviation σ. Then, only pairs with a correlation contained in the interval $(\mu - c\sigma, \mu + c\sigma)$ were considered in LA calculation, where c is a small constant. In other words, no triplet would contain a pair having a correlation coefficient more than $\mu + c\sigma$ or less than $\mu - c\sigma$. For this dataset, we have $(\mu - 0.5\sigma, \mu + 0.5\sigma) = (-0.079, 0.112)$.

Applying Equation 2.1 along with the permutation selection using Equation 2.2 in Section 2.2, a total of 203,330,269 triplets were selected for this dataset at fdr=0.01. Given the information of the selected triplets, both supervised grouping and unsupervised grouping were conducted. We employed the GO term functional modules (Yoshinaga et al., 2005) as the external information for supervised grouping. A subset of informative GO terms with minimal overlap were selected using the procedure described in Yu and Li (2005). The count matrix A and its correlation matrix C, described in Section 2.2.3, were calculated, and the clusters were chosen using the technique proposed by Langfelder et al. (2008), with the minimum cluster size of 100. The final numbers of modules for the two approaches were 423 and 77, respectively.

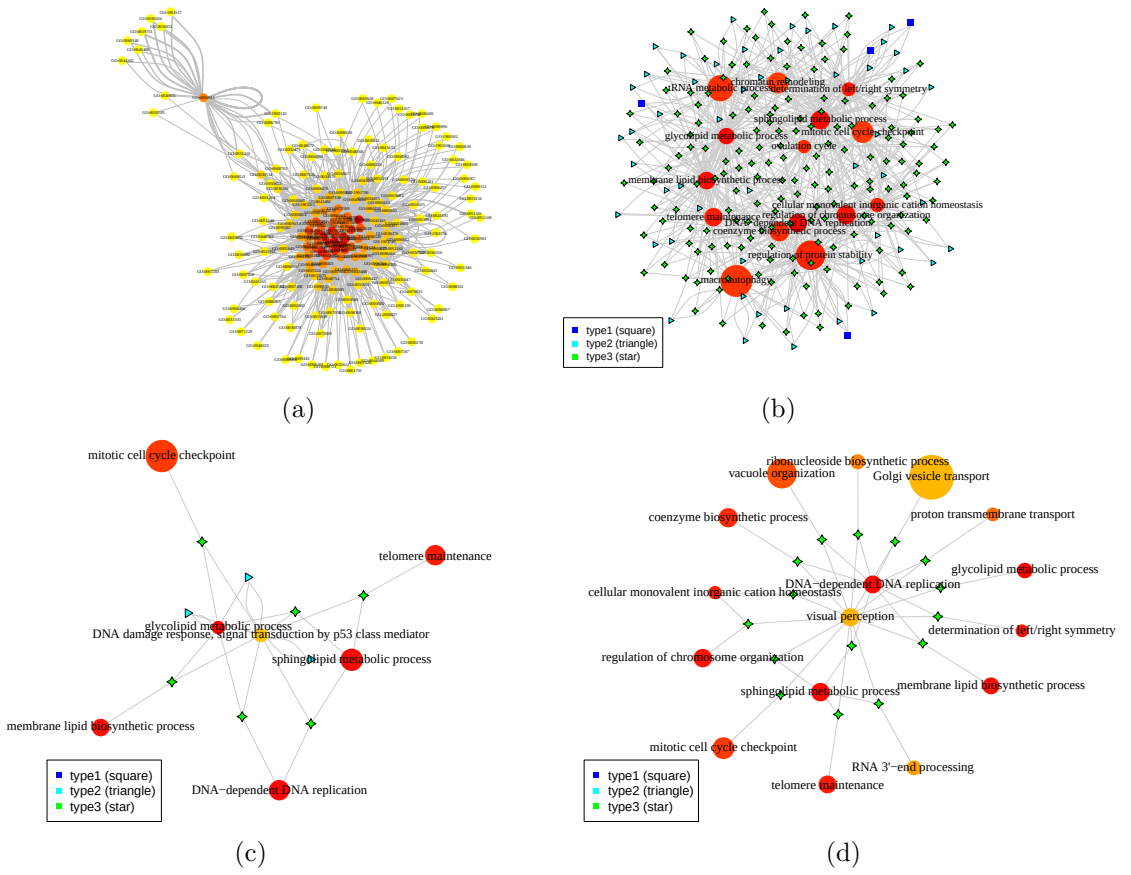


Figure 2.2: Visualization of the hypergraph for the TCGA melanoma dataset with supervised grouping. (a) The plot of the entire network, where hyperedges were reduced to binary edges for visualization; (b) Detailed plot of the top 15 most connected vertices; (c) Sub-hypergraph centered at the module "DNA damage response, signal transduction by p53 class mediator"; (d) Sub-hypergraph centered at the module "DNA dependent DNA replication". Vertex colors reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices.

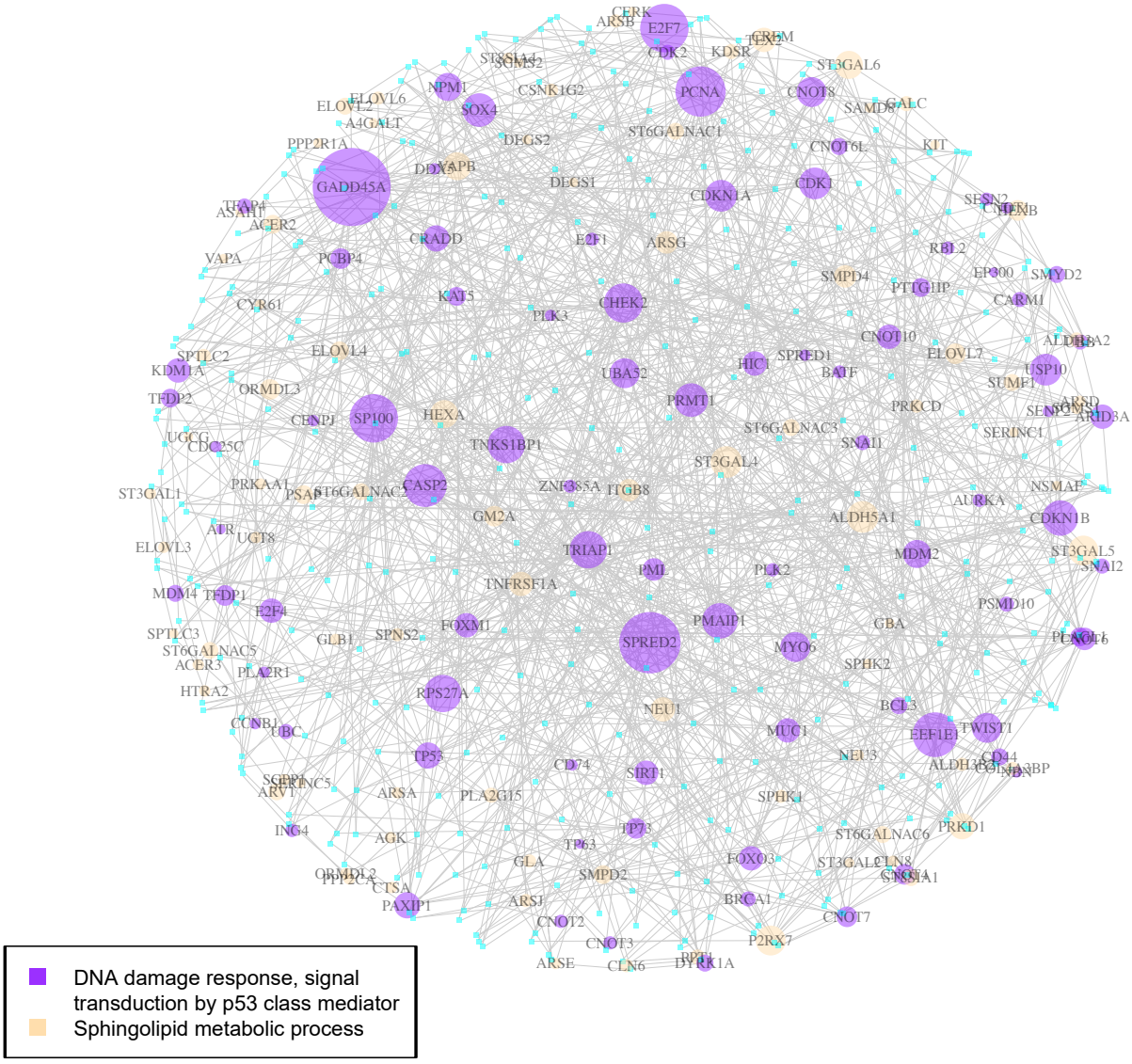


Figure 2.3: Visualization of the gene-level hypergraph of the triplet "DNA damage response, signal transduction by p53 class mediator", "DNA damage response, signal transduction by p53 class mediator", and "sphingolipid metabolic process". Vertex sizes reflect the degree, with more connected nodes larger. All gene-level hyperedges are type 2 edges.

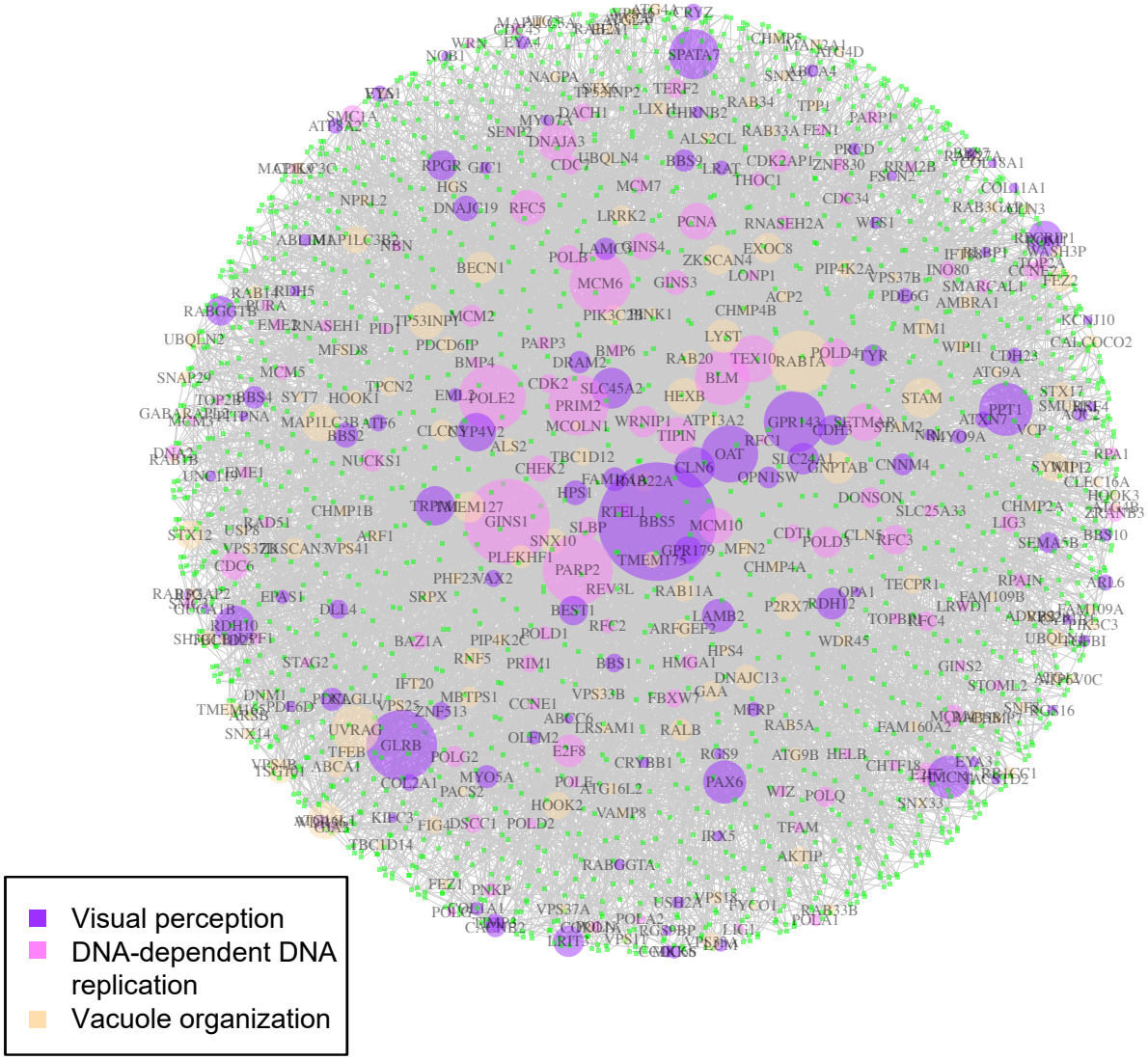
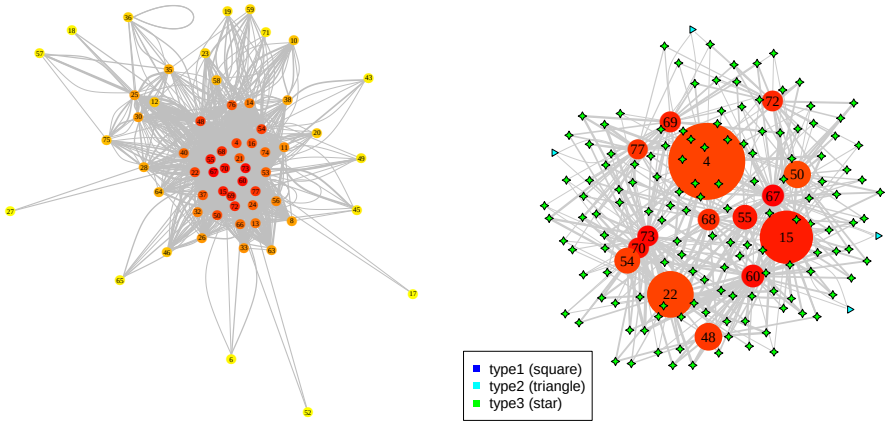


Figure 2.4: Visualization of the gene-level hypergraph of the triplet "visual perception", "DNA-dependent DNA replication", and "vacuole organization". Vertex sizes reflect the degree, with more connected nodes larger. All gene-level hyperedges are type 3 edges.



(a) (b)

Figure 2.5: Visualization of the hypergraph for the TCGA melanoma dataset with unsupervised grouping. (a) The plot of the entire network. (b) Detailed plot of the top 15 most connected vertices. Vertex colors reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices.

Table 2.1: Enrichment analysis of the human dataset for the top 15 most connected clusters. For each cluster, the enriched term that include the most number of genes in the cluster is shown.

Group label	**Group size**	**Hyperedges involved**	**GOBPID**	**Term**	**P-value**
67	109	210	GO:0030968	endoplasmic reticulum unfolded protein response	3.39E-05
			GO:0008015	blood circulation	3.81E-03
73	104	170	GO:0008202	steroid metabolic process	6.38E-04
			GO:0009063	cellular amino acid catabolic process	2.38E-03
60	114	142	GO:0007189	adenylate cyclase-activating G-protein coupled receptor signaling pathway	2.55E-05
			GO:0015698	inorganic anion transport	3.83E-04
70	104	120	GO:0072330	monocarboxylic acid biosynthetic process	3.21E-03
			GO:0019730	antimicrobial humoral response	6.04E-03
55	123	96	GO:0031349	positive regulation of defense response	3.26E-03
			GO:0043161	proteasome-mediated ubiquitin-dependent protein catabolic process	7.11E-03
15	266	83	GO:0006022	aminoglycan metabolic process	1.70E-03
			GO:0090066	regulation of anatomical structure size	2.31E-03
69	105	81	GO:0006302	double-strand break repair	1.47E-03
			GO:0001701	in utero embryonic development	2.93E-03
72	104	77	GO:0098742	cell-cell adhesion via plasma-membrane adhesion molecules	5.97E-03
			GO:0031365	N-terminal protein amino acid modification	7.54E-03
68	108	76	GO:0048608	reproductive structure development	2.43E-03
			GO:0008015	blood circulation	3.81E-03
77	100	71	GO:0051493	regulation of cytoskeleton organization	1.02E-03
			GO:0060560	developmental growth involved in morphogenesis	3.71E-03
48	138	68	GO:0051640	organelle localization	2.60E-04
			GO:0006767	water-soluble vitamin metabolic process	2.21E-03
54	129	56	GO:0051017	actin filament bundle assembly	1.26E-03
			GO:0009100	glycoprotein metabolic process	3.50E-03
4	385	51	GO:0016569	covalent chromatin modification	3.52E-03
			GO:0072331	signal transduction by p53 class mediator	4.46E-03
22	233	51	GO:1901655	cellular response to ketone	2.65E-03
			GO:0010035	response to inorganic substance	8.21E-03
50	136	49	GO:0006414	translational elongation	1.45E-03
			GO:0002791	regulation of peptide secretion	4.27E-03

Two three-uniform hypergraphs were constructed corresponding to the two grouping results. For the hypergraph with supervised grouping, edges were filtered with a minimum fold change of 2. The median number of connections for all the nodes involved in the graph is 4 (Fig.2.2a). Fig. 2.2(b) is a more detailed sub-hypergraph with the top 15 most connected vertices. The vertex color represent the number of connections of a vertex, with the most connected in red and least connected in yellow. The sizes of the vertices represent the number of genes in each module. Three types of edges were annotated corresponding to the three cases discussed in Section 2.2.4. The width of edges are proportional to their weights.

Among the top 15 most connected nodes, 5 were related to the cell cycle and DNA metabolism, indicating the tight regulation in cellular reproduction in cancer cells. Three were related to lipid metabolism, the regulation of which has been shown to play critical roles in cancer progression and metastasis (Beloribi-Djefaflia et al., 2016; Luo et al., 2017), however traditional correlation-based methods haven't shown their prominent role in expression dynamics.

To facilitate detailed examination, we examined sub-hypergraphs centered around a given vertex, together with all vertices directly connected with this vertex. As an example, Fig. 2.2 (c) shows the sub-hypergraph centered at the functional module "DNA damage response, signal transduction by p53 class mediator". Its connections involve both cell cycle modules and lipid metabolism modules. The role p53 pathway plays in lipid metabolism was only recently established (Goldstein et al., 2012). Together with the fact that three lipid metabolism modules were among the most highly connected vertices, the results suggested a prominent role of lipid metabolism pathways, including sphingolipid, glycolipid, and membrane lipid metabolism, in human cutaneous melanoma development. Interestingly, there were three type 2 hyperedges in the subgraph, two of which each had two connections to the p53 module, meaning an excess of gene triplets having two genes falling into this module.

As another example, (Fig. 2.2 (d)) shows the sub-hypergraph centered at the functional module "DNA dependent DNA replication", which is a key process in cancer cell division. Besides other cell cycle related modules, those connected with "DNA dependent DNA replication" included several modules of organization of cellular organelles, as well as several modules of transport, indicating the tight control of the cell cycle process involves much of conditional correlations between genes. Interestingly, the function "visual perception" was at a central position in this subgraph, sharing 10 hyperedges with "DNA dependent DNA replication". In the following analyses, we further explored the gene level relations of some of the hyperedges.

Fig. 2.3 shows the gene-level details of a triplet formed by the two modules "DNA damage response, signal transduction by p53 class mediator" and "sphingolipid metabolic process" in Fig. 2.2 (c). For each triplet, two of the three genes are from "DNA damage response, signal transduction by p53 class mediator" and the other one belongs to "sphingolipid metabolic process", thus all gene-level hyperedges in Fig. 2.3 are type 2 edges. Among the genes belonging to the p53 pathway, several were prominent in terms of the number of hyperedge connections. For example, GADD45A (Growth Arrest And DNA Damage Inducible Alpha) is induced by stressful growth arrest or DNA-damaging agent treatment. The gene mediates stress response by activating the p38/JNK pathway. Down-regulation of the gene increases the chemosensitivity of melanoma (Liu et al., 2018). SPRED2 (Sprouty Related EVH1 Domain Containing 2) is a member of the Sprouty/SPRED family of proteins that regulate growth factor-induced activation of the MAPK cascade, an apoptosis enhancer in melanoma (Haydn et al., 2014). E2F7 (E2F Transcription Factor 7) is among the transcription factors that regulate cell cycle progression, DNA damage repair and genomic stability. It plays a role in multiple types of cancers (Mitxelena et al., 2018).

Among the highly connected genes that belong to the sphingolipid metabolism

pathway, three were sialyltransferases - ST3GAL4 (ST3GAL4 ST3 beta-galactoside alpha-2,3-sialyltransferase 4), ST3GAL5, and ST3GAL6. Increased level of ST3GAL4 mRNA in renal cell carcinoma (RCC) has been shown to be associated with favorable prognosis (Saito et al., 2002). In hepatocellular carcinoma (HCC), the microRNA miR-26a can reduce tumor growth by suppressing the Akt/mTOR pathway through targeting ST3GAL6 (Sun et al., 2017). The role of the sialyltransferases in melanoma is yet to be elucidated.

Beside the sialyltransferases, other highly connected sphingolipid metabolism genes include ALDH5A1 (aldehyde dehydrogenase 5 family member A1), the reduced expression of which in high-grade serous ovarian cancer (HGSOC) causes the accumulation of hydroxybutyric acid (HBA) (Hilvo et al., 2016), and HEXA (hexosaminidase subunit alpha), the protein level of which was found to be increased among metastatic uveal melanoma (Linge et al., 2012).

We further examined the gene-level hypergraph of the triplet "visual perception", "DNA-dependent DNA replication", and "vacuole organization" (Fig. 2.4). Here we focus on the discussion on genes from the first GO term "visual perception", as the other two play obvious roles to melanoma development. The most highly connected gene, BBS5(Bardet-Biedl syndrome 5) has not been fully characterized, and its role in cancers not been well studied. Among other highly connected genes belonging to "visual perception", GLRB (Glycine Receptor Beta) is among the ion channel genes that is associated with the clinical outcome in breast cancer (Ko et al., 2013). GPR143 (G protein-coupled receptor 143, or OA1), codes a protein for pigmentation. SNPs in this gene have been found to be associated with the level of skin pigmentation and sun tolerance (Hernando et al., 2016). The gene is highly expressed in retinal pigment epithelium, as well as in melanoma (Bassi et al., 1995). It is involved in melanoma cell migration through the RAS/RAF/MEK/ERK signaling pathway (Bai et al., 2014). PPT1 (palmitoyl-protein thioesterase 1), is involved in the lipid-modified protein

catabolism in lysosomal degradation. Targeting PPT1 blocks mTOR signaling, which reduces tumor growth of melanoma in mouse models (Rebecca et al., 2017).

For the hypergraph with unsupervised grouping, edges were filtered with a minimum fold change of 10, which yielded a hypergraph with a median of 22 connections per node. Fig. 2.5(a) is the plot of the entire hypergraph, and Fig. 2.5(b) is a more detailed sub-hypergraph with the top 15 most connected vertices. Figure settings are identical to those in the supervised case except for the vertex names. Similar to the supervised approach, the graph is also of scale-free structure, i.e. relatively few nodes were highly connected, while most nodes were connected to few other nodes.

