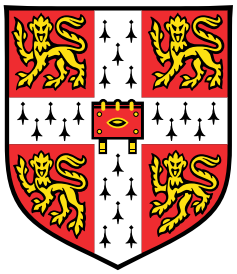
Network-based approaches for multi-omic data integration



Hui Xiao

The Computer Laboratory
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Darwin College

April 2018

Network-based approaches for multi-omic data integration

Hui Xiao

The advent of advanced high-throughput biological technologies provides opportunities to measure the whole genome at different molecular levels in biological systems, which produces different types of omic data such as genome, epigenome, transcriptome, translatome, and interactome. In order to uncover the systematic complexity of biological systems, it is desirable to integrate multi-omic data to transform the multiple level data into biological knowledge about the underlying mechanisms. Due to the heterogeneity and high-dimension of multi-omic data, it is necessary to develop effective and efficient methods for multi-omic data integration.

This thesis aims to develop efficient approaches for multi-omic data integration using machine learning methods and network theory. We assume that a biological system can be represented by a network with nodes denoting molecules and edges indicating functional links between molecules, in which multi-omic data can be integrated as attributes of nodes and edges. We propose four network-based approaches for multi-omic data integration using machine learning methods, with specific aims for (1) gene module detection by integrating multi-condition transcriptome and interactome data using network overlapping module detection method, (2) gene module detection by integrating transcriptome, translatome, and interactome data using multilayer network, (3) feature selection by integrating transcriptome and interactome data using network-constrained regression, and (4) classification by integrating epigenome and transcriptome data using neural networks. By applying the proposed approaches to multi-omic data of human cancer and early embryonic development, several underlying patterns are recognized through the data integration which reveal interested biological insights providing valuable clues for understanding the potential molecular mechanisms.

The approaches proposed in this thesis offer effective and efficient solutions for integration of heterogeneous high-dimensional datasets, which can be easily applied to other datasets presenting the similar structures. They are therefore applicable to many fields including but not limited to Bioinformatics and Computer Science.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Hui Xiao
April 2018

Acknowledgements

First and foremost, I would like to express my heartfelt thanks to my supervisor, Prof Pietro Lio', who has provided me with patient guidance, invaluable advice and constant encouragement throughout my PhD and always been helpful, supportive and understanding. His conscientious and open-minded academic spirit inspire me a lot in academic research.

I would like to extend my sincere gratitude to my cosupervisor, Prof Anne Ferguson-Smith, who has provided me with the great opportunity for interdisciplinary training in a world-leading research group. Her insightful academic guidance and constructive suggestions help me build the critical thinking skills which benefit my future career.

I am extremely grateful to Anne and Pietro for offering me the amazing Marie Curie Early Stage Researcher (ESR) Fellowship in the EU Marie Curie Initial Training Network (ITN) EpiHealthNet. This fellowship provides me not only the funding for undertaking the PhD in Cambridge but also the great opportunities for scientific and complementary training and multidisciplinary collaborations, which prepares me better for my future career in both academia and industry.

I am truly grateful to my examiners, Prof Anna Korhonen and Dr Andrea Bracciali, for their thorough reading of my thesis and the pleasant and instructive viva. Their valuable comments and constructive suggestions greatly help to improve this thesis.

Big thanks go to my project collaboraters, Prof Daniel Brison and Dr Helen Smith from The University of Manchester, Dr Giovanna Lazzari and Dr Maria Barandalla from Avantea Italy, Dr Matthew Trotter from Celgene Research Spain, Petar Velickovic and Ioana Bica from the Computer Laboratory, for the fruitful collaborations, the helpful discussions and the constructive suggestions. This thesis could not be completed without their help.

I deeply appreciate all the help and support from the Computer Laboratory and Darwin College. My appreciation extends to Dr Mateja Jamnik and Dr Sean Holden for their early feedbacks on my work. My thanks also go to all the members of the Computational Biology Research Group in the Computer Laboratory for the enjoyable group life.

I would like to express my heartfelt thanks to my friends for their care and company. Special thanks go to Dr Zheng Yuan, Dr Xue Gong, Dr Lei Hou and Dr Xiaoyuan Guo who are always being there for me and sharing with me my happiness, worries and frustrations all the time.