With the unsupervised approach, functions of each cluster of genes were unknown. Thus, only the cluster IDs are shown in the plots. To further assess the meaning of each cluster, GO enrichment analysis was conducted to determine the relevant biological functions for the clusters using GOstats (Falcon and Gentleman, 2007). The corresponding gene set enrichment results for the top 15 most connected clusters are shown in Table 2.1. The gene set enrichment analysis was limited to GO biological processes with 5 to 500 genes. For each cluster, two significant gene set that included the most number of genes in the cluster, after manual removal of obvious overlapping biological processes, are shown in Table 2.1. The results largely agreed with the supervised grouping approach to some extent. Some of the terms were related to the cell cycle and lipid metabolism themes represented by the top 15 terms in the supervised approach, e.g. “double-strand break repair”, “actin filament bundle assembly”, “regulation of cytoskeleton organization”, “translational elongation”, and “steroid metabolic process”. At the same time, more terms in Table 2.1 point to some other general themes including stress response (e.g. “endoplasmic reticulum unfolded protein response” and “proteasome-mediated ubiquitin-dependent protein catabolic process”), small molecule metabolism (e.g. “cellular amino acid catabolic process” and “water-soluble vitamin metabolic process”), structure developments (e.g. “blood

circulation" and "cell-cell adhesion via plasma-membrane adhesion molecules"), and signal transduction (e.g. "adenylate cyclase-activating G-protein coupled receptor signaling pathway" and "signal transduction by p53 class mediator").

In the unsupervised approach, genes are grouped based on their LA relation patterns with other genes. Thus genes annotated to different biological processes can be grouped together. At the same time, genes in the same biological pathway could have diverse expression activities, and be separated into different groups. Thus unsupervised approach can complement the supervised approach, painting a more complete picture of the global dynamic correlation activities.

2.3.2 Yeast cell cycle dataset

We also applied our methods to the yeast cell cycle microarray dataset (Spellman et al., 1998). The yeast dataset contains 6178 genes (n=6178), and 73 samples in four short time series and 4 control samples (m=77). For the yeast cell cycle dataset, we have restricted the pairwise correlation interval $(\mu - \sigma, \mu + \sigma) = (-0.180, 0.210)$, and a total of 3,782,460 triplets were selected for this dataset at fdr=0.2. Again both supervised grouping and unsupervised grouping were conducted. Given the smaller number of genes, for the dynamic tree cut method we used a minimum cluster size of 20. The final numbers of modules for the two approaches were 251 and 53, respectively.

For the hypergraph with supervised grouping, edges were filtered with a minimum fold change of 8. The median number of connections for all the nodes involved in the graph is 4 (Fig.2.6a). Fig. 2.6(b) is a more detailed sub-hypergraph with the top 15 most connected vertices. Beside some cell-cycle related modules, the majority of the top 15 connected modules were related to small molecule metabolism and membrane organization (2.6b). Although the dataset was generated from synchronized cell cycles, the results suggested that much of the conditional correlations happened

in metabolism, which was consistent with findings of the original LA paper (Li, 2002).

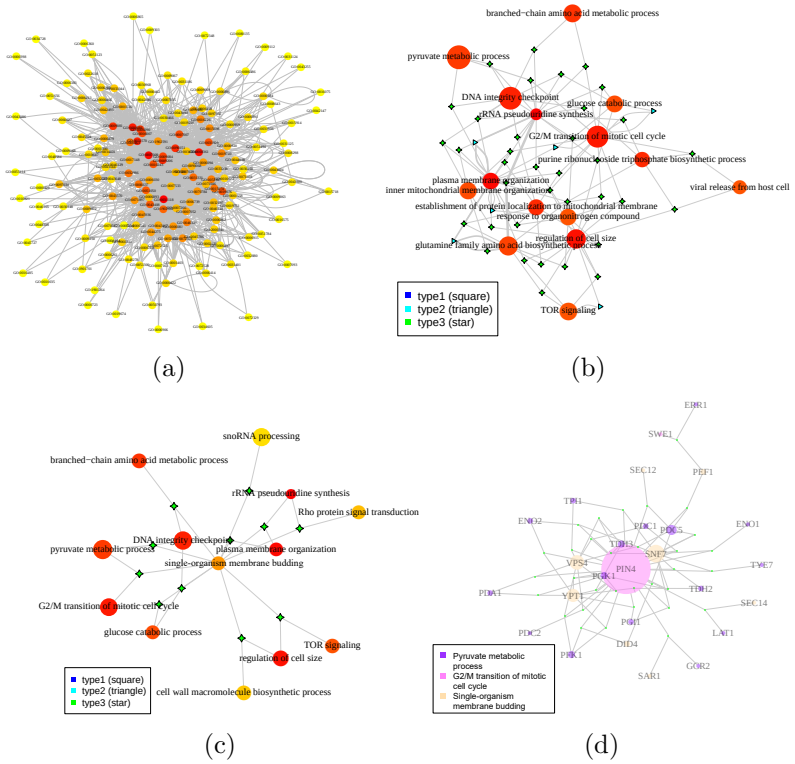


Figure 2.6: Hypergraph of the yeast cell cycle dataset with supervised grouping. (a) The plot of the entire network; (b) Detailed plot of the top 15 most connected vertices; (c) Sub-hypergraph centered at the module Single organism membrane budding; (d) Gene level hypergraph for the module triplet "single-organism membrane budding", "G2M transition of mitotic cell cycle", and "pyruvate metabolism". For the module-level hypergraph, vertex sizes reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices. For the gene-level hypergraph, vertex sizes reflect the degree, with more connected nodes larger.

Fig. 2.6 (c) shows an example sub-hypergraph centered at the functional module "Single organism membrane budding". Besides membrane and cell wall organization terms, most of the terms were related to small molecule metabolism terms. Fig. 2.6 (d) shows the gene-level details of the dynamic correlations of the triplet "Single organism membrane budding", "G2M transition of mitotic cell cycle", and "pyruvate metabolism". It is interesting that PIN4 (YBL051C) played a central role in the

graph. PIN4 functions in G2/M phase transition and DNA damage response. Its expression level didn't simply track the progression of cell cycle. In fact it was not one of the periodic genes found in the original study of Li (2002). Hence its central role in the gene-level graph was not caused by it being a proxy indicator of the cell cycle. Rather, PIN4 expression tend to be lower at the start of three of the four time series, except in the cdc15 time series. The cell cycle synchronization was conducted by blocking the cells at a certain phase of the cell cycle, which understandably put the cells in a stress state and cause irregularities in metabolism. The expression of PIN4 likely represents part of the recovery mechanism to normal growth state.

Conditioned on the level of PIN4, the correlation pattern changed between genes involved in budding and pyruvate metabolism. Three of the budding genes were prominent, SNF7 (YLR025W, vacuolar-sorting protein), VPS4 (YPR173C, vacuolar protein sorting-associated protein) and COX12 (YFL038C, cytochrome c oxidase subunit). Both SNF7 and VPS4 are involved in protein sorting (Babst et al., 1998), and both VPS4 and COX12 are involved in energy production (Taanman and Capaldi, 1992). Pyruvate is at a key intersection in metabolic network. It can be converted into carbohydrates, fatty acids, amino acid, or ethanol (Pronk et al., 1996). A number of the genes involved in pyruvate metabolism show dynamic correlations, either between themselves, or with the budding genes, indicating a change of production and utilization of pyruvate that is dependent on the cells' recovery from the unnatural blockage state as indicated by PIN4 levels. An example gene pair PFK1 (YGR240C, Alpha subunit of heterooctameric phosphofructokinase) and VPS4 are shown in Fig. 2.7. We can observe a strong inverse correlation between the low-PIN4 and high-PIN4 states.

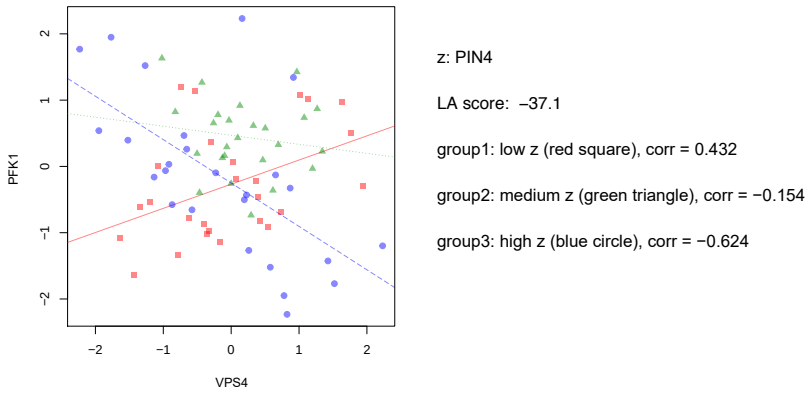
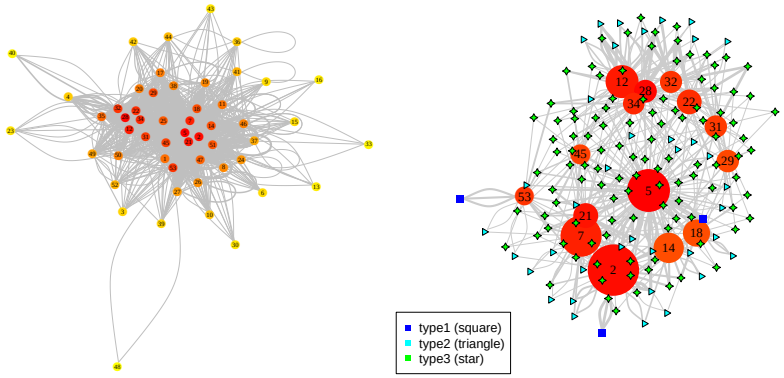


Figure 2.7: An example triplet of yeast genes. The 2-D scatter plot of expression values from PFK1 and VRS4 is given, where PIN4 serves as the "scouting gene" z. The points are divided into three groups according to the expression level of z, with low level the first 1/3 percentile, medium level the middle 1/3 percentile, and high level the last 1/3 percentile. Three lines (red solid, green dotted, and blue dashed) denote the corresponding linear regression lines of PFK1 over VRS4, given the three levels of PIN4.



(a) (b)

Figure 2.8: Visualization of the hypergraph for the yeast dataset with unsupervised grouping. (a) The plot of the entire network; (b) Detailed plot of the top 15 most connected vertices. Vertex sizes reflect the degree of connections, with more connected more red and less connected more yellow. Vertex sizes reflect the module sizes. The width of each edge is the rescaled edge weight. Three types of hypergraph edges are presented: type 1 edge connects only one vertex; type 2 edge connects two different vertices; and type 3 edge connects three different vertices.

Table 2.2: Enrichment analysis of the yeast dataset for the top 15 most connected clusters. For each cluster, the enriched term that include the most number of genes in the cluster is shown.

Group label	**Group size**	**Hyperedges involved**	**GOBPID**	**Term**	**P-value**
5	180	432	GO:0005979	regulation of glycogen biosynthetic process	1.34E-03
			GO:0016052	carbohydrate catabolic process	5.17E-03
28	46	85	GO:0032543	mitochondrial translation	2.28E-03
			GO:0072329	monocarboxylic acid catabolic process	6.64E-03
2	237	82	GO:0030154	cell differentiation	2.15E-03
			GO:0030435	sporulation resulting in formation of a cellular spore	5.63E-03
21	60	69	GO:0006259	DNA metabolic process	1.10E-09
			GO:1903047	mitotic cell cycle process	7.00E-06
12	114	61	GO:0022613	ribonucleoprotein complex biogenesis	4.25E-13
			GO:0034660	ncRNA metabolic process	1.97E-10
7	168	59	GO:0005996	monosaccharide metabolic process	7.62E-03
34	38	55	GO:0016236	macroautophagy	4.56E-03
			GO:0048193	Golgi vesicle transport	9.15E-03
22	59	52	GO:0055114	oxidation-reduction process	3.84E-05
			GO:0071822	protein complex subunit organization	2.70E-04
32	41	48	GO:0042254	ribosome biogenesis	2.77E-15
			GO:0034470	ncRNA processing	1.02E-09
53	20	48	GO:0032465	regulation of cytokinesis	1.28E-03
			GO:0045839	negative regulation of mitotic nuclear division	9.30E-03
31	43	40	GO:0000278	mitotic cell cycle	5.15E-07
			GO:0007010	cytoskeleton organization	3.73E-04
45	28	39	GO:0016054	organic acid catabolic process	1.21E-03
			GO:0015074	DNA integration	2.56E-03
29	44	36	GO:0034637	cellular carbohydrate biosynthetic process	2.43E-03
			GO:0055114	oxidation-reduction process	6.09E-03
14	96	35	GO:0006412	translation	1.64E-32
			GO:0022613	ribonucleoprotein complex biogenesis	2.99E-16
18	75	35	GO:0007030	Golgi organization	5.04E-04
			GO:0051784	negative regulation of nuclear division	2.15E-03

For the hypergraph with unsupervised grouping, edges were filtered with a minimum fold change of 4, which yielded a median of 20 connections per node involved in the graph (Fig. 2.8a). Fig. 2.8(b) is a more detailed sub-hypergraph with the top 15 most connected vertices. The enrichment results for the top 15 most connected clusters are shown in Table 2.2. Four of the top 15 clusters were dominated by cell cycle processes (e.g. "mitotic cell cycle" and "regulation of cytokinesis"). In addition, three of the terms were dominated by protein synthesis (e.g. "translation" and "ribosome biogenesis"). The other clusters were mostly dominated by small molecule metabolism/transport (e.g. "oxidation-reduction process" and "organic acid catabolism"), especially in relation to carbohydrate and energy (e.g. "regulation of glycogen biosynthetic process" and "monosaccharide metabolic process"). These results largely agreed with those from the supervised approach.

2.4 Discussion

The method involves several hyper-parameters. To calculate the LA score of a triplet, we tried to create a sufficient condition according to Theorem 1 of Ho et al. (2011), to discover "real" dynamic correlation. It requires that any pair of genes should not be linearly associated in a triplet. Thus, the threshold c is a hyper-parameter controlling how strict the user wants to obey the sufficient condition. If c is too small, one can hardly find triplets as few pairs would have strictly zero correlation from the sample correlation perspective. However, if c is too large, the sufficient condition for real LA would be violated too much, leading to false discovery for the entire downstream analysis. Therefore, the constant c can be set partially heuristically to decide the trade-off. On the other hand, the sample size of the data determines the sampling variation of the Pearson's correlation between pairs of genes that are truly uncorrelated. The TCGA melanoma data contains 474 samples. Based on Fisher's

transformation of the Pearson's correlation, if two genes are truly uncorrelated, by random sampling variation, the standard deviation of their correlation value is 0.046. Thus if two genes are uncorrelated, the 95% confidence interval (CI) of their sample correlation is (-0.09, 0.09) without adjusting for multiple testing. For this dataset, as the actual average of correlation values was not exactly zero, we used $c = 0.5$ and the corresponding interval of (-0.079, 0.112), which roughly matched that of the 95% CI. Similarly, the yeast cell cycle data contains 77 samples, which means if two genes are truly uncorrelated, by random sampling variation, the standard deviation of their correlation value is 0.114. Hence the 95% CI of the sample correlation coefficient if two genes are uncorrelated is (-0.22, 0.22). We used $c = 1$ that yielded an interval of (-0.18, 0.21), which again roughly matched the 95% CI while allowing the mean to be non-zero.

Another important parameter is the selection of fold change threshold to generate the module-level graph. As the fold change threshold increases, more connection information would be lost, though the hypergraph would be less dense and easier to investigate. Hence, similar to the correlation threshold c, the fold change threshold is also a user-specified parameter to balance the trade-off between information cleanness and completeness. In practice, we selected fold change thresholds such that the median of the degrees of the modules was 4 in the supervised case, where hundreds of modules were involved. For the unsupervised results, as roughly 50 modules were involved, we selected fold change thresholds such that the median degree was around 20. These choices made it easy to visually inspect the resulting graphs.

In this manuscript, we proposed two routes of data analysis, the supervised approach and the unsupervised approach. The supervised approach relies on existing annotations of the genes to determine the modules, while the unsupervised approach uses the gene-level connection patterns to group genes into modules. As we have seen in the results, the two approaches generated results that largely agree, while each

provided insights that complement the other approach. The supervised approach was generally easier to interpret. It allowed us to focus on important biological processes, such as the p53 pathway in the melanoma data. For a poorly annotated species, the unsupervised approach will help group genes that share similar LA relations. If genes are poorly annotated, this grouping can potentially shed light on their functional relations, and may help their functional annotation based on other genes in the same module that are well annotated.

2.5 Conclusion

We presented a method to examine dynamic correlations in an unbiased manner at the transcriptomic scale. It uses an inference framework to defend against false positives, and reduces the large amounts of triplets into a manageable hypergraph that can be visually examined relatively easily. Complimenting existing correlation-based and partial correlation-based network construction methods, the new method provides a useful tool for users to study dynamic relations in gene expression profiling datasets.

Chapter 3

GEDFN: Graph-Embedded Deep Feedforward Network

3.1 Introduction

With the clear evidence mentioned in Section 1.3 that gene networks can lead to novel variants of traditional classifiers, we are motivated to incorporate gene networks with deep feedforward networks (DFN), which is closely related to the state-of-the-art technique deep learning (LeCun et al., 2015). Although nowadays deep learning has been constantly shown to be one of the most powerful tools in classification, its application in bioinformatics is limited (Min et al., 2016). This is due to many reasons including the $n \ll p$ issue, the large heterogeneity in cell populations and clinical subject populations, as well as inconsistent data characteristics across different laboratories, resulting in difficulties merging datasets. Consequently, the relatively small number of samples compared to the large number of features in a gene expression dataset obstructs the use of deep learning techniques, where the training process usually requires a large amount of samples such as in image classification (Russakovsky et al., 2015). Therefore, there is a need to modify deep learning models for disease outcome classification using gene expression data, which naturally leads us to the development a variant of deep learning models specifically fitting the practical situation with the help of gene networks.

Incorporating gene networks as relational information in the feature space into DFN classifiers is a natural option to achieve sparse learning with less parameters compared to the usual DFN. However, to the best of our knowledge, few existing work has been done on this track. Bruna et al. (2013); Henaff et al. (2015) started the direction of sparse deep neural networks for graph-structured data. The authors developed hierarchical locally connected network architectures with newly defined convolution operations on graph-structured data. The methods have novel mathematical formulation, however, the applications are yet to be generalized. In both of the two papers, by using the two benchmark datasets MINST (LeCun and Cortes, 2010) and ImageNet (Russakovsky et al., 2015) respectively, the authors have treated

2-D grid images as a special form of graph-structured data in their experiments. This is based on the fact that an image can be regarded as a graph in which each pixel is a vertex connected with four neighbors in the four directions. However, graph-structured data can be much more complex in general, as the degree of each vertex can vary widely, and the edges do not have orientations as in image data. For a gene network, the degree of vertices is power-law distributed as the network is scale-free (Kolaczyk, 2009). In this case, convolution operations are not easy to define. In addition, with tens of thousands of vertices in the graph, applying multiple convolution operations results in huge number of parameters, which easily leads to over-fitting given the small number of training samples. By taking an alternative approach of modifying a usual DFN, our newly proposed graph-embedded DFN can serve as a convenient tool to fill the gap. It avoids over-fitting in the $n \ll p$ scenario, as well as achieves good feature selection results using the structure of the feature space.

3.2 Methodology

3.2.1 Deep feedforward networks

A deep feedforward network (DFN, or *deep neural network* (DNN), *multilayer perceptron* (MLP)) with l hidden layers has a standard architecture

$$
\begin{aligned}
Pr(\mathbf{y}|\mathbf{X},\theta) &= f(\mathbf{Z}_{out}\mathbf{W}_{out} + \mathbf{b}_{out}) \\
\mathbf{Z}_{out} &= \sigma(\mathbf{Z}_l\mathbf{W}_l + \mathbf{b}_l) \\
&\cdots \\
\mathbf{Z}_{k+1} &= \sigma(\mathbf{Z}_k\mathbf{W}_k + \mathbf{b}_k) \\
&\cdots \\
\mathbf{Z}_1 &= \sigma(\mathbf{X}\mathbf{W}_{in} + \mathbf{b}_{in}),
\end{aligned}
$$

where $\mathbf{X} \in \mathcal{R}^{n \times p}$ is the input data matrix with n samples and p features, $\mathbf{y} \in \mathcal{R}^n$ is the outcome vector containing classification labels, θ denotes all the parameters in the model, $\mathbf{Z}_{out}$ and $\mathbf{Z}_k, k = 1, \ldots, l-1$ are hidden neurons with corresponding weight matrices $\mathbf{W}_{out}$, $\mathbf{W}_k$ and bias vectors $\mathbf{b}_{out}$, $\mathbf{b}_k$. The dimensions of $\mathbf{Z}$ and $\mathbf{W}$ depend on the number of hidden neurons h_{in} and $h_k, k = 1, \ldots, l$, as well as the input dimension p and the number of classes h_{out} for classification problems. In this paper, we mainly focus on binary classification problems hence the elements of $\mathbf{y}$ simply take binary values and $h_{out} \equiv 2$. $\sigma(\cdot)$ is the activation function such as sigmoid, hyperbolic tangent (tanh) or rectifiers. $f(\cdot)$ is the softmax function converting values of the output layer into probability prediction i.e.

$$p_i = f(\mu_{i1}) = \frac{e^{\mu_{i1}}}{e^{\mu_{i0}} + e^{\mu_{i1}}}$$

where

$$\begin{aligned} p_i &:= Pr(y_i = 1 | \mathbf{x}_i) \\ \mu_{i0} &:= [\mathbf{z}_i^{(out)}]^T \mathbf{w}_0^{(out)} + b_0^{(out)} \\ \mu_{i1} &:= [\mathbf{z}_i^{(out)}]^T \mathbf{w}_1^{(out)} + b_1^{(out)}, \end{aligned}$$

for binary classification where $i = 1, \ldots, n$.

The parameters to be estimated in this model are all the weights and biases. For a training dataset given true labels, the model is trained using a stochastic gradient decent (SGD) based algorithm (Goodfellow et al., 2016) by minimizing the cross-entropy loss function

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \{y_i log(\hat{p}_i) + (1 - y_i) log(1 - \hat{p}_i)\},$$

where again θ denotes all the model parameters, and $\hat{p}_i$ is the fitted value of p_i. More

details about DFN can be found in Goodfellow et al. (2016).

3.2.2 Graph-embedded deep feedforward networks

Our newly proposed DNN model is based on two main assumptions. The first assumption is that neighboring features on a known scale-free feature network or feature graph[1] tend to be statistically dependent. The second assumption is that only a small number of features are true predictors for the outcome, and the true predictors tend to form cliques in the feature graph. These assumptions have been commonly used and justified in previous works reviewed in Section 1.3.

To incorporate the known feature graph information to DNN, we propose the graph-embedded deep feedforward network (GEDFN) model. The key idea is that, instead of letting the input layer and the first hidden layer to be fully connected, we embed the feature graph in the first hidden layer so that a fixed informative sparse connection can be achieved.

Let $\mathbf{G} = (V, E)$ be a known graph of p features, with V the collection of p vertices and E the collection of all edges connecting vertices. A common representation of a graph is the corresponding adjacency matrix A. Given a graph $\mathbf{G}$ with p vertices, the adjacency A is a $p \times p$ matrix with

$$A_{ij} = \begin{cases} 1, & \text{if } V_i \text{ and } V_j \text{ are connected, } \forall i, j = 1, \ldots, p \\ 0, & \text{otherwise.} \end{cases}$$

In our case A is symmetric since the graph is undirected. Also, we require $A_{ii} = 1$ meaning each vertex is regarded to connecting itself.

Now to mathematically formulate our idea, we construct the DNN such that the

[1]Since in this chapter we interchangeably discuss feature networks and neural networks, to avoid confusion, the equivalent term "graph" is used to refer to the feature network from now on, while "networks" naturally refer to neural networks.

dimension of the first hidden layer (h_{in}) is the same as the original input i.e. $h_{in} = p$, hence $\mathbf{W}_{in}$ has a dimension of $p \times p$. Between the input layer $\mathbf{X}$ and the first hidden layer $\mathbf{Z}_1$, instead of fully connecting the two layers with $\mathbf{Z}_1 = \sigma(\mathbf{X}\mathbf{W}_{in} + \mathbf{b}_{in})$, we have

$$\mathbf{Z}_1 = \sigma(\mathbf{X}(\mathbf{W}_{in} \odot A) + \mathbf{b}_{in}) \tag{3.1}$$

where the operation $\odot$ is the Hadamard (element-wise) product. Thus, the connections between the first two layers of the feedforward network are "filtered" by the feature graph adjacency matrix. Through the one-to-one $\mathcal{R} : p \to p$ transformation, all features have their corresponding hidden neurons in the first hidden layer. A feature can only feed information to hidden neurons that correspond to features connecting to it in the feature graph.

Specifically, let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T, i = 1, \ldots, n$ be any instance (one row) of the input matrix $\mathbf{X}$, in the usual DFN, the first hidden layer of this instance is calculated as

$$\mathbf{z}_i^{(1)} = \sigma([\sum_{j=1}^{p} x_{ij} w_{1j}^{(in)} + b_1^{(in)}, \ldots, \sum_{j=1}^{p} x_{ij} w_{h_{in}j}^{(in)} + b_{h_{in}}^{(in)}]^T),$$

where $\mathbf{z}_i^{(1)}$ is the i-th row of $\mathbf{Z}_1$, and $w_{kj}^{(in)}, b_k^{(in)}, k = 1, \ldots, h_{in}$ are the weight and bias for this layer. Now in our model, $h_{in} = p$ and each $w_{kj}^{(in)}$ is multiplied by an indicator function i.e.

$$\mathbf{z}_i^{(1)} = \sigma([\sum_{j=1}^{p} x_{ij} w_{1j}^{(in)} \mathcal{I}(A_{1j} = 1) + b_1^{(in)}, \ldots, \sum_{j=1}^{p} x_{ij} w_{pj}^{(in)} \mathcal{I}(A_{pj} = 1) + b_p^{(in)}]^T).$$

Therefore, the feature graph helps achieve sparsity for the connection between the input layer and the first hidden layer.

3.2.3 Evaluation of feature importance

Beside improving classification, it is also of great interest to find features that significantly contribute to the classification, as they can reveal the underlying biological mechanisms. Therefore, for GEDFN, we also develop a feature ranking method according to a relative importance score. The idea is analogous to the Connection Weights (CW) method introduced by Olden and Jackson (2002). Extended from CW, we propose the Graph Connection Weights (GCW) method, which emphasizes the significance of the feature graph in our newly proposed neural network architecture.

The main idea of GCW is that, the contribution of a specific variable is directly reflected by the magnitude of all the weights that directly associated with the corresponding hidden neuron in the graph-embedded layer (the first hidden layer). Summing over the absolute values of the directly associated weights gives the relative importance of the specific feature, i.e.

$$s_j = \gamma_j \sum_{k=1}^{p} |w_{kj}^{(in)} \mathcal{I}(A_{kj} = 1)| + \sum_{m=1}^{h_1} |w_{jm}^{(1)}|, \tag{3.2}$$

$$\gamma_j = min(c/\sum_{k=1}^{p} \mathcal{I}(A_{kj} = 1), 1), j = 1, \ldots, p, \tag{3.3}$$

where s_j is the importance score for feature j, $w^{(in)}$ denotes weights between the input and first hidden layers, and $w^{(1)}$ denotes weights between the first hidden layer and the second hidden layer. A constant c is imposed to penalize feature vertices with too many connections, so that they will not be overly influential. In subsequent experiments, we take $c = 50$.

Note that the importance score consists of two parts according to Equation 3.2. The left term summarizes the importance of a feature according to the connection on the feature graph, coherent with the property of the graph-embedded layer. The right term then summarizes the contribution of the feature according to the connection to

the hidden neurons in the next fully-connected layer. Input data are required to be Z-score transformed (the original value minus the mean across all samples and then divided by the standard deviation) before entered into the model, and this will guarantee all variables are of the same scale so that the magnitude of weights are comparable. After training GEDFN, the importance scores for all the variables can be calculated using trained weights, which leads to a ranked feature list.

3.2.4 Detailed model settings

For the choice of activation functions in GEDFN, the rectified linear unit (ReLU) (Nair and Hinton, 2010) with the form (in scalar case)

$$\sigma_{ReLU}(x) = max(x, 0)$$

is employed. This activation has an advantage over sigmoid and tanh as it can avoid the vanishing gradient problem (Hochreiter et al., 2001) during training using SGD. To train the model, we choose the Adam optimizer (Kingma and Ba, 2014), which is the most widely used variant of traditional gradient descent algorithms in deep learning. Also, we use the mini-batch training strategy by which the optimizer randomly trains a small proportion of the samples in each iteration. Details about the Adam optimizer and mini-batch training can be found in Goodfellow et al. (2016); Kingma and Ba (2014).

The classification performance of a DNN model is associated with many hyper-parameters, including architecture-related parameters such as the number of layers and the number of hidden neurons in each layer, regularization-related parameters such as the dropout proportion, and model training-related parameters such as the learning rate and the batch size. These hyper-parameters can be fine-tuned using advanced hyper-parameter training algorithm such as Bayesian Optimization (Mockus,

2012), however, as the hyper-parameters are not of primary interest in our work, in later sections, we simply tune them using grid search in a feasible hyper-parameter space. A visualization of our tuned GEDFN model for simulation and real data experiments is shown in Fig. 3.1. More details of hyper-parameter tuning can be found in A.

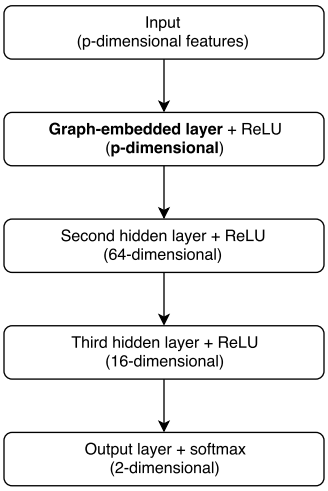


Figure 3.1: Network architecture of the GEDFN model for experiments in Section 3.3 and Section 3.4.

3.3 Simulations

We conducted extensive simulation experiments to mimic disease outcome classification using gene expression and network data, and explored the performance of our new method in comparison with the usual DFN and other proven methods. Robustness was also tested by simulating datasets that did not fully satisfy the main assumptions. The method was applied to examine whether it could still achieve a reasonable performance.

3.3.1 Synthetic data generation

For a given number of features p, we employed the preferential attachment algorithm proposed by Barabási and Albert (1999) to generate a scale-free feature graph. The $p \times p$ distance matrix D recording pairwise distances among all vertices was then calculated. Next, we derived the covariance matrix Σ by transforming the distances between verticies by letting

$$\Sigma_{ij} = 0.7^{D_{ij}}, i, j = 1, \ldots, p.$$

Here by convention the diagonal elements of D are all zeros meaning the distance between a vertex to itself is zero.

After simulating the feature graph and obtaining the covariance matrix of features, we generate n multivariate Gaussian samples as the input matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ i.e.

$$\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \Sigma), i = 1, \ldots\ n,$$

where $n \ll p$ for imitating gene expression data. Using this setup, vertices that are several steps away could naturally become negatively correlated when we sample the expression values from multivariate normal distribution using Σ as the variance-covariance matrix. Figure A.1 in A shows sample plots of the pairwise feature correlation distributions for the simulated data.

To generate outcome variables, we first select a subset of features to be the "true" predictors. Following our assumptions mentioned in Section 3.2.2, we intend to select cliques in the feature graph. Among vertices with relatively high degrees, part of them are randomly selected as "cores", and part of the neighboring vertices of cores are also selected. Denoting the number of true predictors as p_0, we sample a set of parameters $\beta = (\beta_1, \ldots, \beta_{p_0})^T$ and an intercept β_0 within a certain range. In our experiments, we first uniformly sample β's from $(0.1, 0.2)$, and randomly turn some

of the parameters into negative, so that we accommodate both positive and negative coefficients. Finally, the outcome variable $\mathbf{y}$ is generated through a generalized linear model framework

$$Pr(y_i = 1|\mathbf{x_i}) = \eta^{-1}(\mathbf{x_i}^T\beta + \beta_0)$$
$$y_i = \mathcal{I}(Pr(y_i = 1|\mathbf{x_i}) > t), i = 1, \dots n,$$

where t is a threshold and $\eta(\cdot)$ is the link function. We consider two cases of $\eta^{-1}(\cdot)$in our experiments, one is the sigmoid function, which is equivalent to the binary softmax and monotone

$$\eta^{-1}(x) = \frac{1}{1+e^x}$$

and the other is a weighted tanh plus quadratic function, which is non-monotone

$$\eta^{-1}(x) = 0.7\phi(tanh(x)) + 0.3\phi(x^2),$$

where $\phi(\cdot)$ is the min_max function scaling the input to $[0, 1]$.

Following the above procedure, corresponding to the two cases of inverse link functions, we simulate two sets of synthetic datasets with 5,000 features and 400 samples. We compare our method with the usual DFN, the feature graph-embedded classification method network-guided forest (NGF) (Dutkowski and Ideker, 2011) mentioned in Section 1.3, as well as the traditional logistic regression with lasso (LRL) (Tibshirani, 1996). In gene expression data, the number of true predictors account for only a small proportion of the features. Taking this aspect into consideration, we examine different numbers, i.e. 40, 80, 120, 160 and 200, of true predictors, corresponding to 2, 4, 6, 8, and 10 cores among all the high-degree vertices in the feature graph. However, in reality, the true predictors may not be perfectly distributed in the feature graph as cliques. Instead, some of the true predictors, which we call "singletons", can

be quite scattered. To create this possible circumstance, we simulate three series of datasets with singleton proportions 0%, 50% and 100% among all the true predictors. Therefore, we investigate three situations where all true predictors are in cliques, half of the true predictors are singletons, and all of the true predictors are scattered in the graph, respectively.

3.3.2 Simulation results and discussion

In our simulation studies, as shown in Fig. 3.1, the GEDFN had three hidden layers, where the first hidden layer was the graph adjacency embedded layer. Thus the dimension of its output is the same as the input, namely 5,000. The second and third hidden layers had 64 and 16 hidden neurons respectively, which are the same for the usual DFN. The number of the first layer hidden neurons in the usual DFN, 1024 neurons, was selected using grid search.

For each of the data generation settings, ten independent datasets were generated, and the GEDFN, DFN, NGF and LRL methods were applied. For each simulated dataset, we randomly split the dataset into training and testing sets at a 4:1 ratio. The models were trained using the training dataset, and used to predict the class probabilities of the testing dataset. To evaluate the classification results, receiver operating characteristic (ROC) curve was generated using the predicted class probabilities and the true class membership of the testing dataset, and the area under the curve (AUC) was calculated. The final testing results were then averaged across the ten datasets.

Fig. 4.2 shows the results of the case with the sigmoid inverse link function. The error bars denote intervals of estimated mean AUC values plus/minus their standard errors. Corresponding to the case that singleton proportion is 0%, Fig. 4.2(a) shows GEDFN and LRL outperformed the other two methods. As the number of true predictors increased, all of the methods performed better as there were more signals

in the feature set. As the singleton proportion increased to 0.5 (Fig. 4.2(b)), GEDFN was the best among the four though the difference between GEDFN and LRL was not big. In Fig. 4.2(c), when the singleton proportion was increased to 1, all of the methods performed worse, but GEDFN performed better than the others overall. The close results of GEDFN and LRL were expected, as in the sigmoid case LRL was in fact the true model.

As for feature selection, GEDFN uses Equation 3.2 to rank features. The feature ranking method for the usual DFN was similar to the one for GEDFN, except that for DFN each variable's importance was given only by the second term in Equation 3.2, that was to consider only the weights connecting the input layer and the first hidden layer. For NGF, the variable importance calculation based on cumulative reduction of Gini impurity in random forests (Breiman, 2001) could be directly applied. Therefore, knowing the true predictors for simulated data, we were able to compare feature selection results for different methods by computing and comparing the AUC of the precision-recall curves, which were constructed using the feature ranking of the models and the 0/1 vector indicating the true predictor status of each feature. Fig. 4.2(d)(e)(f) show the average precision-recall AUC (error bars: the intervals of mean AUC plus/minus one standard error) for each simulation setting of the sigmoid case. We found that DFN was not able to effectively rank features, resulting in precision-recall AUC less than 0.05 for all the datasets, and thus they were not included in the plots. From Fig. 4.2(d)(e)(f), one can conclude that GEDFN ranked features more effectively than NGF.

LRL did not rank features but directly gave the selected feature subset based on cross-validation. To compare feature selection between GEDFN and LRL, for each dataset, we fixed the precision of GEDFN to be the same as LRL, and then compared their recall values. The recall plots (error bars: the intervals of mean recall plus/minus one standard error) for different simulation settings are shown in Fig. 4.2(g)(h)(i).

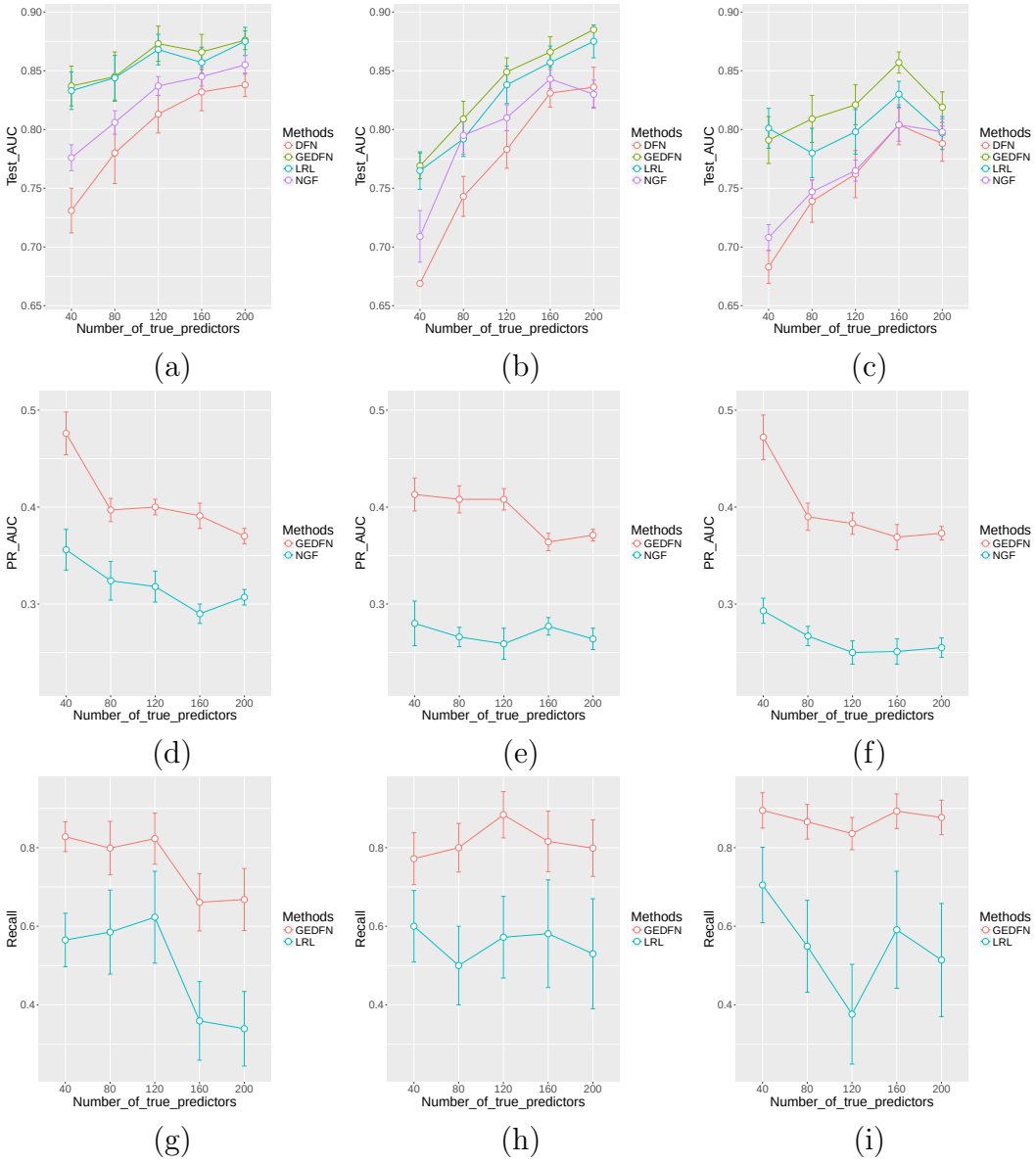


Figure 3.2: Plots of the classification and feature selection comparison for the case with the sigmoid inverse link function. Singleton proportions: left column 0%, middle column 50%, right column 100%. First row: AUC of ROC for classification; second row: AUC of precision-recall for feature selection; third row: recall plots given fixed precision from LRL. Error bars represent the estimated mean quantity plus/minus the standard error.

Again, it is evident that GEDFN achieved better feature selection results.

Simulation results for the case with the weighted tanh plus quadratic inverse link function are shown in Fig. 3.3. From the first row of Fig. 3.3, all the methods' AUC decreased compared to their counterparts in the case of sigmoid inverse link, as the non-monotone function brought more difficulty to classification. However, GEDFN again outperformed the other methods in general, and the difference between GEDFN and LRL was enlarged compared to the sigmoid function case since the non-monotone inverse link was more challenging, and LRL was no longer the true model in this case. The second row and third row of Fig. 3.3 indicate GEDFN's better feature selection than NGF and LRL across all simulation settings in this case. DFN was again proved not to have good feature selection capability through the experiment, with precision-recall AUC no more than 0.04.

The above simulation experiment results showed nice performance of GEDFN in both classification accuracy and feature selection in both the sigmoid case and the tanh plus quadratic case. The method was robust across different number of true predictors and different proportions of singletons in feature graphs. To further test the robustness of GEDFN, we considered cases that the known feature graph was completely misspecified, i.e. the graph structure bears misleading information with regard to feature correlation and true predictor location. This extreme situation is unlikely in applications. We employed the synthetic datasets used above with singleton proportion 50%, destroyed the true feature graphs, and re-constructed random feature graphs using the preferential attachment algorithm. The comparison of classification and feature selection between the GEDFN with correct feature graph and the GEDFN with misspecified graphs is shown in Figure A.2 in A. From the results, misspecified feature graphs negatively affected GEDFN regarding both classification and feature selection. For classification, GEDFN was robust enough to obtain acceptable accuracies. In contrast, feature selection was more influenced, which was

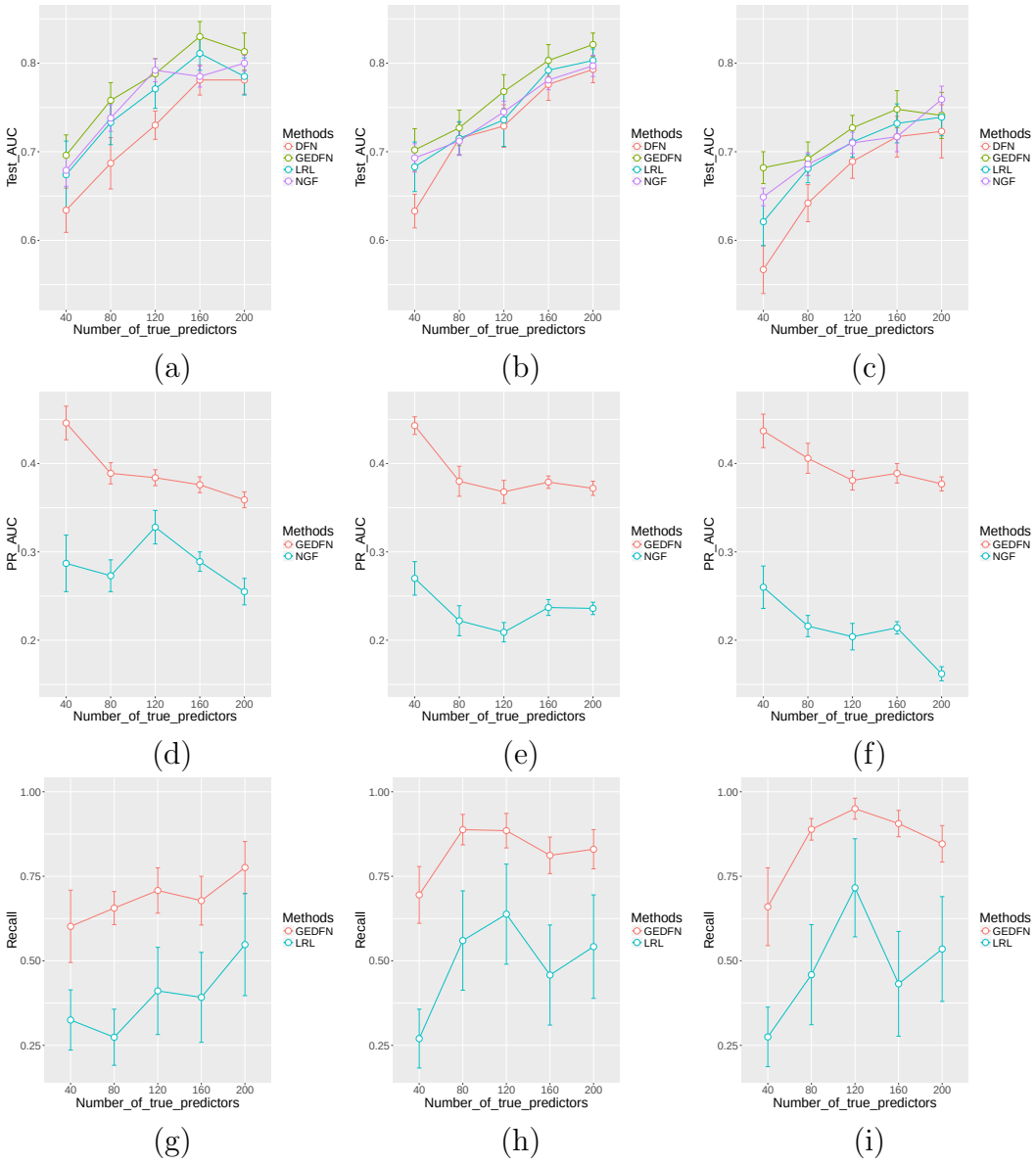


Figure 3.3: Plots of the classification and feature selection comparison for the case with the weighted tanh plus quadratic inverse link function. Singleton proportions: left column 0%, middle column 50%, right column 100%. First row: AUC of ROC for classification; second row: AUC of precision-recall for feature selection; third row: recall plots given fixed precision from LRL. Error bars represent the estimated mean quantity plus/minus the standard error.

expected as the feature ranking mechanism of GEDFN relied on the feature graph connections.

Another concern about the robustness of GEDFN is the reproducibility of feature selection. For a fixed dataset, we were interested in whether a relatively stable set of important features would be selected across different times of model fitting. To explore this, we randomly chose a synthetic dataset with 40 true predictors, 50% singleton and sigmoid inverse link, and experimented GEDFN feature selection repeatedly for 10 times. Ten ranked feature lists were obtained, and the top 40 ranked variables were selected for each experiment. Among the ten sets of 40 selected features, 19 features were repeatedly selected as top 40 over seven times, and they covered 40% of the 40 true predictors. Also, 70% of the union of the ten sets of top 40 features turned out to be relevant for prediction. Here "relevant" means a feature was either a true predictor, or a neighbor of a true predictor in the feature graph, since in our simulation settings, neighbors of true predictors can be useful in classification even if they were not chosen as true predictors themselves. This small specific experiment indicated the relative stable performance of GEDFN feature selection.

3.4 Data Analysis

3.4.1 Breast invasive carcinoma data

We applied our GEDFN method to the Cancer Genome Atlas (TCGA) breast cancer (BRCA) RNA-seq dataset (Koboldt et al., 2012). The dataset consisted of a gene expression matrix with 20,532 genes of 707 cancer patients, as well as the clinical data containing various disease status measurements. The gene network came from the HINT database (Das and Yu, 2012). We were interested in the relation between gene expression and a molecular subtype of breast cancer - the tumor's Estrogen Receptor (ER) status. ER is expressed in more than 2/3 of breast tumors, and plays

Table 3.1: Classification results for BRCA data

Methods	GEDFN	DFN	NGF	LRL
Mean AUC	0.945	0.938	0.922	0.940
Standard deviation	0.005	0.013	0.012	0.008

a critical role in tumor growth (Sorlie et al., 2003). Elucidating the relation between gene expression pattern and ER status can shed light on the subtypes of breast cancer and their specific regulations. After screening genes that were not involved in the gene network, a total of 9,211 genes were used as the final feature set in our classification. For each gene, the expression value was Z-score transformed.

Using the HINT network architecture, we tested the four methods GEDFN, DFN, NGF and LRL on the BRCA data with ten repeated experiments respectively. The computation time of GEDFN was around 3 min each time on a workstation with dual Xeon E5-2660 processors, 256Gb RAM, and a single GTX Titan Xp GPU. The summary of test-set classification accuracies is seen in Table 3.1. From the classification results, all the methods achieved excellent AUC scores, and we concluded that the dataset contained strong signals for ER status. Thus, for this dataset, the improvement of incorporating feature graph regarding classification was limited, as traditional methods already pushed the performance to the upper bound.

However, GEDFN exhibited advantages over other methods in terms of feature selection. To analyze the feature selection results for this dataset, we first averaged the importance scores across the ten repeated model trainings from GEDFN and NGF. DFN was proved not able to achieve good feature selection results in Section 4.3.3 and thus was excluded from this analysis. For LRL, the features selected over the ten times were quite stable with only one or two different variables, hence we took the union of the 10 selected feature sets as the feature selection result for LRL. In the end, selected features from LRL and the top 1% ranked features from GEDFN and NGF were compared. They contained 89, 92 and 92 features respectively.

We invested the functional consistency of the selected features, as reflected by

Table 3.2: Selected feature sub-graph analysis for BRCA data

Methods	GEDFN	NGF	LRL
# connected components	3	4	80
Within-component average distance	3.181	3.169	1.700
Average distance	2.263	2.393	3.822

how close the selected features were in the original feature graph. On the feature graph, which was based on protein-protein interaction (Das and Yu, 2012), functionally related genes tend to be closer in distance. For each method, we extracted the sub-graph of the selected features from the entire feature graph, and examined the connection of the sub-graph. A better feature selection method was expected to choose features that fall into cliques of the overall graph, resulting in fewer connected components in the selected sub-graph. Table 3.2 shows the results of sub-graph analysis. The first row is the number of connected components for each sub-graph. The second row is the within-component average distances in the sub-graph. The third row is the average distances in the entire feature graph. From Table 3.2, one can see that features selected by GEDFN formed more closely connected sub-graphs (seen in Fig. 3.4), while NGF resulted in more scattered sub-graphs with 4 connected components. Features selected by LRL had no graph structure at all, with 89 features forming 80 connected components, meaning most of which were unconnected. The average distance in the entire feature graph for GEDFN was smaller than that for NGF, indicating the closer relationship among genes selected by GEDFN. Although the within-component average distance for LRL is the smallest, the large amount of connected components made this statistic meaningless for LRL.

Functional analysis of the genes selected by GEDFN were conducted by testing for enrichment of the Gene Ongoloty (GO) biological processes using GOstats (Falcon and Gentleman, 2007). The results can be found in Table 3.3. Fifteen of the 92 selected genes belong to the autophagy process, which was the most significant GO term. In addition, "regulation of apoptotic signaling pathway" and "ubiquitin-dependent

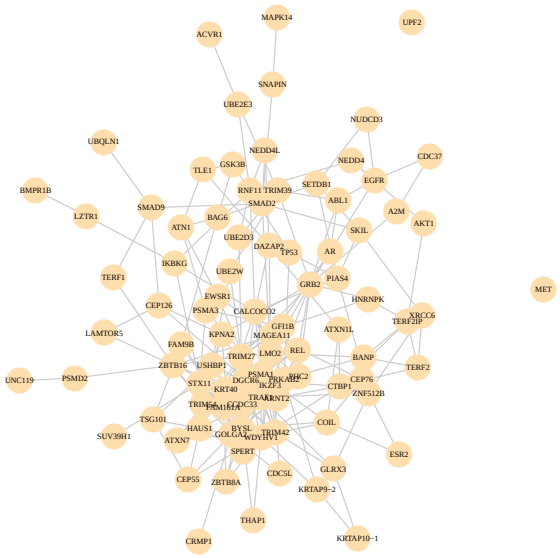


Figure 3.4: Feature sub-graph selected by GEDFN for BRCA data.

protein catabolic process" were also among the top terms. Breast cancer cells that express ERα have a higher autophagic activity than cells that express ER-β and ER- cells (Felzen et al., 2015). It has been documented that the unfolded protein response and autophagy play a role in the development of anti-estrogen therapy resistance in ER+ breast cancer (Cook and Clarke, 2014).

The second most significant term was "negative regulation of cell cycle". ERα regulates the cell cycle by regulating the S and G2/M phases in a ligand-dependent fashion (JavanMoghadam et al., 2016). Several of the top terms were signal transduction process. It has been long established that there are cross-talks between BMP and estrogen signaling, as well as between growth factor receptor pathways and estrogen signaling (Osborne et al., 2005). BMPs are repressed by estrogen through estrogen receptor signaling (Yamamoto et al., 2002). NF-κB is a crucial player in cancer initiation and progression. Direct binding to NF-κB is documented for p53 and estrogen receptor (Hoesel and Schmid, 2013). It exhibits differential function in ER- and ER+ hormone-independent breast cancer cells (Gionet et al., 2009).

The remaining top GO terms were related to stress response. Breast cancer cells

Table 3.3: Top GO biological processes for the sub-graph selected by GEDFN (BRCA data). Manual pruning of partially overlapping GO terms was conducted.

GOBPID	Pvalue	Term
GO:0006914	5.02E-07	autophagy
GO:0045786	1.16E-05	negative regulation of cell cycle
GO:0030509	1.27E-05	BMP signaling pathway
GO:2001233	1.74E-05	regulation of apoptotic signaling pathway
GO:0006511	1.78E-05	ubiquitin-dependent protein catabolic process
GO:0071363	3.01E-05	cellular response to growth factor stimulus
GO:0038061	5.56E-05	NIK/NF-kappaB signaling
GO:0097576	5.97E-05	vacuole fusion
GO:0071456	6.68E-05	cellular response to hypoxia
GO:2001020	1.69E-04	regulation of response to DNA damage stimulus

Table 3.4: GO enrichment analysis for features selected by GEDFN only (BRCA data). Manual pruning of partially overlapping GO terms was conducted.

GOBPID	Pvalue	Term
GO:2001233	4.81E-06	regulation of apoptotic signaling pathway
GO:0006511	1.12E-05	ubiquitin-dependent protein catabolic process
GO:0030509	2.39E-05	BMP signaling pathway
GO:0071363	1.24E-04	cellular response to growth factor stimulus
GO:0045786	1.89E-04	negative regulation of cell cycle

adapt to reduced oxygen concentrations by increasing levels of hypoxia-inducible factors. The increase of such factors cause higher risk of metastasis (Gilkes and Semenza, 2013). Hypoxia inducible factors can influence the expression of estrogen receptor in breast cancer cells (Wolff et al., 2017). Estrogen changes the DNA damage response by regulating proteins including ATM, ATR, CHK1, BRCA1, and p53 (Caldon, 2014). Thus it is expected that DNA damage response is closely related to ER status.

Finally, we analyzed the 69 genes that were only selected by GEDFN but not the other methods. The top five GO terms of this feature set are listed in Table 3.4. Clearly these functions agree very well with the biological processes based on all the selected genes listed in Table 3.3.

Table 3.5: Classification results for KIRC data

Methods	GEDFN	DFN	NGF	LRL
Mean AUC	0.743	0.643	0.521	0.698
Standard deviation	0.047	0.038	0.012	0.003

3.4.2 Kidney renal clear cell carcinoma data

We also tested GEDFN on the kidney renal clear cell carcinoma (KIRC) RNA-seq dataset from TCGA (Network et al., 2013). The dataset contained the gene expression matrix with 20,502 genes from 537 subjects, as well as the clinical data including survival information. The gene network again came from the HINT database. For KIRC, We tried to study the relation between gene expression and the five-year survival outcome, which was a much more difficult task compared to cancer subtypes. After screening genes that were not involved in the gene network, a total of 8,630 genes were used as the final feature set in our classification. For each gene, the expression value was again Z-score transformed.

As in Section 4.4.1, we again tested the four methods GEDFN, DFN, NGF and LRL on the KIRC data with ten repeated experiments respectively. The computation time of GEDFN was around 2.5 min each time on the same workstation as for BRCA data. Classification results are summarized in Table 3.5. Given the 5-year survival outcome variable was much more challenging to predict, the AUC scores were much lower for all the methods. NGF was not able to classify instances at all with AUC of ROC near 0.5. At the same time, GEDFN performed substantially better than the other three methods. Therefore, the KIRC data demonstrated that incorporating feature graph would improve classification accuracy for DNN models.

Due to the poor classification of NGF, it was unnecessary to examine its feature selection for KIRC. Similar to the BRCA results in Section 4.4.1, LRL selected scattered variables on the feature graph with few connections between them. For GEDFN, we obtained 86 top 1% important features that fall into 3 connected components, with

an average within-component distance of 3.111, and an average distance in the entire feature graph of 2.257. Thirty of the 86 genes overlap with the top genes in the breast cancer study, which was not a surprise given both datasets are based on tumor tissues.

The sub-graph of top 1% of genes selected by GEDFN is shown in Fig. 3.5. GO enrichment analysis was conducted for the 86 genes, and the top 10 GO terms are shown in Table 3.6. The top GO terms were predominantly regulatory and signal transduction processes, several of which were well-known for their association with tumor development. However their role in survival were previously not clear. A key regulator in the oncogenesis of renal cell carcinoma inhibits apoptosis through apoptosis signaling pathway, which was the top GO term (Banumathy and Cairns, 2010). The second GO term, regulation of binding is a relatively broad term. The selected genes associated with this term fell mostly into protein and DNA binding processes. The 17 selected genes that were in this process include known oncogenes JUN (Jones et al., 2016) and TFIP11 (Tang et al., 2015), tumor suppressors CRMP1 (Cai et al., 2017) and LDOC1 (Ambrosio et al., 2017), target of tumorcide Manumycin-A PPP1CA (Carey et al., 2015), three SMAD family proteins SMAD2/SMAD3/SMAD4 that are involved in multiple cancers (Samanta and Datta, 2012), as well as several genes involved in various other cancers, e.g. PIN1 (Cheng et al., 2016), MDF1 (Li et al., 2017), AES (Sarma and Yaseen, 2011), MAPK8 (Recio-Boiles et al., 2016), CTNNB1 (Na et al., 2017), KDM1A (Ambrosio et al., 2017), and SUMO1 (Jin et al., 2017).

The term "cellular response to growth factor stimulus" includes the epidermal growth factor receptor (EGFR) pathway, and BMP signaling pathway. Both are related to the development of renal cell cancer (Edeline et al., 2010; Zhang et al., 2016). Increased EGFR expression occurs in some renal cell carcinoma patients with an unfavorable histologic phenotype (Minner et al., 2012). Many genes in the "heart

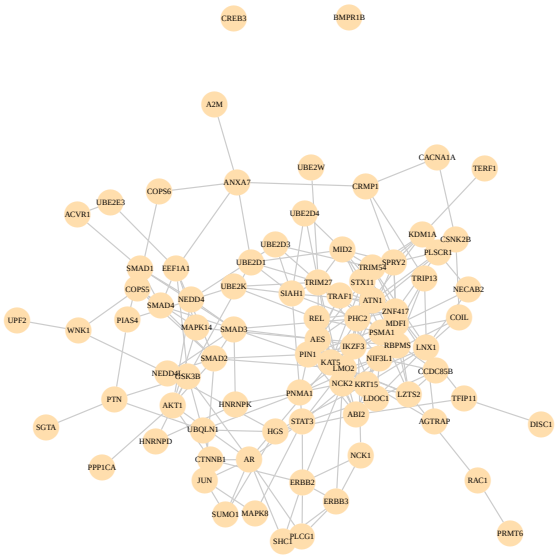


Figure 3.5: Feature sub-graph selected by GEDFN for KIRC data.

development" and "developmental growth" processes are also part of the response to growth factor stimulus, causing those terms to be significant.

The serine/threonine kinase signaling pathway includes SMAD and mTOR signal transduction, both of which are involved in renal cell cancer development (Edeline et al., 2010). Both cell proliferation regulation and ubiquitin-dependent protein catabolism are commonly affected pathways in multiple cancers. Specifically, the ubiquitin-dependent protein catabolic process is impacted by a key genetic defect of clear cell kidney cancer in the VHL tumor suppressor gene, which is part of a multiprotein E3 ubiquitin ligase (Corn, 2007).

Overall, with the KIRC data, GEDFN was able to achieve better prediction, and select genes that were easily interpretable. The results pointed to several important pathways, the behavior of which may predispose patients to certain survival outcomes.

Table 3.6: Top GO biological processes for the sub-graph selected by GEDFN (KIRC data). Manual pruning of partially overlapping GO terms was conducted.

GOBPID	**Pvalue**	**Term**
GO:2001233	7.10E-10	regulation of apoptotic signaling pathway
GO:0051098	1.14E-09	regulation of binding
GO:0071363	1.27E-09	cellular response to growth factor stimulus
GO:0007178	1.48E-07	transmembrane receptor protein serine/threonine kinase signaling pathway
GO:1903827	2.27E-07	regulation of cellular protein localization
GO:0042176	5.72E-07	regulation of protein catabolic process
GO:0007507	1.66E-06	heart development
GO:0008285	1.72E-06	negative regulation of cell proliferation
GO:0048589	3.07E-06	developmental growth
GO:0007183	3.52E-06	SMAD protein complex assembly

3.5 Conclusion

We presented a new deep feedforward network classifier embedding feature graph information. It achieves sparse connected neural networks by constraining connections between the input layer and the first hidden layer according to the feature graph. Simulation experiments have shown its relatively higher classification accuracy and better feature selection ability compared to existing methods, and the real data applications demonstrated the utility of the new model in both classification and the selection of biologically relevant features.

Chapter 4

forgeNet: Forest Graph-Embedded Deep Feedforward Network

4.1 Introduction

We propose a supervised feature graph construction framework using tree-based ensemble models, as literature shows that tree-based ensemble methods such as the Random Forest (RF) (Breiman, 2001) and the Gradient Boosting Machine (GBM) (Friedman, 2002) are excellent tools for feature selection (Tang and Foong, 2014; Vens and Costa, 2011). These tree-based methods also provide relational information between features in terms compensating each other in the classification task. We develop the forest graph-embedded deep feedforward network (forgeNet) model, with a built-in tree-based ensemble classifier as a feature graph extractor on top of a modified GEDFN model. The feature extractor selects features that span a reduced feature space, and constructs a graph between the selected features based on their directional relations in the decision tree ensemble.

The application of tree-based ensemble methods as feature graph extractor is mainly based on two considerations: 1) the extractor selects effective features in a supervised manner. Thus the target outcome directly participates the feature graph construction. Compared to unsupervised feature construction such as using marginal or conditional correlation graphs, the resulting graph from trees is more informative and relevant to the specific classification task; 2) the feature extraction procedure helps reduce the dimension of the original feature space, alleviating the $n \ll p$ problem for the downstream neural network model. Similar feature representation learning scheme has been shown successful in Kong and Yu (2018), where RF is employed as a supervised feature detector. However, in that study, only the output of RF, i.e. the predicted score for each sample is used as the feature representation, and the relation between features are not considered. In contrast, our forgeNet model examines more detailed information provided by the forest and re-trains neural networks at the feature level instead of using feature representations. This way, we expect the downstream neural network part utilizing the feature information more thoroughly.

4.2 Methodology

4.2.1 The forgeNet model

We refer to Chapter 3 for the GEDFN model as our new method utilizes a similar neural network architecture. Mathematical notations remain consistent with those in Chapter 3. The newly proposed forest graph-embedded deep feedforward network (forgeNet) model consists of two components - the extractor component and the neural network component. The extractor component uses a forest model to select useful features from raw inputs with the supervision of training labels, as well as constructs a directed feature graph according to the splitting order in the individual decision trees. The neural network component feeds the generated feature graph and the raw inputs to GEDFN, and serves as the learner to predict outcomes. In forgeNet, a forest is defined as any ensemble of decision trees but not limited to random forests. In fact, any tree-based ensemble approach is applicable within the forgeNet framework. Besides RF and GBM mentioned in Section 4.1, their variants with similar outputs are also possible options, or the forest can be simply built through bagging trees (Breiman, 1996). However, since RF and GBM models are the most commonly used tree ensembles, in this paper, we only employ these two methods for a proof-of-concept purpose.

In forgeNet, a forest $\mathcal{F}$ is denoted as a collection of decision trees

$$\mathcal{F}(\Theta) = \{\mathcal{T}_m(\Theta_m)\},\ m = 1, \ldots, M,$$

where M is the total number of trees in the forest, $\Theta = \{\Theta_1, \ldots, \Theta_M\}$ represents the parameters, which include splitting variables and splitting values. In the feature graph extraction stage, $\mathcal{F}$ is fitted by training data $\mathbf{X}_{train}$ and training label $\mathbf{y}_{train}$, where $\mathbf{X}_{train} \in \mathcal{R}^{n_{train} \times p}$ and $\mathbf{y}_{train} \in \mathcal{R}^{n_{train}}$. After fitting the forest, we obtain M decision

trees, each of which contains a subset of features and their directed connections according to the tree splitting. At the same time, a binary tree can be viewed as a special case of a graph with directed edges. Hence, we can construct a set of graphs

$$\mathcal{G} = \{G_m(V_m, E_m)\},\ m = 1, \ldots, M,$$

where V_m and E_m are collections of vertices and edges in G_m respectively. Next, by merging all graphs in $\mathcal{G}$, the aggregated feature graph

$$\mathbf{G}(V, E) = \bigcup_{m=1}^{M} G_m(V_m, E_m)$$

is obtained, where $V = \bigcup_{m=1}^{M} V_m$ and $E = \bigcup_{m=1}^{M} E_m$.

In the form of its adjacency matrix, $\mathbf{G}$ is the feature graph to be embedded into the second stage of the forgeNet. Note that regardless which tree-based ensemble methods we use, it is likely that not all predictors in the original feature space can enter the forest model. A feature is included in $\mathbf{G}$ if and only if it is used at least once by the forest to split samples. As a result, the original feature space is reduced after the feature extraction. Denoting the number of vertices of $\mathbf{G}$ as $|V|$, we have $|V| < p$, and the input data matrix for the second stage is thus $\tilde{\mathbf{X}}_{train} \in \mathcal{R}^{n \times |V|}$. The columns in $\tilde{\mathbf{X}}_{train}$ corresponds to selected features in the original data $\mathbf{X}_{train} \in \mathcal{R}^{n \times p}$, and the order of columns does not matter.

The resulting feature graph $\mathbf{G}$ of feature extraction is a directed network, which differs from the one used in the original GEDFN. In Chapter 3, the adjacency matrix A in Eq. 3.1 represents an undirected feature graph. In the case of forgeNet, the adjacency matrix is naturally generalized to the directed version, and replacing A in Eq. 3.1 with an asymmetric adjacency does not affect the model construction and training. A visualization of the entire forgeNet architecture is seen in Fig. 4.1.

After fitting forgeNet with the training data, only the reduced input $\tilde{\mathbf{X}}_{test}$ and the

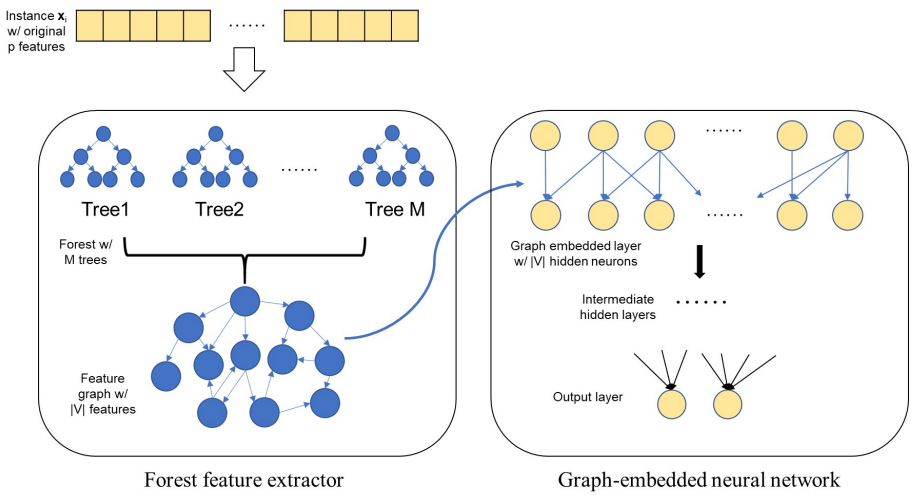


Figure 4.1: Illustration of the forgeNet model. Notations are consistent with those in the text.

testing label $\mathbf{y}_{test}$ are required for testing the prediction results, as $\tilde{\mathbf{X}}_{test}$ can be directly fed into the downstream neural nets together with the feature graph constructed from the forest.

4.2.2 Evaluation of feature importance

The selection of predictors that significantly contribute to the prediction is another major aspect of the analysis of profiling data, as they can reveal underlying biological mechanisms. Thus in forgeNet, we introduce a feature importance evaluation mechanism, which is closely related to the Graph Connection Weights (GCW) method proposed in Section 3.2.3 for the original GEDFN model. However, since the feature graph used in forgeNet has a different property from that in GEDFN where the feature graph is given, certain modifications of GCW are needed.

The main idea of GCW is that, the contribution of a specific predictor is directly reflected by the magnitude of all the weights that are directly associated with the corresponding hidden neuron in the graph-embedded layer (the first hidden layer). In forgeNet, since the connection between the input layer and the first hidden layer is no longer symmetric due to the directed feature graph structure, to evaluate the

importance of a given feature, we examine both hidden neurons in the first hidden layer and the nodes in the input layer. The importance score is thereby calculated as the summation of absolute values of the weights that are directly associated with the feature node itself and its corresponding hidden neuron in the graph-embedded layer:

$$s_j = \sum_{u=1}^{p} |w_{ju}^{(in)} \mathcal{I}(A_{ju} = 1)| + \sum_{v=1}^{p} |w_{vj}^{(in)} \mathcal{I}(A_{vj} = 1)| + \sum_{m=1}^{h_1} |w_{jm}^{(1)}|, \quad j = 1, \ldots, p,$$

where s_j is the importance score for feature j, $w^{(in)}$ denotes weights between the input and first hidden layers, and $w^{(1)}$ denotes weights between the first hidden layer and the second hidden layer. The score consists of three parts: the first two terms summarize the importance of a feature according to the directed edge connection in the feature graph $\mathbf{G}$; the third term summarizes the contribution of the feature according to the connection with the second hidden layer $\mathbf{Z}_2$. Note that the input data $\mathbf{X}$ are required to be Z-score transformed (the original value minus the mean across all samples and then divided by the standard deviation), ensuring all variables are of the same scale so that the magnitude of weights are comparable. Once the forgeNet is trained, the importance scores for all the variables can be calculated using trained weights.

4.2.3 Implementation

We employ the Scikit-learn (Pedregosa et al., 2011) package for the implementation of RF, the Xgboost package (Chen and Guestrin, 2016) for GBM, and the Tensorflow library (Abadi et al., 2016) for deep neural networks. For the choice of activation functions of neural nets, the rectified linear unit (ReLU) (Nair and Hinton, 2010) is employed. This non-linear activation has an advantage over the sigmoid function and

the hyperbolic tangent function as it avoids the vanishing gradient problem (Hochreiter et al., 2001) during model training. The entire neural net part of forgeNets is trained using the Adam optimizer (Kingma and Ba, 2014), which is the state-of-the-art version of the popular stochastic gradient descent algorithm. Also, we use the mini-batch training strategy by which the optimizer loops over randomly divided small proportions of the training samples in each iteration. Details about the Adam optimizer and the mini-batch strategy applications in deep learning can be found in (Goodfellow et al., 2016; Kingma and Ba, 2014).

The performance of a deep neural network model is associated with many hyper-parameters, including the number of hidden layers, the number of hidden neurons in each layer, the dropout proportion of training, the learning rate and the batch size. As the hyper-parameters are not of primary interest in our research, in the simulation and real data experiments, we simply tune hyper-parameters using grid search in a feasible parameter space. An example of hyper-parameter tuning can be found in Appendix B. Also, since our experiments contains a number of datasets, it is not plausible to fine tune models for each dataset. Instead, we tune hyper-parameters using some preliminary synthetic datasets, and apply the set of parameters to all experimental data. For simulation experiments, the number of trees of our forgeNets is 1000 and the number of hidden layers of the neural net is three with p (graph-embedded layer), 64 and 16 hidden neurons respectively. For real data analyses, the number of trees in the forest part is adjusted according to the size of the corresponding feature space, and the neural net structure is the same as it is in simulation.

4.3 Simulations

The goal of the simulation experiments is to mimic disease outcome classification using profiling data with $n \ll p$. Effective features are sparse and potentially correlated

through an underlying unknown structure. Several benchmark methods are experimented in addition to the new forgeNet model for comparison purpose. Through simulation, we intend to investigate whether the forgeNet model is able to outperform other classifiers without knowing the underlying structure of features.

4.3.1 Synthetic data generation

We follow a similar procedure described in Section 3.3.1. While the pipeline of data generation remains unchanged, we modified certain quantities for the purpose of improving classification difficulty. For a given number of features p, the preferential attachment algorithm (BA model) (Barabási and Albert, 1999) is employed to generate a scale-free network as the underlying true feature graph. Defining the distance between two features in the network as the shortest path between them, we calculate the $p \times p$ matrix D recording pairwise distances among features. Next, the distance matrix is transformed into a covariance matrix Σ by letting

$$\Sigma_{ij} = 0.6^{D_{ij}}, i, j = 1, \ldots, p.$$

After obtaining the covariance matrix between features, we generate n multivariate Normal samples as the data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ i.e.

$$\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \Sigma), i = 1, \ldots\ n,$$

where $n \ll p$ for imitating gene expression data. In order to add negative correlations as well, we randomly flipped the signs of 1% of the $\mathbf{X}$ columns (genes). Figure B.1 in Appendix B shows empirical pairwise feature correlation distributions for the simulated data. The plots confirm that there are significant proportions of negative correlations. To generate outcome variables, we first select a subset of features to be "true" predictors. Among vertices with relatively high degrees ("hub nodes") in

the feature graph, part of them are randomly selected as "cores", and a proportion of the neighboring vertices of cores are also selected. Denoting the number of true predictors as p_0, we uniformly sample a set of parameters $\beta = (\beta_1, \ldots, \beta_{p_0})^T$ and an intercept β_0 from a small range, say $(-0.15, 0.15)$. Finally, the outcome variable $\mathbf{y}$ is generated through a procedure similar to the generalized linear model framework

$$y_i = \mathcal{I}\{g(\beta_0 + (\mathbf{x_i}^{(true)})^T\beta) > t\}, \quad i = 1, \ldots n,$$

where $\mathbf{x_i}^{(true)} \in \mathcal{R}^{p_0}$ is the sub-vector of $\mathbf{x_i}$ and t is a threshold. For the transformation function $g(\cdot)$, we consider a weighted sum of hyperbolic tangent and quadratic function

$$g(x) = 0.7\phi(tanh(x)) + 0.3\phi(x^2).$$

The reason of using this $g(\cdot)$ function is that the transformation is non-monotone, which brings in more challenges for classification. The function $\phi(\cdot)$ is the min-max transformation scaling the input to $[0, 1]$, i.e., the original value minus the sample minimum and then divided by the difference between the sample maximum and the sample minimum.

Following the above data generation scheme, we simulate a set of synthetic datasets with $p = 5,000$ features and $n = 400$ samples. Since in profiling data, the true signals for a certain prediction task are sparse ($p_0 \ll p$), We choose $p_0 = 15, 30, 45, 60$ and 75 as the numbers of true predictors, corresponding to 1 to 5 cores selected among all hub nodes in the feature graph.

4.3.2 Evaluation of simulation experiments

We compare our method with several benchmark models. First, since the true feature graphs are known for simulation data, we are able to test the original GEDFN model with correctly specified feature graphs. At the same time, we also experiment GEDFN

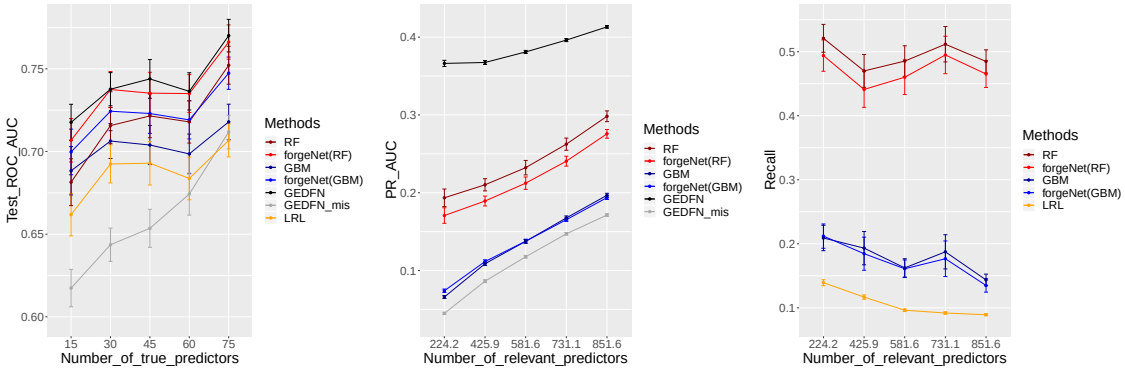


Figure 4.2: Comparison of classification and feature selection for the simulation study. (a) AUC of ROC for classification; (b) AUC of precision-recall for feature selection; (c) recall plots given fixed precision from LRL. Error bars represent the estimated mean quantities plus/minus the estimated standard errors.

with mis-specified feature graphs by randomly generating Erdo-Renyi random graphs (Erdös and Rényi, 1959), which have a different graph topology structure from the true scale-free networks. Also, since forgeNet inherently fits a tree-based ensemble classifier, it is natural to compare the performance of a forgeNet with its forest part alone. We choose two representative tree methods RF and GBM for the experiments, and correspondingly test two versions of forgeNets - forgeNet-RF and forgeNet-GBM. Finally, the logistic regression classifier with lasso (LRL) (Tibshirani, 1996) is also added as a representative of linear machines.

For each of the data generation settings, fifty independent datasets are generated. For each dataset, we randomly split samples into training and testing sets at a ratio of 4:1. All models are fitted using the training dataset and then used to predict the testing dataset. To evaluate classification results, areas under Receiver Operating Characteristic curves (ROC-AUC) are calculated using the predicted class probabilities and the labels of the testing set. The final testing result for a simulation case is then given by the average testing ROC-AUC across the 50 datasets.

As for feature selection, all the methods except LRL provide relative feature importance scores; LRL does not rank features but directly gave the selected feature

subset. Knowing the true predictors for simulated data, we could use the binary true predictor labels to evaluate the accuracy of feature selection. However, in preliminary numerical experiments, it is observed that though we fix the number of true features in each case, neighboring features of true predictors in the feature graph are also informative for classification even if they are not in the true feature set. This is because these neighboring features have a relatively high correlation with selected true predictors (0.6 according to Section 4.3.1). Therefore, when evaluating the results of feature selection, it is more appropriate to investigate a set of "relevant" features including those neighboring features, rather than the "true" feature set only. The average numbers of relevant features are 208.8, 460.4, 615.4, 717.8, and 864.7 respectively, corresponding to the five cases of true features $p_0 = 15, 30, 45, 60$ and 75.

Since the relevant feature sets are still small compared to the entire feature space ($p = 5000$), the AUC of the precision-recall curve is a more appropriate metric here. We thus compare feature selection results using binary labels of relevant features for all methods providing feature scores. As for LRL, for each dataset, we compare recall values of our methods and LRL given the precision value of LRL. That is, the precision of LRL helps locate points on the precision-recall curves of forgeNets, and corresponding recall values are used for comparison.

4.3.3 Simulation results

Fig. 4.2(a) shows the results of classification accuracy comparison. With the increasing number of true predictors, all of the methods performed better as there were more signals in the entire feature space. From the figure, the two versions of forgeNets, forgeNet-RF and forgeNet-GBM, significantly improved the classification performance of their forest counterparts, i.e., RF and GBM. Also, the forgeNet-RF was the only method that achieved similar classification accuracy as GEDFN which

benefited from the use of true feature graphs. When GEDFN was given mis-specified feature graphs (GEDFN_mis), its classification ability was weakened with AUC values even worse than LRL. In summary, in terms of prediction, forgeNets beat all classic machine learning methods compared here (RF, GBM, LRL), achieved very similar accuracy compared to GEDFN using true feature graphs, and significantly outperformed GEDFN once its feature graphs were mis-specified.

Feature selection results can be seen in Fig. 4.2(b) and (c). Comparing the precision-recall AUCs from Fig. 4.2(b), it can be observed that GEDFN using true feature graph was the best method for feature importance ranking, yet again the outstanding performance was ruined by mis-specified feature graphs. The results of forgeNets were significantly better than GEDFN_mis, and were consistent with their forest counterparts. As the training of neural networks in forgeNets largely relied on feature graphs given by forests, it is not surprising to see that forgeNets could achieve similar feature selection results as their forest counterparts. In Fig. 4.2(c), both forgeNet-RF and forgeNet-GBM were able to achieve higher recall values than LRL. In summary, in terms of feature selection, forgeNets outperformed the traditional lasso method and had consistent performance with their forest counterparts. Although not as good as GEDFN with true feature graphs, forgeNets produced significantly better feature selection than GEDFN using mis-specified feature graphs. Finally, we observe that the choice of the forest in forgeNets mattered, and among the two versions in our experiments, forgeNet-RF was a more powerful model.

The simulation study proved the forgeNet a powerful classifier, with reasonably good feature selection ability. Through the experiment results, one can easily conclude the novelty of forgeNets is that, by borrowing the neural net architecture of the original GEDFN, forgeNets utilize feature information more effectively in classification tasks compared to regular tree-based ensemble methods.

ForgeNets involve stochastic model fitting. There can be concerns about for-

geNet's stability and scalability. The former refers to the sensitivity regarding different initial values in training the deep neural network. To test the reproducibility of the forgeNet model, we examined the classification accuracy of 10 repeated forgeNet runs for fixed synthetic datasets. The results for both forgeNet-RF and forgeNet-GBM are shown in Table B.1. Despite a little variability in cases where the numbers of true features are small, forgeNets exhibited robustness with respect to initial values in general. The second aspect is forgeNet's capability of tackling large-scale datasets (i.e., larger samples and/or extremely large feature spaces) without inducing impractical cost in time and memory, compared to traditional classification methods. To answer this question, we designed additional experiments for forgeNet-RF and forgeNet-GBM to analyze their computational cost and compared the cost with their tree-ensemble counterparts respectively. The analysis is reported in Tables B.3, B.4, B.5, B.6, where we concluded that the extra computation time and memory usage induced by forgeNets stayed in a limited scale, indicating the usability of the method for large-scale data.

4.3.4 Analysis of estimated feature graphs

ForgeNets use feature graphs constructed by tree ensemble methods. It is of interest to investigate the feature graphs constructed by the tree-based feature extractors. The comparison between the estimated feature graphs and the true simulated feature graphs were based on two aspects, vertices and edges. For each synthetic dataset, we selected the sub-network, denoted as **H**, containing all relevant features defined in Section 4.3.2 and their neighbors (i.e., second neighbors of true features) in the true feature graph. To compare vertices, we calculated the proportion of features in the estimated feature graph that fell in **H**. Table 4.1 (row "RF (vertex)" and row "GBM (vertex)") shows the averaged vertex proportions for different simulation cases. As for edges, it is noted that the feature graph construction by tree-based methods is not for

Table 4.1: Analysis of feature graphs constructed by RF and GBM. Proportions are averaged across the 50 datasets in each simulation case.

#true features	15	30	45	60	75
RF (vertex)	0.429	0.585	0.663	0.723	0.768
RF (edge)	0.284	0.447	0.546	0.625	0.692
GBM (vertex)	0.437	0.582	0.660	0.718	0.764
GBM (edge)	0.226	0.376	0.467	0.548	0.609

recovering the original correlation feature graph. Instead, two adjacent features in a tree are more likely to be complementary to each other regarding a given classification task. Consequently, the estimated feature graphs were expected to be more similar to the complement graph of $\mathbf{H}$, denoted as $\mathbf{H}^c$, rather than $\mathbf{H}$ itself. In graph theory, the complement graph $\mathbf{H}^c$ of $\mathbf{H}$ is a graph with the same vertices such that two vertices of $\mathbf{H}^c$ are connected if and only if they are not connected in $\mathbf{H}$ (Bondy et al., 1976). The averaged proportions of the estimated feature graph edges that fell in the edge set of $\mathbf{H}^c$ can be also found in Table 4.1 (row "RF (edge)" and row "GBM (edge)").

The analysis of estimated feature graphs indicates that forgeNet selects relevant features but views the feature interactions from a different perspective. On one hand, forgeNet's tree-based feature extractor identifies relevant features for classification that are consistent with those in the original correlation feature graph. on the other hand, the feature extractor constructs feature graphs based on a complementary relationship among features instead of direct correlation. This again aligns with the concept of the supervised feature extractor, as the estimated feature graph is not necessarily recovering the correlation graph, as long as it contains useful information of feature interactions in predicting a certain classification outcome.

Table 4.2: Classification results for BRCA data

Methods	forgeNet-RF	RF	forgeNet-GBM	GBM	LRL
Avg. ROC-AUC	0.742	0.672	0.716	0.691	0.689
s.d.	0.066	0.048	0.100	0.022	0.084

4.4 Data Analysis

4.4.1 Breast invasive carcinoma RNAseq data

We applied forgeNets to the Cancer Genome Atlas (TCGA) breast cancer (BRCA) RNA-seq dataset (Koboldt et al., 2012). The dataset consists of a gene expression matrix with 20155 genes and 1097 cancer patients, as well as the clinical data including survival information. The classification task is to predict the three-year survival outcome. We excluded patients with missing or censored survival time for which the three-year survival outcome could not be decided. Also, genes with more than 10% of zero values were screened out. As a result, the final dataset contains a total of $p = 16027$ genes and $n = 506$ patients, with 86% positive cases. For each gene, its expression value was Z-score transformed.

Using the BRCA data, we again tested two versions of forgeNets together with RF, GBM, and LRL. The classification was conducted using a 5-fold stratified cross validation process, and the final prediction AUC for each method is computed by averaging the five validation results.

Table 4.2 summarizes the classification results. From the table, forgeNets again outperformed their forest counterpart models and LRL. Therefore, this real data application also led to a similar conclusion as in Section 4.3 that forgeNets brought in significant improvement for classification.

Feature selection was also conducted for BRCA data. We obtained ranked gene importance lists by averaging importance scores across the five cross validation results from all methods except LRL. For LRL, the intersection (456 genes) of the five selected feature sets is used as the final selected features. We chose top 500 ranked genes for

Table 4.3: Top 3 GO biological processes for each method, after manual removal of redundant GO terms.

ID	Term	P-value	Count	Size
forgeNet-RF				
GO:0031647	Regulation of protein stability	0.00123	17	229
GO:0090502	RNA phosphodiester bond hydrolysis, endonucleolytic	0.00369	7	62
GO:1901998	Toxin transport	0.00499	5	35
RF				
GO:2000679	Positive regulation of transcription regulatory region DNA binding	0.00255	4	19
GO:0010172	Embryonic body morphogenesis	0.00313	3	10
GO:0090042	Tubulin deacetylation	0.0042	3	11
forgeNet-GBM				
GO:0001676	Long-chain fatty acid metabolic process	0.00138	9	84
GO:0032890	Regulation of organic acid transport	0.00155	6	40
GO:0046470	Phosphatidylcholine metabolic process	0.00449	7	65
GBM				
GO:0006633	Fatty acid biosynthetic process	0.000454	12	121
GO:0030520	Intracellular estrogen receptor signaling pathway	0.000643	7	47
GO:0010763	Positive regulation of fibroblast migration	0.00322	3	10
LRL				
GO:0051047	Positive regulation of secretion	0.000609	20	317
GO:0006090	Pyruvate metabolic process	0.000911	9	90
GO:0019359	Nicotinamide nucleotide biosynthetic process	0.00204	8	82

each ranked list so that the numbers are of a similar magnitude as the genes selected by LRL. Functional analysis of all final gene lists was conducted by the Gene Ontology (GO) enrichment test using GOstats package (Falcon and Gentleman, 2007). We limited the analysis to GO biological processes containing 10-500 genes, and a p-value cutoff of 0.005. After manual removal of highly overlapping GO terms, the top 3 GO terms that contained the most number of selected genes are found in Table 4.3.

The top GO term selected by forgeNet-RF was regulation of protein stability. It has been found that estrogen receptor (ER) alpha has increased abundance and activity in breast cancer. One of the mechanisms facilitating this change is the protection of ER from degradation by the ubiquitin-proteasome system (Tecalco-Cruz and Ramirez-Jarquin, 2017). Another critical protein, HER2 (human epidermal growth factor receptor 2), has also been found to have increased stability and activity in some breast cancer tissues through the formation of Her2-Heat-shock protein 27 (HSP27) complex (Kang et al., 2008). The protein stability mechanism has not been previously linked to the survival outcome of breast cancer. The second GO term found by

forgeNet-RF, RNA phosphodiester bond hydrolysis, endonucleolytic, is part of rNRA and tRNA processing. It plays a critical role in the protein synthesis of the cancer cells. The third term, toxin transport, is specific to breast cancer. It is suggested that increased toxin presence in the mammary tissue is a pre-disposing factor to breast cancer (McManaman and Neville, 2003; Quezada and Vafai, 2014).

The forgeNet-GBM and GBM results both point to fatty acid metabolism, which is known to be dysregulated in breast cancer (Monaco, 2017). The GBM selected the estrogen receptor signaling pathway, which is critically important in breast cancer development. The LRL selected GO terms include positive regulation of secretion, which includes lactation, in addtion to a number of metabolic processes.

In this real data analysis, we were also interested in examining the feature graphs constructed by the two tree-based ensemble methods. We compared the estimated feature graphs with the real gene network employed in Section 3.4 from the HINT database (Das and Yu, 2012). Among the 16027 genes, 7816 of them were involved in the HINT network, and there was no connectivity (edge) information for the remaining 8211 genes. The estimated feature graphs by forgeNet-RF had an average[1] of 8997.8 vertices, and 44.2% of them overlapped with the HINT gene network. The estimated feature graphs by forgeNet-GBM had an average of 428 vertices, and 52.6% of them fell in the HINT network. The difference of vertex numbers of the two tree-based methods were caused by their own tree construction mechanisms, and the percentages were roughly proportional to the genes covered by the HINT network.

Unlike the analysis in Section 4.3.4, comparison between the estimated feature graphs and the HINT network regarding edges was not feasible, as the underlying true predictive feature sub-graph structure was unknown. We observed few overlapping edges between the estimated feature graphs and the HINT network, for both forgeNet-RF and forgeNet-GBM. This is expected. As seen in the simulation study,

[1]The average was taken over the graphs constructed from different folds of samples.

the estimated edges by RF and GBM tend to be the complimentary edges in the sub-network involving true predictors. In addition, the true biological network is much more complex than a simple correlation network.

It can be noted that, in the case of real data applications, both GEDFN and forgeNet can be regarded as a way of feature pre-screening. GEDFN utilizes external knowledge (e.g., the HINT network data), which cannot utilize features not presenting in the known feature graph. In contrast, forgeNet examines initial input with a larger feature space and screens features in a supervised manner, following the philosophy that the forest feature extractor should be able to decide the usefulness of a feature. Depending on the real dataset and the classification outcome of interest, the two ways of pre-screening can agree or differ with each other, and there is no way to guarantee which mechanism dominates the other.

4.4.2 Breast invasive carcinoma microRNA data

We further applied forgeNets to the BRCA microRNA dataset (Koboldt et al., 2012). There was no readily available feature graph for the microRNA data. The dataset consists of 2588 microRNAs and 848 BRCA patients. Again, we examined the classification task for predicting the three-year survival outcome. Similar to Section 4.4.1, we excluded patients with missing or censored survival time for which the three-year survival outcome could not be determined. MicroRNAs with more than 50% of missing values were also screened out. As a result, the final dataset contained a total of $p = 310$ microRNAs and $n = 424$ patients, with 85% positive cases. Although this was not strictly an "$n \ll p$" dataset, the number of features were on the same scale as the sample size. Therefore, it was still a problem that challenges traditional classification methods. We applied the K-Nearest Neighbor (KNN) imputation (Troyanskaya et al., 2001) for the remaining missing values, and each microRNA was Z-score transformed.

Following the same 5-fold stratified cross validation procedure as in Section 4.4.1,

Table 4.4: Classification results for BRCA microRNA data

Methods	forgeNet-RF	RF	forgeNet-GBM	GBM	LRL
Avg. ROC-AUC	0.637	0.528	0.617	0.560	0.571
s.d.	0.066	0.123	0.052	0.042	0.061

we obtained the classification results of the microRNA data, shown in Table 4.4. The microRNA data were more challenging than the gene expression data, as the ROC-AUCs for all methods were lower. Nevertheless, the forgeNets were again able to outperform their tree-based counterparts, as well as the logistic regression with lasso.

We analyzed the functions of the selected microRNAs using DIANA mirPath V.3 using a microT score threshold of 0.95 (Vlachos et al., 2012). The top 5 KEGG pathways for each method are shown in Table 4.5. As the logistic regression with lasso selected 29 microRNAs, we used the top 30 microRNAs for each of the other methods. All five methods selected "Hippo signaling pathway" among the top pathways. The dysregulation of the pathway is associated with the metastasis and resistance to chemotherapy in breast cancer (Wei et al., 2018; Wu et al., 2020).

Among the top 5 pathways selected by forgeNet-RF, three were signalling pathways, which was the most among all methods. The Rap1 singalling pathway is well-known for regulating breast cancer cell migration through modulating matrix metalloproteinases (MMPs) (McSherry et al., 2011). AMP-activated protein kinase (AMPK) signaling responds to a number of endocrine signals, and regulates energy, growth and motility of cells (Zhao et al., 2017). Its role in breast cancer progression and therapy has been well documented (Zou et al., 2017; Cao et al., 2019). The AMPK pathway was selected by both forgeNet-RF and forgeNet-GBM as the top 5.

Besides signaling pathways, forgeNet-RF also selected the glycosaminoglycan - keratan sulfate pathway and the glycosphingolipid pathway. Keratan sulfate (KS) is the newest glycosaminoglycan, and its roles in cancer hasn't been clearly elucidated (Caterson and Melrose, 2018). Recently it's been found that increased KS epitope is associated with worse survival in pancrease cancer (Leiphrakpam et al.,

Table 4.5: Top 5 pathways selected by each method using mirPath V.3.

Method	**# significant pathways (p¡0.01)**	**Top 5 pathways**	**P-value**
forgeNet-RF	12	Hippo signaling pathway	6.63E-06
		Glycosaminoglycan biosynthesis - keratan sulfate	0.000752
		Rap1 signaling pathway	0.000882
		AMPK signaling pathway	0.000928
		Glycosphingolipid biosynthesis - lacto and neolacto series	0.00119
RF	3	Prion diseases	2.67E-20
		Hippo signaling pathway	4.69E-10
		Thyroid hormone synthesis	0.00651
		Adrenergic signaling in cardiomyocytes	0.0165
		Long-term potentiation	0.0165
forgeNet-GBM	6	Pathways in cancer	0.000248
		Transcriptional misregulation in cancer	0.000248
		Hippo signaling pathway	0.000417
		AMPK signaling pathway	0.000950
		Maturity onset diabetes of the young	0.00127
GBM	15	Prion diseases	3.75E-20
		Hippo signaling pathway	3.87E-16
		Signaling pathways regulating pluripotency of stem cells	5.31E-06
		Proteoglycans in cancer	0.000532
		Colorectal cancer	0.000794
LRL	8	GABAergic synapse	2.19E-05
		ECM-receptor interaction	2.19E-05
		Hippo signaling pathway	2.19E-05
		Morphine addiction	8.90E-05
		Proteoglycans in cancer	0.000251

Table 4.6: Classification results for healthy human metabolomics data

Methods	forgeNet-RF	RF	forgeNet-GBM	GBM	LRL
Avg. ROC-AUC	0.686	0.649	0.682	0.666	0.649
s.d.	0.066	0.042	0.044	0.039	0.077

2019). Glycolipids are essential in maintaining plasma membrane stability. Aberrant glycosphingolipid metabolism play critical roles in cancer progression and metastasis (Zhuo et al., 2018).

Comparatively, among the top five pathways selected by LRL, two were neurological pathways that bear no clear relation to breast cancer - GABAergic synapse, and morphine addiction. The extracellular matrix (ECM)-receptor interaction pathway is important in cancer progression (Walker et al., 2018), and proteoglycans are important for cell surface adhesion and cancer invasion (Nikitovic et al., 2018). Overall, forgeNet-RF achieved better performance in classification, as well as selected more interpretable features.

4.4.3 Healthy human metabolomics dataset

Another real dataset we experimented was the untargeted metabolomics dataset measured by high-resolution liquid chromatography - mass spectrometry (LC/MS) from the Emory/Georgia Tech Center for Health Discovery and Well Being (CHDWB). The cohort was made up of healthy adults. The data was processed using apLCMS with hybrid mode (Yu et al., 2009, 2013). We limited the analysis to the baseline measurements of the subjects with available clinical data. The metabolic feature matrix contained 8807 features and 382 subjects, as well as clinical and demographic information. The classification task was to predict obesity as indicated by the BMI index. Metabolites with more than 10% of zero values were screened out. Other general confounders including age, gender (male/female) and ethnicity (3 races) were included as predictors. As a result, the final dataset contained a total of $p = 4997$

Table 4.7: Top 5 pathways selected by each method using Mummichog.

Method	**# significant pathways (p¡0.05)**	**Top 5 pathways**	**P-value**
forgeNet-RF	17	Tryptophan metabolism	0.00126
		Histidine metabolism	0.00681
		Lipoate metabolism	0.00832
		Glycosphingolipid metabolism	0.00865
		Glutathione Metabolism	0.00924
RF	26	Alanine and Aspartate Metabolism	0.00008
		Urea cycle/amino group metabolism	0.00134
		Nitrogen metabolism	0.00185
		Aspartate and asparagine metabolism	0.00294
		Tryptophan metabolism	0.00378
forgeNet-GBM	5	Histidine metabolism	0.00059
		Vitamin B12 (cyanocobalamin) metabolism	0.00681
		Squalene and cholesterol biosynthesis	0.01949
		Androgen and estrogen biosynthesis and metabolism	0.0268
		Ubiquinone Biosynthesis	0.03655
GBM	14	Glycosphingolipid metabolism	0.00823
		Blood Group Biosynthesis	0.01361
		Glycosylphosphatidylinositol(GPI)-anchor biosynthesis	0.01361
		Glycosphingolipid biosynthesis - lactoseries	0.01361
		Glycosphingolipid biosynthesis - neolactoseries	0.01361
LRL	14	Glycerophospholipid metabolism	0.00151
		Lysine metabolism	0.00176
		Prostaglandin formation from dihomo gama-linoleic acid	0.00193
		Arachidonic acid metabolism	0.00378
		Saturated fatty acids beta-oxidation	0.01294

predictors, including 4993 metabolic features and 4 confounding variables. The obesity outcome was defined as BMI > 30, and 25.6% of the subjects were positive cases. For each continuous predictor, its value was Z-score transformed.

The 5-fold stratified cross validation classification results of the metabolomics data are shown in Table 4.6. Although the data were again challenging and no method performed very well, the forgeNets were better classifiers compared to other benchmarks. Using the top 10% of the metabolic features selected by each method, we conducted pathway analysis using Mummichog (Li et al., 2013). As shown in Table 4.7, RF selected the largest number of significant metabolic pathways, followed by forgeNet-RF. This is consistent with the simulation results. The top pathways selected by RF were all focused on amino acids metabolism. The top pathways selected by forgeNet-RF included amino acids metabolism, membrane lipid metabolism, and reduction-oxidation pathways, most of which were also in the list of the RF results. LRL selected a slightly smaller number of pathways than forgeNet-RF. Its top pathways were diverse with some pathways with no apparent relation to BMI, such as the prostaglandin and arachidonic metabolism pathways. The pathways selected by GBM were more focused on glycolipid metabolism, and those selected by forgeNet-GBM were diverse, some of which don't have a clear link to BMI. Overall, RF and forgeNet-RF showed the most interpretable pathway analysis results. Combined with its better predictive power, forgeNet-RF was again the preferred method among all those being compared.

4.5 Conclusion

We presented forgeNet that uses tree-based ensemble methods to extract feature connectivity information, and uses GEDFN for graph-based predictive model building. The new method was able to achieve sparse connection for neural nets without seek-

ing external information, i.e., known feature graphs. It works well in the "$n \ll p$" situation. Simulation experiments showed forgeNets' relatively higher classification accuracy compared to existing methods; the TCGA BRCA RNA-seq dataset, the TCGA BRCA microRNA dataset and a metabolomics dataset demonstrated the utility of forgeNets in both classification and the selection of biologically interpretable predictors.

Appendix A

Appendix for Chapter 3

A.1 Hyper-parameter tuning of GEDFN

A.1.1 Overview

As illustrated in the main article, the GEDFN model is associated with several types of hyper-parameters. Although we generated and used a large amount of datasets in our simulation study, the number of samples and number of features were fixed as 400 and 5000 respectively. Hence, instead of tuning GEDFN dataset by dataset which was infeasible, we could simply tune a "uniformly applicable" GEDFN model for our experiments. This was also because the different simulation settings (#true predictors, singleton proportions, inverse link functions etc.) should not be regarded as known when training the classifier. We generated additional synthetic datasets apart from the ones we used in simulation experiments, with training and validating samples. The hyper-parameter tuning process was then guided by the validation AUC of ROC, namely we would choose the best candidate hyper-parameter with the best validation AUC score.

A.1.2 Architecture

The skeleton of a DNN model is its architecture, so our first step was to tune the number of hidden layers and the number of hidden neurons in each layer. Layers could not be too many, since we were dealing with such small samples. Meanwhile, a too shallow neural network would not fit well for our complex classification problem. Keeping this trade-off in mind, we decided to try from 2 to 5 hidden layers. For the numbers of hidden neurons, we followed the convention in the deep learning field that set the numbers to be powers of two, with decreasing magnitude from the input layer to the output layer. With other hyper-parameters temporarily chosen by convention, this step led us to build the skeleton of our GEDFN as three hidden layers with neurons 5000 (graph-embedded layer), 64, and 16.

A.1.3 Regularization

In GEDFN, we employed the dropout technique to avoid over-fitting. Dropout were only applied to the second and third hidden layers but not the first, otherwise the connection of the first hidden layer based on the feature graph would be destroyed. We tuned the dropout proportion according to a 1-d grid with candidates 0.5, 0.6, 0.7, 0.8 and 0.9, with 0.9 selected as the final value.

A.1.4 Training

Hyper-parameters associated with the optimization process would not affect the classification performance in general. However, they could result in different convergence rate and thus were also tuned. A relatively large learning rate would accelerate the convergence, but it also bore the risk of "skipping" the optimum. We tuned the learning rate and the batch size using a 2-D grid with candidate sets 0.05, 0.01, 0.005, 0.001, 0.005, 0.0001 and 1, 8, 16, 32, 64 along each axis respectively. Finally,

the combination with learning rate 0.0001 and batch size 8 turned out to be the best choice.

A.1.5 Note

For the two real datasets (BRCA and KIRC) we experimented, the feature spaces were larger than the synthetic datasets. Nevertheless, the sample sizes were still limited which obstructed the use of larger GEDFN models. Therefore, we stuck to the GEDFN model tuned in simulation experiments. Larger models with more hidden layers was tried, but only to obtain the same or even worse results.

A.2 Correlation distributions of synthetic data

To ensure that features in our synthetic datasets were both positively and negatively correlated, we randomly selected nine generated datasets for investigating. For each dataset, we compute the empirical correlation matrix and plot the pairwise feature correlation histogram, shown in Figure A.1.

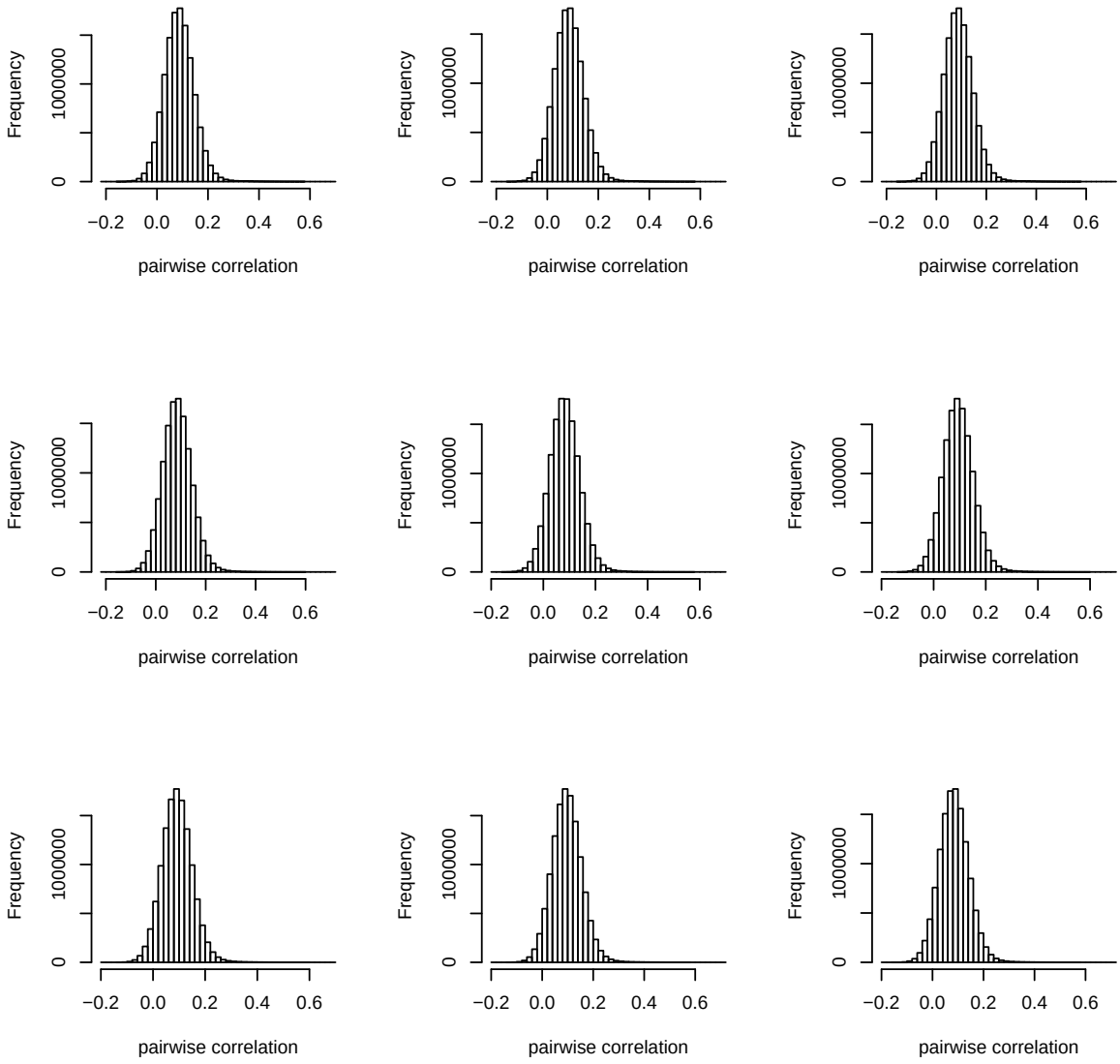


Figure A.1: Histograms of the pairwise feature correlation distributions for randomly selected simulation datasets.

A.3 Mis-specification of feature graphs

The comparison of classification and feature selection between GEDFN using informative graphs and GEDFN with misspecified graphs is shown in Figure A.2. Datasets are of 50% singletons.

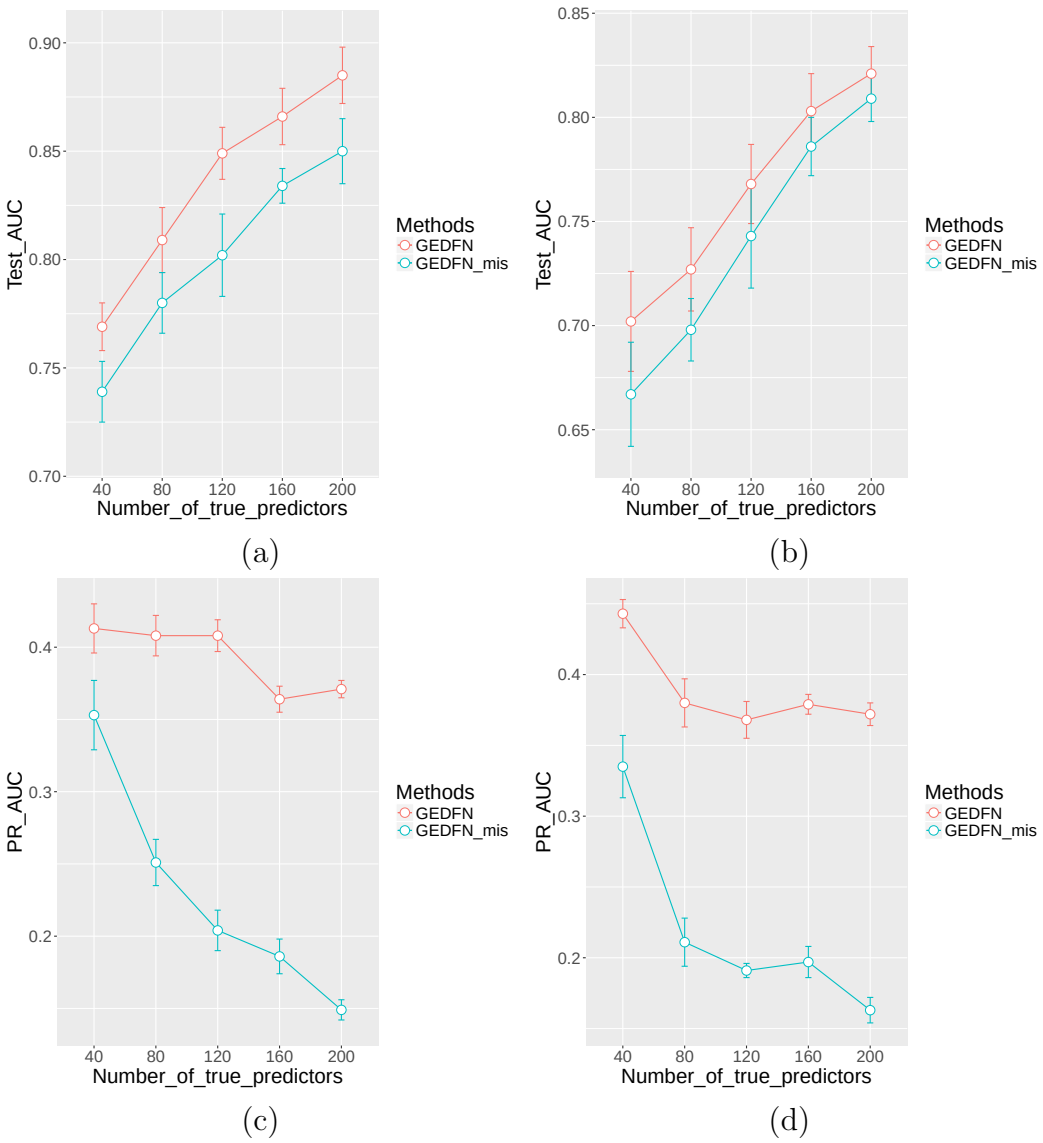


Figure A.2: Comparison of classification and feature selection between GEDFN using informative graphs and GEDFN with misspecified graphs. Left column: sigmoid inverse link; right column: tanh plus quadratic inverse link. First row: AUC of ROC for classification; second row: AUC of precision-recall for feature selection. Error bars represent the estimated mean quantity plus/minus the standard error.

Appendix B

Appendix for Chapter 4

B.1 Correlation distributions of synthetic data

We intended to simulate the synthetic datasets with both positive and negative feature correlation. To confirm, we randomly selected nine simulated datasets for investigating. For each dataset, we computed the empirical correlation matrix and plotted the pairwise feature correlation histogram, shown in Figure A.1.

B.2 Sensitivity analysis

B.2.1 Initial values

The initial values of the neural network part in forgeNet are generated by the deep learning library (i.e. Tensorflow). To test the reproducibility of forgeNet, we used one dataset for each simulation case and ran 10 repeated experiments. The testing results of forgeNet-RF and forgeNet-GBM are shown in Table B.1.

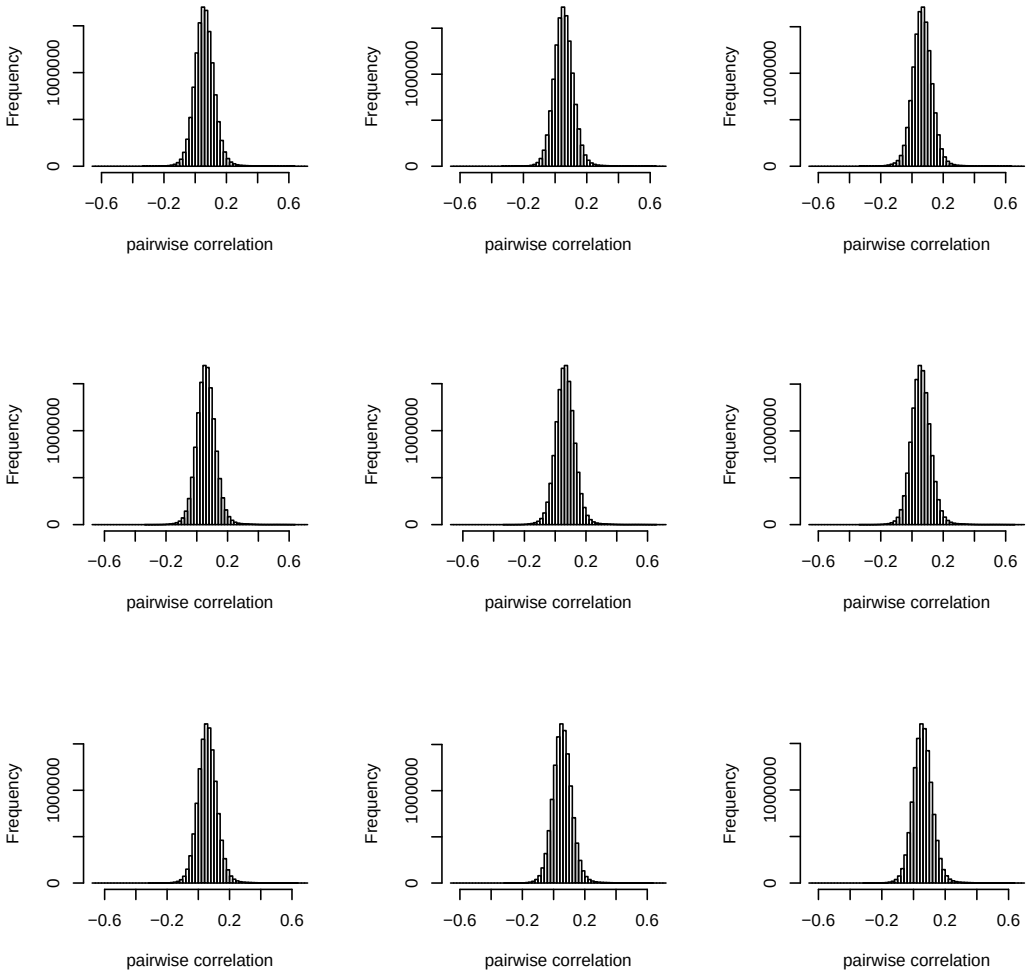


Figure B.1: Histograms of the pairwise feature correlation distributions for randomly selected simulation datasets.

forgeNet-RF	#true predictors					forgeNet-GBM	#true predictors				
	15	30	45	60	75		15	30	45	60	75
1	0.638	0.822	0.709	0.786	0.772	1	0.602	0.798	0.687	0.791	0.771
2	0.609	0.817	0.663	0.798	0.792	2	0.606	0.8	0.733	0.775	0.773
3	0.556	0.809	0.721	0.79	0.778	3	0.567	0.809	0.713	0.79	0.723
4	0.65	0.817	0.701	0.776	0.791	4	0.552	0.789	0.652	0.77	0.737
5	0.624	0.839	0.673	0.778	0.774	5	0.587	0.786	0.651	0.773	0.755
6	0.596	0.834	0.725	0.792	0.763	6	0.631	0.791	0.667	0.784	0.741
7	0.627	0.816	0.7	0.788	0.791	7	0.645	0.791	0.648	0.79	0.73
8	0.567	0.812	0.7	0.784	0.787	8	0.618	0.803	0.662	0.759	0.777
9	0.637	0.838	0.722	0.79	0.783	9	0.657	0.827	0.671	0.768	0.748
10	0.645	0.824	0.732	0.8	0.755	10	0.635	0.805	0.701	0.778	0.743
Avg.	0.615	0.823	0.705	0.788	0.778	Avg.	0.61	0.8	0.679	0.778	0.75
S.d.	0.033	0.011	0.023	0.008	0.013	S.d.	0.034	0.012	0.029	0.011	0.019

Table B.1: Testing ROC-AUC of repeated experiments for fixed datasets. Left: forgeNet-RF. Right: forgeNet-GBM.

B.2.2 Hyper-parameters

In this subsection, we demonstrate an example of how we choose the hyper-parameters using grid search. We divided the hyper-parameters into two sets: network architecture-related parameters, and training-related parameters. The former include the number of hidden layers and the number of hidden neurons in each layer, and the latter include the dropout proportion, the learning rate and the batch size.

In forgeNet, the first hidden layer (graph-embedded layer) always has the same amount of hidden neurons as the number of features p, hence we only need to tune subsequent hidden layers. We set a feasible space with 2, 3 and 4 hidden layers (including the first hidden layer), as the neural network could not be too deep in our small sample scenarios. For the number of hidden neurons, we used the conventional choices such as 128, 64, 32 and 16. Again, to avoid overfitting, hidden neurons over 128 were not considered. For training related parameters, we designed several representative options and tested their combinations. A hyper-parameter tuning example is shown in Table B.2. In this particular example, the best neural network structure was with three hidden layers, and the two fully connected layers had 64 and 16 neurons respectively. Also, the best dropout proportion/learning rate/batch size

combination was 0.5/0.0001/32.

Dropout/Learning rate/Batch size	#hidden layers & #hidden neurons					
	p+64	p+128	p+64+16	p+128+32	p+128+32+16	p+128+64+16
0.2/0.0001/8	0.79	0.802	0.8	0.767	0.817	0.795
0.2/0.0001/16	0.821	0.772	0.798	0.762	0.814	0.81
0.2/0.0001/32	0.804	0.806	0.781	0.781	0.815	0.795
0.2/0.001/8	0.735	0.766	0.719	0.746	0.739	0.747
0.2/0.001/16	0.772	0.718	0.7	0.756	0.728	0.738
0.2/0.001/32	0.739	0.764	0.737	0.721	0.768	0.768
0.2/0.01/8	0.734	0.735	0.756	0.729	0.722	0.725
0.2/0.01/16	0.758	0.679	0.702	0.724	0.712	0.71
0.2/0.01/32	0.782	0.696	0.711	0.747	0.729	0.717
0.5/0.0001/8	0.799	0.799	0.807	0.794	0.804	0.787
0.5/0.0001/16	0.797	0.811	0.809	0.787	0.805	0.779
0.5/0.0001/32	0.803	0.824	**0.837**	0.799	0.801	0.797
0.5/0.001/8	0.739	0.739	0.76	0.735	0.734	0.736
0.5/0.001/16	0.761	0.772	0.777	0.761	0.754	0.737
0.5/0.001/32	0.775	0.779	0.781	0.787	0.799	0.78
0.5/0.01/8	0.708	0.697	0.692	0.716	0.675	0.749
0.5/0.01/16	0.657	0.746	0.735	0.747	0.725	0.724
0.5/0.01/32	0.735	0.735	0.718	0.72	0.774	0.706
0.8/0.0001/8	0.804	0.794	0.789	0.799	0.816	0.783
0.8/0.0001/16	0.804	0.801	0.826	0.812	0.81	0.774
0.8/0.0001/32	0.787	0.782	0.794	0.797	0.816	0.816
0.8/0.001/8	0.762	0.739	0.818	0.812	0.783	0.807
0.8/0.001/16	0.781	0.764	0.802	0.783	0.806	0.793
0.8/0.001/32	0.808	0.777	0.819	0.799	0.786	0.79
0.8/0.01/8	0.727	0.699	0.781	0.782	0.51	0.52
0.8/0.01/16	0.731	0.711	0.781	0.764	0.79	0.525
0.8/0.01/32	0.736	0.733	0.783	0.743	0.797	0.65

Table B.2: An example of hyper-parameter tuning for forgeNet (RF). The column names denote **hidden layers** and their corresponding numbers of hidden neurons. For example, "p+64+16" stands for a neural network architecture with a p-dimensional input layer, a p-dimensional graph-embedded layer, a 64-dimensional fully connected hidden layer, a 16-dimensional fully connected hidden layer and a two-dimensional output layer.

B.3 Computational cost

In this section, we examine the scalability of the forgeNet model by using synthetic datasets with large sample sizes and extremely large feature spaces. We set candidate sample sizes as 400, 800, 2000 and 5000 and candidate number of features 5000,

10000, 20000 and 50000. The numbers should cover the upper size limit of a typical trancriptomic dataset well. We compared both the time cost and the memory cost of forgeNets with their tree-ensemble counterparts. The experiments were run on a workstation with dual Xeon Gold 6136 processors, 256 GB RAM, and a single Nvidia Quadro P6000 GPU.

The results for forgeNet-RF are shown in Table B.3 and B.4, and the results for forgeNet-GBM are shown in Table B.5 and B.4. Detailed reading instruction can be found in corresponding table captions. From the tables, it was not surprising to see that forgeNets took more time and used more space than their tree feature extractors alone, since forgeNets bore additional deep neural network computation. However, the extra time and memory (GPU) induced by forgeNets were well acceptable, indicating the plausibility of the method for large-scale data.

Time (sec)	#features			
#samples	5000	10000	20000	50000
400	4.5/14.3	4.9/15.5	5.3/18.7	5.6/24.0
800	5.3/19.7	5.4/24.0	5.7/26.3	6.3/31.9
2000	5.9/34.4	6.4/39.7	7.1/45.6	9.3/56.2
5000	7.5/68.8	9/81.4	11.6/88.3	15.6/108.6

Table B.3: Computational time for forgeNet-RF and the corresponding RF model alone. The time used by RF and by forgeNet-RF are separated by "/". For example, "4.5/14.3" means the time of running RF is 4.5 seconds while running the entire forgeNet takes 14.3 seconds.

B.4 Simulation experiments for datasets with no signal

It is also of interest to conduct additional simulation experiments using "null" datasets, which contain no real signals for the given classification outcome. This is a way of examining false positive effect of classifiers. The analysis was conducted using feature

Memory (MB)	#features			
#samples	5000	10000	20000	50000
400	141.2 (73.1)	171.8 (98.6)	233.2 (145.6)	417.2 (234.6)
800	171.7 (91.4)	232.9 (173.1)	355.3 (203.2)	722.4 (292.0)
2000	263.3 (135.6)	416.0 (181.9)	721.5 (271.2)	1637.9 (352.2)
5000	492.2 (137.9)	873.8 (221.6)	1637.1 (266.9)	3926.8 (381.3)

Table B.4: Memory usage for forgeNet-RF and the corresponding RF model alone. Since we used the GPU version of the Tensorflow deep learning library, forgeNet merely induced additional space cost in terms of GPU memory only, and the RAM usage remained the same as the corresponding tree-ensemble method. The extra (GPU) memory usage of forgeNet-RF is shown in a bracket. For example, "141.2 (73.1)" means the RF extractor used 141.2 MB RAM memory, and to train the entire forgeNet, 73.1 MB extra GPU memory were also used.

Time (sec)	#features			
#samples	5000	10000	20000	50000
400	5.5/11.0	8.4/14.4	11.8/17.9	26/32.7
800	8.3/17.9	13.3/23.5	23.7/33.6	49.9/59.9
2000	20.4/41.7	38.4/59.3	68.3/89.8	143/159.3
5000	57.1/104.9	114.5/177.1	190.7/237.4	539/604.9

Table B.5: Computational time for forgeNet-GBM and the corresponding GBM model alone. The time used by GBM and by forgeNet-GBM are separated by "/". For example, "5.5/11.0" means the time of running GBM is 5.5 seconds while running the entire forgeNet takes 11.0 seconds.

Memory (MB)	#features			
#samples	5000	10000	20000	50000
400	138.4 (2.4)	169.4 (4.5)	231.3 (2.2)	417 (2.8)
800	168.9 (6.5)	230.4 (8.7)	353.4 (10.1)	722.2 (10.9)
2000	260.5 (25.2)	413.5 (44.4)	719.6 (38.6)	1637.7 (21.9)
5000	489.5 (70.5)	871.4 (145.7)	1635.2 (72.9)	3926.6 (166.6)

Table B.6: Memory usage for forgeNet-GBM and the corresponding GBM model alone. Since we used the GPU version of the Tensorflow deep learning library, forgeNet merely induced additional space cost in terms of GPU memory only, and the RAM usage remained the same as the corresponding tree-ensemble method. The extra (GPU) memory usage of forgeNet-GBM is shown in a bracket. For example, "138.4 (2.4)" means the GBM extractor used 138.4 MB RAM memory, and to train the entire forgeNet, 2.4 MB extra GPU memory were also used.

matrices from the 50 synthetic datasets generated in Section 4.3.1 for the case of 15 true features. The classification outcomes were randomly sampled from a Bernoulli distribution with probability 0.5. In this setting, the feature matrices contained no information regarding the outcomes.

The classification results are shown in Table B.7. It turned out that, with "null" datasets, all methods had an average of testing ROC-AUC around 0.5. Thus, we conclude that the false positive effect was not observed.

Methods	forgeNet-RF	RF	forgeNet-GBM	GBM	LRL	GEDFN	GEDFN_mis
Avg. ROC-AUC	0.511	0.525	0.535	0.511	0.493	0.479	0.513
s.d.	0.063	0.055	0.066	0.076	0.066	0.072	0.059

Table B.7: Classification results for "null" datasets

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. and Zheng, X. (2016), Tensorflow: A system for large-scale machine learning, in '12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)', pp. 265–283.
URL: *https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf*

Algamal, Z. Y. and Lee, M. H. (2015), 'Penalized logistic regression with the adaptive lasso for gene selection in high-dimensional cancer classification', Expert Systems with Applications **42**(23), 9326–9332.

Ambrosio, S., Sacca, C. D., Amente, S., Paladino, S., Lania, L. and Majello, B. (2017), 'Lysine-specific demethylase LSD1 regulates autophagy in neuroblastoma through SESN2-dependent pathway', Oncogene **36**(48), 6701–6711.

Babst, M., Wendland, B., Estepa, E. J. and Emr, S. D. (1998), 'The Vps4p AAA ATPase regulates membrane association of a Vps protein complex required for normal endosome function', EMBO J. **17**(11), 2982–2993.

Bai, J., Xie, X., Lei, Y., An, G., He, L. and Lv, X. (2014), 'Ocular albinism type 1-induced melanoma cell migration is mediated through the RAS/RAF/MEK/ERK signaling pathway', Mol Med Rep **10**(1), 491–495.

Bailey, S. T., Shin, H., Westerling, T., Liu, X. S. and Brown, M. (2012), 'Estrogen receptor prevents p53-dependent apoptosis in breast cancer', Proc. Natl. Acad. Sci. U.S.A. **109**(44), 18060–18065.

Banumathy, G. and Cairns, P. (2010), 'Signaling pathways in renal cell carcinoma', Cancer Biol. Ther. **10**(7), 658–664.

Barabási, A.-L. and Albert, R. (1999), 'Emergence of scaling in random networks', science **286**(5439), 509–512.

Barzel, B. and Barabási, A.-L. (2013), 'Universality in network dynamics', Nature physics **9**(10), 673–681.

Bassi, M. T., Schiaffino, M. V., Renieri, A., De Nigris, F., Galli, L., Bruttini, M., Gebbia, M., Bergen, A. A., Lewis, R. A. and Ballabio, A. (1995), 'Cloning of the gene for ocular albinism type 1 from the distal short arm of the X chromosome', Nat. Genet. **10**(1), 13–19.

Beloribi-Djefaflia, S., Vasseur, S. and Guillaumond, F. (2016), 'Lipid metabolic reprogramming in cancer cells', Oncogenesis **5**, e189.

Berge, C. and Minieka, E. (1973), Graphs and hypergraphs, Vol. 7, North-Holland publishing company Amsterdam.

Bondy, J. A., Murty, U. S. R. et al. (1976), Graph theory with applications, Vol. 290, Macmillan London.

Boscolo, R., Liao, J. C. and Roychowdhury, V. P. (2008), 'An information theoretic exploratory method for learning patterns of conditional gene coexpression from microarray data', IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **5**(1), 15–24.

Breiman, L. (1996), 'Bagging predictors', Machine learning **24**(2), 123–140.

Breiman, L. (2001), 'Random forests', Machine learning **45**(1), 5–32.

Bruna, J., Zaremba, W., Szlam, A. and LeCun, Y. (2013), 'Spectral networks and locally connected networks on graphs', arXiv preprint arXiv:1312.6203 .

Cai, G., Wu, D., Wang, Z., Xu, Z., Wong, K. B., Ng, C. F., Chan, F. L. and Yu, S. (2017), 'Collapsin response mediator protein-1 (CRMP1) acts as an invasion and metastasis suppressor of prostate cancer via its suppression of epithelial-mesenchymal transition and remodeling of actin cytoskeleton organization', Oncogene **36**(4), 546–558.

Cai, Z., Xu, D., Zhang, Q., Zhang, J., Ngai, S.-M. and Shao, J. (2015), 'Classification of lung cancer using ensemble-based feature selection and machine learning methods', Molecular BioSystems **11**(3), 791–800.

Caldon, C. E. (2014), 'Estrogen signaling and the DNA damage response in hormone dependent breast cancers', Front Oncol **4**, 106.

Cao, W., Li, J., Hao, Q., Vadgama, J. V. and Wu, Y. (2019), 'AMP-activated protein kinase: a potential therapeutic target for triple-negative breast cancer', Breast Cancer Res. **21**(1), 29.

Carey, G. B., Roy, S. K. and Daino, H. (2015), 'The natural tumorcide Manumycin-A targets protein phosphatase 1Î± and reduces hydrogen peroxide to induce lymphoma apoptosis', Exp. Cell Res. **332**(1), 136–145.

Caterson, B. and Melrose, J. (2018), 'Keratan sulfate, a complex glycosaminoglycan with unique functional capability', Glycobiology **28**(4), 182–206.

Chen, J., Xie, J. and Li, H. (2011), 'A penalized likelihood approach for bivariate conditional normal models for dynamic co-expression analysis', Biometrics **67**(1), 299–308.

Chen, T. and Guestrin, C. (2016), XGBoost: A scalable tree boosting system, in 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '16, ACM, New York, NY, USA, pp. 785–794. **URL:** *http://doi.acm.org/10.1145/2939672.2939785*

Chen, Y.-C., Ke, W.-C. and Chiu, H.-W. (2014), 'Risk classification of cancer survival using ann with gene expression data from multiple laboratories', Computers in biology and medicine **48**, 1–7.

Cheng, C. W., Leong, K. W. and Tse, E. (2016), 'Understanding the role of PIN1 in hepatocellular carcinoma', World J. Gastroenterol. **22**(45), 9921–9932.

Chowdhury, S. and Sarkar, R. R. (2015), 'Comparison of human cell signaling pathway databases–evolution, drawbacks and challenges', Database (Oxford) **2015**.

Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D. and Ideker, T. (2007), 'Network-based classification of breast cancer metastasis', Molecular systems biology **3**(1), 140.

Clauset, A., Newman, M. E. and Moore, C. (2004), 'Finding community structure in very large networks', Physical review E **70**(6), 066111.

Consortium, G. O. et al. (2015), 'Gene ontology consortium: going forward', Nucleic acids research **43**(D1), D1049–D1056.

Cook, K. L. and Clarke, R. (2014), 'Estrogen receptor-Î± signaling and localization regulates autophagy and unfolded protein response activation in ER+ breast cancer', Receptors Clin Investig **1**(6).

Corn, P. G. (2007), 'Role of the ubiquitin proteasome system in renal cell carcinoma', BMC Biochem. **8 Suppl 1**, S4.

Creighton, C. J. e. a. (2013), 'Comprehensive molecular characterization of clear cell renal cell carcinoma', Nature **499**(7456), 43–49.

Das, J. and Yu, H. (2012), 'HINT: High-quality protein interactomes and their applications in understanding human disease', BMC Syst Biol **6**, 92.

De Oña, J. and Garrido, C. (2014), 'Extracting the contribution of independent variables in neural network models: a new approach to handle instability', Neural Computing and Applications **25**(3-4), 859–869.

Dimopoulos, Y., Bourret, P. and Lek, S. (1995), 'Use of some sensitivity criteria for choosing networks with good generalization ability', Neural Processing Letters **2**(6), 1–4.

Duong, T. (2007), 'ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r', Journal of Statistical Software **21**(7), 1–16.

Dutkowski, J. and Ideker, T. (2011), 'Protein networks as logic functions in development and cancer', PLoS computational biology **7**(9), e1002180.

Edeline, J., Vigneau, C., Patard, J. J. and Rioux-Leclercq, N. (2010), '[Signalling pathways in renal-cell carcinoma: from the molecular biology to the future therapy]', Bull Cancer **97**, 5–15.

Erdös, P. and Rényi, A. (1959), 'On random graphs, i', Publicationes Mathematicae (Debrecen) **6**, 290–297.

Falcon, S. and Gentleman, R. (2007), 'Using GOstats to test gene lists for GO term association', Bioinformatics **23**(2), 257–258.

Felzen, V., Hiebel, C., Koziollek-Drechsler, I., Reissig, S., Wolfrum, U., Kogel, D., Brandts, C., Behl, C. and Morawe, T. (2015), 'Estrogen receptor Î± regulates non-canonical autophagy that provides stress resistance to neuroblastoma and breast cancer cells and involves BAG3 function', Cell Death Dis **6**, e1812.

Filardo, E. J., Quinn, J. A., Frackelton, A. R. and Bland, K. I. (2002), 'Estrogen action via the G protein-coupled receptor, GPR30: stimulation of adenylyl cyclase and cAMP-mediated attenuation of the epidermal growth factor receptor-to-MAPK signaling axis', Mol. Endocrinol. **16**(1), 70–84.

Friedman, J. H. (2002), 'Stochastic gradient boosting', Computational statistics & data analysis **38**(4), 367–378.

Gilkes, D. M. and Semenza, G. L. (2013), 'Role of hypoxia-inducible factors in breast cancer metastasis', Future Oncol **9**(11), 1623–1636.

Gionet, N., Jansson, D., Mader, S. and Pratt, M. A. (2009), 'NF-kappaB and estrogen receptor alpha interactions: Differential function in estrogen receptor-negative and -positive hormone-independent breast cancer cells', J. Cell. Biochem. **107**(3), 448–459.

Goldstein, I., Ezra, O., Rivlin, N., Molchadsky, A., Madar, S., Goldfinger, N. and Rotter, V. (2012), 'p53, a novel regulator of lipid metabolism pathways', J. Hepatol. **56**(3), 656–662.

Goodfellow, I., Bengio, Y. and Courville, A. (2016), Deep learning, MIT press.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J. and Seung, H. S. (2000), 'Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit', Nature **405**(6789), 947–951.

Hastie, T., Tibshirani, R., Narasimhan, B. and Chu, G. (2018), impute: impute: Imputation for microarray data. R package version 1.56.0.

Haydn, J. M., Hufnagel, A., Grimm, J., Maurus, K., Schartl, M. and Meierjohann, S. (2014), 'The MAPK pathway as an apoptosis enhancer in melanoma', Oncotarget **5**(13), 5040–5053.

Henaff, M., Bruna, J. and LeCun, Y. (2015), 'Deep convolutional networks on graph-structured data', arXiv preprint arXiv:1506.05163 .

Hernando, B., Ibarrola-Villava, M., Fernandez, L. P., Pena-Chilet, M., Llorca-Cardenosa, M., Oltra, S. S., Alonso, S., Boyano, M. D., Martinez-Cadenas, C. and Ribas, G. (2016), 'Sex-specific genetic effects associated with pigmentation, sensitivity to sunlight, and melanoma in a population of Spanish origin', Biol Sex Differ **7**, 17.

Hilvo, M., de Santiago, I., Gopalacharyulu, P., Schmitt, W. D., Budczies, J., Kuhberg, M., Dietel, M., Aittokallio, T., Markowetz, F., Denkert, C., Sehouli, J., Frezza, C., Darb-Esfahani, S. and Braicu, E. I. (2016), 'Accumulated Metabolites of Hydroxybutyric Acid Serve as Diagnostic and Prognostic Biomarkers of Ovarian High-Grade Serous Carcinomas', Cancer Res. **76**(4), 796–804.

Ho, Y.-Y., Parmigiani, G., Louis, T. A. and Cope, L. M. (2011), 'Modeling liquid association', Biometrics **67**(1), 133–141.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J. et al. (2001), 'Gradient flow in recurrent nets: the difficulty of learning long-term dependencies'.

Hoesel, B. and Schmid, J. A. (2013), 'The complexity of NF-Î¹B signaling in inflammation and cancer', Mol. Cancer **12**, 86.

Ideker, T. and Krogan, N. J. (2012), 'Differential network biology', Molecular systems biology **8**(1), 565.

Improta-Brears, T., Whorton, A. R., Codazzi, F., York, J. D., Meyer, T. and McDonnell, D. P. (1999), 'Estrogen-induced activation of mitogen-activated protein kinase requires mobilization of intracellular calcium', Proc. Natl. Acad. Sci. U.S.A. **96**(8), 4686–4691.

INRA and Leger, J.-B. (2015), blockmodels: Latent and Stochastic Block Model Estimation by a 'V-EM' Algorithm. R package version 1.1.1.
URL: *https://CRAN.R-project.org/package=blockmodels*

JavanMoghadam, S., Weihua, Z., Hunt, K. K. and Keyomarsi, K. (2016), 'Estrogen receptor alpha is cell cycle-regulated and regulates the cell cycle in a ligand-dependent fashion', Cell Cycle **15**(12), 1579–1590.

Jin, L., Shen, K., Chen, T., Yu, W. and Zhang, H. (2017), 'SUMO-1 Gene Silencing Inhibits Proliferation and Promotes Apoptosis of Human Gastric Cancer SGC-7901 Cells', Cell. Physiol. Biochem. **41**(3), 987–998.

Jones, M. R., Schrader, K. A., Shen, Y., Pleasance, E., Ch'ng, C., Dar, N., Yip, S., Renouf, D. J., Schein, J. E., Mungall, A. J., Zhao, Y., Moore, R., Ma, Y., Sheffield, B. S., Ng, T., Jones, S. J., Marra, M. A., Laskin, J. and Lim, H. J. (2016), 'Response to angiotensin blockade with irbesartan in a patient with metastatic colorectal cancer', Ann. Oncol. **27**(5), 801–806.

Jung, Y.-S., Chun, H.-Y., Yoon, M.-H. and Park, B.-J. (2014), 'Elevated estrogen receptor-α in vhl-deficient condition induces microtubule organizing center amplification via disruption of brca1/rad51 interaction', Neoplasia **16**(12), 1070–1081.

Kang, S. H., Kang, K. W., Kim, K. H., Kwon, B., Kim, S. K., Lee, H. Y., Kong, S. Y., Lee, E. S., Jang, S. G. and Yoo, B. C. (2008), 'Upregulated HSP27 in human breast cancer cells reduces Herceptin susceptibility by increasing Her2 protein stability', BMC Cancer **8**, 286.

Kato, S., Endoh, H., Masuhiro, Y., Kitamoto, T., Uchiyama, S., Sasaki, H., Masushige, S., Gotoh, Y., Nishida, E., Kawashima, H., Metzger, D. and Chambon, P. (1995), 'Activation of the estrogen receptor through phosphorylation by mitogen-activated protein kinase', Science **270**(5241), 1491–1494.

Kim, S., Pan, W. and Shen, X. (2013), 'Network-based penalized regression with application to genomic data', Biometrics **69**(3), 582–593.

Kingma, D. P. and Ba, J. (2014), 'Adam: A method for stochastic optimization', CoRR **abs/1412.6980**.
URL: *http://arxiv.org/abs/1412.6980*

Ko, J. H., Ko, E. A., Gu, W., Lim, I., Bang, H. and Zhou, T. (2013), 'Expression profiling of ion channel genes predicts clinical outcome in breast cancer', Mol. Cancer **12**(1), 106.

Koboldt, D. C., Fulton, R. S., McLellan, M. D., Schmidt, H., Kalicki-Veizer, J., McMichael, J. F., Fulton, L. L., Dooling, D. J., Ding, L., Mardis, E. R. et al. (2012), 'Comprehensive molecular portraits of human breast tumours', Nature **490**, 61–70.

Kolaczyk, E. D. (2009), Statistical Analysis of Network Data: Methods and Models, 1st edn, Springer Publishing Company, Incorporated.

Kong, Y. and Yu, T. (2018), 'A deep neural network model using random forest to extract feature representation for gene expression data classification', Scientific reports **8**(1), 16477.

Kursa, M. B. (2014), 'Robustness of random forest-based gene selection methods', BMC bioinformatics **15**(1), 8.

Langfelder, P., Zhang, B. and Horvath, S. (2008), 'Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r', Bioinformatics **24**(5), 719–720.

Lanzino, M., Morelli, C., Garofalo, C., Panno, M. L., Mauro, L., Ando, S. and Sisci, D. (2008), 'Interaction between estrogen receptor alpha and insulin/IGF signaling in breast cancer', Curr Cancer Drug Targets **8**(7), 597–610.

Lavi, O., Dror, G. and Shamir, R. (2012), 'Network-induced classification kernels for gene expression profile analysis', Journal of Computational Biology **19**(6), 694–709.

LeCun, Y., Bengio, Y. and Hinton, G. (2015), 'Deep learning', Nature **521**(7553), 436–444.

LeCun, Y. and Cortes, C. (2010), 'MNIST handwritten digit database'.
URL: *http://yann.lecun.com/exdb/mnist/*

Leiphrakpam, P. D., Patil, P. P., Remmers, N., Swanson, B., Grandgenett, P. M., Qiu, F., Yu, F. and Radhakrishnan, P. (2019), 'Role of keratan sulfate expression in human pancreatic cancer malignancy', Sci Rep **9**(1), 9665.

Li, C. and Li, H. (2008), 'Network-constrained regularization and variable selection for analysis of genomic data', Bioinformatics **24**(9), 1175–1182.

Li, J., Chen, C., Bi, X., Zhou, C., Huang, T., Ni, C., Yang, P., Chen, S., Ye, M. and Duan, S. (2017), 'DNA methylation of CMTM3, SSTR2, and MDFI genes in colorectal cancer', Gene **630**, 1–7.

Li, J. J., Weroha, S. J., Lingle, W. L., Papa, D., Salisbury, J. L. and Li, S. A. (2004), 'Estrogen mediates Aurora-A overexpression, centrosome amplification, chromosomal instability, and breast cancer in female ACI rats', Proc. Natl. Acad. Sci. U.S.A. **101**(52), 18123–18128.

Li, K.-C. (2002), 'Genome-wide coexpression dynamics: theory and application', Proceedings of the National Academy of Sciences **99**(26), 16875–16880.

Li, K.-C., Liu, C.-T., Sun, W., Yuan, S. and Yu, T. (2004), 'A system for enhancing genome-wide coexpression dynamics study', Proceedings of the National Academy of Sciences of the United States of America **101**(44), 15561–15566.

Li, S., Park, Y., Duraisingham, S., Strobel, F. H., Khan, N., Soltow, Q. A., Jones, D. P. and Pulendran, B. (2013), 'Predicting network activity from high throughput metabolomics', PLoS Comput. Biol. **9**(7), e1003123.

Liang, Y., Liu, C., Luan, X.-Z., Leung, K.-S., Chan, T.-M., Xu, Z.-B. and Zhang, H. (2013), 'Sparse logistic regression with a l 1/2 penalty for gene selection in cancer classification', BMC bioinformatics **14**(1), 198.

Linge, A., Kennedy, S., O'Flynn, D., Beatty, S., Moriarty, P., Henry, M., Clynes, M., Larkin, A. and Meleady, P. (2012), 'Differential expression of fourteen proteins between uveal melanoma from patients who subsequently developed distant metastases versus those who did Not', Invest. Ophthalmol. Vis. Sci. **53**(8), 4634–4643.

Liu, J., Jiang, G., Mao, P., Zhang, J., Zhang, L., Liu, L., Wang, J., Owusu, L., Ren, B., Tang, Y. and Li, W. (2018), 'Down-regulation of GADD45A enhances chemosensitivity in melanoma', Sci Rep **8**(1), 4111.

Luo, X., Cheng, C., Tan, Z., Li, N., Tang, M., Yang, L. and Cao, Y. (2017), 'Emerging roles of lipid metabolism in cancer metastasis', Mol. Cancer **16**(1), 76.

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A. and Gerstein, M. (2004), 'Genomic analysis of regulatory network dynamics reveals large topological changes', Nature **431**(7006), 308–312.

Ma, S. and Dai, Y. (2011), 'Principal component analysis based methods in bioinformatics studies', Briefings in bioinformatics **12**(6), 714–722.

Mariadassou, M., Robin, S. and Vacher, C. (2010), 'Uncovering latent structure in valued graphs: a variational approach', The Annals of Applied Statistics pp. 715–742.

McManaman, J. L. and Neville, M. C. (2003), 'Mammary physiology and milk secretion', Adv. Drug Deliv. Rev. **55**(5), 629–641.

McSherry, E. A., Brennan, K., Hudson, L., Hill, A. D. and Hopkins, A. M. (2011), 'Breast cancer cell migration is regulated through junctional adhesion molecule-A-mediated activation of Rap1 GTPase', Breast Cancer Res. **13**(2), R31.

Min, S., Lee, B. and Yoon, S. (2016), 'Deep learning in bioinformatics', Briefings in bioinformatics p. bbw068.

Minner, S., Rump, D., Tennstedt, P., Simon, R., Burandt, E., Terracciano, L., Moch, H., Wilczak, W., Bokemeyer, C., Fisch, M., Sauter, G. and Eichelberg, C. (2012), 'Epidermal growth factor receptor protein expression and genomic alterations in renal cell carcinoma', Cancer **118**(5), 1268–1275.

Mitxelena, J., Apraiz, A., Vallejo-Rodriguez, J., Garcia-Santisteban, I., Fullaondo, A., Alvarez-Fernandez, M., Malumbres, M. and Zubiaga, A. M. (2018), 'An E2F7-dependent transcriptional program modulates DNA damage repair and genomic stability', Nucleic Acids Res. **46**(9), 4546–4559.

Mockus, J. (2012), Bayesian approach to global optimization: theory and applications, Vol. 37, Springer Science & Business Media.

Monaco, M. E. (2017), 'Fatty acid metabolism in breast cancer subtypes', Oncotarget **8**(17), 29487–29500.

Na, K., Kim, E. K., Jang, W. and Kim, H. S. (2017), 'CTNNB1 Mutations in Ovarian Microcystic Stromal Tumors: Identification of a Novel Deletion Mutation and the Use of Pyrosequencing to Identify Reported Point Mutation', Anticancer Res. **37**(6), 3249–3258.

Nair, V. and Hinton, G. E. (2010), Rectified linear units improve restricted boltzmann machines, in 'Proceedings of the 27th international conference on machine learning (ICML-10)', pp. 807–814.

Network, C. G. A. R. et al. (2013), 'Comprehensive molecular characterization of clear cell renal cell carcinoma', Nature **499**(7456), 43.

Nikitovic, D., Berdiaki, A., Spyridaki, I., Krasanakis, T., Tsatsakis, A. and Tzanakakis, G. N. (2018), 'Proteoglycans-Biomarkers and Targets in Cancer Therapy', Front Endocrinol (Lausanne) **9**, 69.

Nowicki, K. and Snijders, T. A. B. (2001), 'Estimation and prediction for stochastic blockstructures', Journal of the American statistical association **96**(455), 1077–1087.

Ocone, A., Millar, A. J. and Sanguinetti, G. (2013), 'Hybrid regulatory models: a statistically tractable approach to model regulatory network dynamics', Bioinformatics **29**(7), 910–916.

Olden, J. D. and Jackson, D. A. (2002), 'Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks', Ecological modelling **154**(1-2), 135–150.

Osborne, C. K., Shou, J., Massarweh, S. and Schiff, R. (2005), 'Crosstalk between estrogen receptor and growth factor receptor pathways as a cause for endocrine therapy resistance in breast cancer', Clin. Cancer Res. **11**(2 Pt 2), 865s–70s.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', Journal of Machine Learning Research **12**, 2825–2830.

Pronk, J. T., Yde Steensma, H. and Van Dijken, J. P. (1996), 'Pyruvate metabolism in Saccharomyces cerevisiae', Yeast **12**(16), 1607–1633.

Quezada, A. and Vafai, K. (2014), 'Modeling and analysis of transport in the mammary glands', Physical Biology **11**(4), 045004.

Rebecca, V. W., Nicastri, M. C., McLaughlin, N., Fennelly, C., McAfee, Q., Ronghe, A., Nofal, M., Lim, C. Y., Witze, E., Chude, C. I., Zhang, G., Alicea, G. M., Piao, S., Murugan, S., Ojha, R., Levi, S. M., Wei, Z., Barber-Rotenberg, J. S., Murphy, M. E., Mills, G. B., Lu, Y., Rabinowitz, J., Marmorstein, R., Liu, Q., Liu, S., Xu, X., Herlyn, M., Zoncu, R., Brady, D. C., Speicher, D. W., Winkler, J. D. and Amaravadi, R. K. (2017), 'A Unified Approach to Targeting the Lysosome's Degradative and Growth Signaling Roles', Cancer Discov **7**(11), 1266–1283.

Recio-Boiles, A., Ilmer, M., Rhea, P. R., Kettlun, C., Heinemann, M. L., Ruetering, J., Vykoukal, J. and Alt, E. (2016), 'JNK pathway inhibition selectively primes pancreatic cancer stem cells to TRAIL-induced apoptosis without affecting the physiology of normal tissue resident stem cells', Oncotarget **7**(9), 9890–9906.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L. (2015), 'ImageNet Large Scale Visual Recognition Challenge', International Journal of Computer Vision (IJCV) **115**(3), 211–252.

Saito, S., Yamashita, S., Endoh, M., Yamato, T., Hoshi, S., Ohyama, C., Watanabe, R., Ito, A., Satoh, M., Wada, T., Paulson, J. C., Arai, Y. and Miyagi, T. (2002), 'Clinical significance of ST3Gal IV expression in human renal cell carcinoma', Oncol. Rep. **9**(6), 1251–1255.

Samanta, D. and Datta, P. K. (2012), 'Alterations in the Smad pathway in human cancers', Front Biosci (Landmark Ed) **17**, 1281–1293.

Sarma, N. J. and Yaseen, N. R. (2011), 'Amino-terminal enhancer of split (AES) interacts with the oncoprotein NUP98-HOXA9 and enhances its transforming ability', J. Biol. Chem. **286**(45), 38989–39001.

Sorlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., Geisler, S., Demeter, J., Perou, C. M., L?nning, P. E., Brown, P. O., B?rresen-Dale, A. L. and Botstein, D. (2003), 'Repeated observation of breast tumor subtypes in independent gene expression data sets', Proc. Natl. Acad. Sci. U.S.A. **100**(14), 8418–8423.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D. and Futcher, B. (1998), 'Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization', Mol. Biol. Cell **9**(12), 3273–3297.

Sun, M., Zhao, X., Liang, L., Pan, X., Lv, H. and Zhao, Y. (2017), 'Sialyltransferase ST3GAL6 mediates the effect of microRNA-26a on cell growth, migration, and invasion in hepatocellular carcinoma through the protein kinase B/mammalian target of rapamycin pathway', Cancer Sci. **108**(2), 267–276.

Szklarczyk, D. and Jensen, L. J. (2015), 'Protein-protein interaction databases', Methods Mol. Biol. **1278**, 39–56.

Taanman, J. W. and Capaldi, R. A. (1992), 'Purification of yeast cytochrome c oxidase with a subunit composition resembling the mammalian enzyme', J. Biol. Chem. **267**(31), 22481–22485.

Tang, A. and Foong, J. T. (2014), A qualitative evaluation of random forest feature learning, in 'Recent Advances on Soft Computing and Data Mining', Springer, pp. 359–368.

Tang, Y., Yan, G., Song, X., Wu, K., Li, Z., Yang, C., Deng, T., Sun, Y., Hu, X., Yang, C., Bai, H., Li, H., Tan, W., Ye, M. and Liu, J. (2015), 'STIP overexpression confers oncogenic potential to human non-small cell lung cancer cells by regulating cell cycle and apoptosis', J. Cell. Mol. Med. **19**(12), 2806–2817.

Tecalco-Cruz, A. C. and Ramirez-Jarquin, J. O. (2017), 'Mechanisms that Increase Stability of Estrogen Receptor Alpha in Breast Cancer', Clin. Breast Cancer **17**(1), 1–10.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', Journal of the Royal Statistical Society. Series B (Methodological) pp. 267–288.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R. B. (2001), 'Missing value estimation methods for DNA microarrays', Bioinformatics **17**(6), 520–525.

Vanitha, C. D. A., Devaraj, D. and Venkatesulu, M. (2015), 'Gene expression data classification using support vector machine and mutual information-based gene selection', Procedia Computer Science **47**, 13–21.

Vens, C. and Costa, F. (2011), Random forest based feature induction, in 'Data Mining (ICDM), 2011 IEEE 11th International Conference on', IEEE, pp. 744–753.

Vlachos, I. S., Kostoulas, N., Vergoulis, T., Georgakilas, G., Reczko, M., Maragkakis, M., Paraskevopoulou, M. D., Prionidis, K., Dalamagas, T. and Hatzigeorgiou, A. G. (2012), 'DIANA miRPath v.2.0: investigating the combinatorial effect of microRNAs in pathways', Nucleic Acids Res. **40**(Web Server issue), 498–504.

Walker, C., Mojares, E. and Del R?o Hern?ndez, A. (2018), 'Role of Extracellular Matrix in Development and Cancer Progression', Int J Mol Sci **19**(10).

Wang, L., Chen, G. and Li, H. (2007), 'Group scad regression analysis for microarray time course gene expression data', Bioinformatics **23**(12), 1486–1494.

Wang, L., Liu, S., Ding, Y., Yuan, S. S., Ho, Y. Y. and Tseng, G. C. (2017), 'Meta-analytic framework for liquid association', Bioinformatics **33**(14), 2140–2147.

Wei, C., Wang, Y. and Li, X. (2018), 'The role of Hippo signal pathway in breast cancer metastasis', Onco Targets Ther **11**, 2185–2193.

Wei, P. and Pan, W. (2007), 'Incorporating gene networks into statistical tests for genomic data via a spatially correlated mixture model', Bioinformatics **24**(3), 404–411.

Wei, W., Chen, Z. J., Zhang, K. S., Yang, X. L., Wu, Y. M., Chen, X. H., Huang, H. B., Liu, H. L., Cai, S. H., Du, J. and Wang, H. S. (2014), 'The activation of G protein-coupled receptor 30 (GPR30) inhibits proliferation of estrogen receptor-negative breast cancer cells in vitro and in vivo', Cell Death Dis **5**, e1428.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J. M., Network, C. G. A. R. et al. (2013), 'The cancer genome atlas pan-cancer analysis project', Nature genetics **45**(10), 1113–1120.

Wolff, M., Kosyna, F. K., Dunst, J., Jelkmann, W. and Depping, R. (2017), 'Impact of hypoxia inducible factors on estrogen receptor expression in breast cancer cells', Arch. Biochem. Biophys. **613**, 23–30.

Wu, X., Zhang, X., Yu, L., Zhang, C., Ye, L., Ren, D., Li, Y., Sun, X., Yu, L., Ouyang, Y., Chen, X., Song, L., Liu, P. and Lin, X. (2020), 'Zinc finger protein 367 promotes metastasis by inhibiting the Hippo pathway in breast cancer', Oncogene .

Xu, R. and Wunsch, D. C. (2010), 'Clustering algorithms in biomedical research: A review', IEEE Reviews in Biomedical Engineering **3**, 120–154.

Yamamoto, T., Saatcioglu, F. and Matsuda, T. (2002), 'Cross-talk between bone morphogenic proteins and estrogen receptor signaling', Endocrinology **143**(7), 2635–2642.

Yan, Y., Qiu, S., Jin, Z., Gong, S., Bai, Y., Lu, J. and Yu, T. (2017), 'Detecting subnetwork-level dynamic correlations', Bioinformatics **33**(2), 256–265.

Yoshinaga, M., Tanaka, S., Shimago, A., Sameshima, K., Nishi, J., Nomura, Y., Kawano, Y., Hashiguchi, J., Ichiki, T. and Shimizu, S. (2005), 'Metabolic syndrome in overweight and obese Japanese children', Obes. Res. **13**(7), 1135–1140.

Yu, T. (2018), 'A new dynamic correlation algorithm reveals novel functional aspects in single cell and bulk rna-seq data', PLoS computational biology **14**(8), e1006391.

Yu, T. and Li, K. C. (2005), 'Inference of transcriptional regulatory network by two-stage constrained space factor analysis', Bioinformatics **21**(21), 4033–4038.

Yu, T., Park, Y., Johnson, J. M. and Jones, D. P. (2009), 'apLCMS–adaptive processing of high-resolution LC/MS data', Bioinformatics **25**(15), 1930–1936.

Yu, T., Park, Y., Li, S. and Jones, D. P. (2013), 'Hybrid feature detection and information accumulation using high-resolution LC-MS metabolomics data', J. Proteome Res. **12**(3), 1419–1427.

Yu, T., Sun, W., Yuan, S. and Li, K. C. (2005), 'Study of coordinative gene expression at the biological process level', Bioinformatics **21**(18), 3651–3657.

Zhang, J., Ji, Y. and Zhang, L. (2007), 'Extracting three-way gene interactions from microarray data', Bioinformatics **23**(21), 2903–2909.

Zhang, L., Ye, Y., Long, X., Xiao, P., Ren, X. and Yu, J. (2016), 'BMP signaling and its paradoxical effects in tumorigenesis and dissemination', Oncotarget **7**(47), 78206–78218.

Zhao, H., Orhan, Y. C., Zha, X., Esencan, E., Chatterton, R. T. and Bulun, S. E. (2017), 'AMP-activated protein kinase and energy balance in breast cancer', Am J Transl Res **9**(2), 197–213.

Zhao, Y., Kang, J. and Yu, T. (2014), 'A bayesian nonparametric mixture model for selecting genes and gene subnetworks', The annals of applied statistics **8**(2), 999.

Zheng, X., Resnick, R. J. and Shalloway, D. (2008), 'Apoptosis of estrogen-receptor negative breast cancer and colon cancer cell lines by PTP alpha and src RNAi', Int. J. Cancer **122**(9), 1999–2007.

Zhu, Y., Shen, X. and Pan, W. (2009), 'Network-based support vector machine for classification of microarray samples', BMC bioinformatics **10**(1), S21.

Zhuo, D., Li, X. and Guan, F. (2018), 'Biological Roles of Aberrantly Expressed Glycosphingolipids and Related Enzymes in Human Cancer Development and Progression', Front Physiol **9**, 466.

Zou, Y.-F., Xie, C.-W., Yang, S.-X. and Xiong, J.-P. (2017), 'Ampk activators suppress breast cancer cell growth by inhibiting dvl3-facilitated wnt/β-catenin signaling pathway activity', Molecular medicine reports **15**(2), 899–907.