Last but not least, my deep heartfelt gratitude goes to my parents, Yuhua Xiao and Wenxia Xu, and my brother, Peng Xiao, whose continuous love and unconditional support encourage me to be myself.

Abstract

The advent of advanced high-throughput biological technologies provides opportunities to measure the whole genome at different molecular levels in biological systems, which produces different types of omic data such as genome, epigenome, transcriptome, translatome, proteome, metabolome and interactome. Biological systems are highly dynamic and complex mechanisms which involve not only the within-level functionality but also the between-level regulation. In order to uncover the complexity of biological systems, it is desirable to integrate multi-omic data to transform the multiple level data into biological knowledge about the underlying mechanisms. Due to the heterogeneity and high-dimension of multi-omic data, it is necessary to develop effective and efficient methods for multi-omic data integration.

This thesis aims to develop efficient approaches for multi-omic data integration using machine learning methods and network theory. We assume that a biological system can be represented by a network with nodes denoting molecules and edges indicating functional links between molecules, in which multi-omic data can be integrated as attributes of nodes and edges. We propose four network-based approaches for multi-omic data integration using machine learning methods. Firstly, we propose an approach for gene module detection by integrating multi-condition transcriptome data and interactome data using network overlapping module detection method. We apply the approach to study the transcriptome data of human pre-implantation embryos across multiple development stages, and identify several stage-specific dynamic functional modules and genes which provide interesting biological insights. We evaluate the reproducibility of the modules by comparing with some other widely used methods and show that the intra-module genes are significantly overlapped between the different methods. Secondly, we propose an approach for gene module detection by integrating transcriptome, translatome, and interactome data using multilayer network. We apply the approach to study the ribosome profiling data of mTOR perturbed human prostate cancer cells and mine several translation efficiency regulated modules associated with mTOR perturbation. We develop an R package, TERM, for implementation of the proposed approach which offers a useful tool for the research field. Next, we propose an approach for feature selection by integrating transcriptome and interactome data using network-constrained regression. We develop a more efficient

network-constrained regression method eGBL. We evaluate its performance in term of variable selection and prediction, and show that eGBL outperforms the other related regression methods. With application on the transcriptome data of human blastocysts, we select several interested genes associated with time-lapse parameters. Finally, we propose an approach for classification by integrating epigenome and transcriptome data using neural networks. We introduce a superlayer neural network (SNN) model which learns DNA methylation and gene expression data parallelly in superlayers but with cross-connections allowing crosstalks between them. We evaluate its performance on human breast cancer classification. The SNN provides superior performances and outperforms several other common machine learning methods.

The approaches proposed in this thesis offer effective and efficient solutions for integration of heterogeneous high-dimensional datasets, which can be easily applied to other datasets presenting the similar structures. They are therefore applicable to many fields including but not limited to Bioinformatics and Computer Science.

Contents

Chapter 1

Introduction

1.1 Motivation

The advent of advanced high-throughput technologies in molecular biology, such as microarray and next generation sequencing, provides opportunities to measure the whole genomes at different molecular levels in biological systems [1, 2]. The genome-wide experiments of the molecular levels produce various omic data including genome, epigenome, transcriptome, translatome, proteome, metabolome, interactome and so on. Each type of omic data has its unique characteristics and provides a comprehensive view of functionality at the corresponding molecular level. Most biological systems are highly dynamic and complex mechanisms which involves not only the within-level functionality but also the inter-level regulation. In order to uncover the complexity of biological systems, it is desirable to integrate the multi-omic data to transform the heterogeneous high-throughput data into biological knowledge about the underlying mechanisms [3, 4].

Machine learning techniques have been successfully used to carry out the biological knowledge transformation from omic data [5, 6]. Machine learning is the method of fitting an analytical model for the given data [7]. It is a data-driven process which automatically learn processing rules, identify patterns and make decisions. There are two main disciplines in machine learning, supervised learning and unsupervised learning. Supervised learning, applies on labelled data, infers the discriminating rules from the given data and then make predictions on the new unlabelled data. Unsupervised learning explores the data and infers the hidden structures from unlabelled data independently. Both of the two disciplines have been widely applied to analyse omic data to address biological questions. For example, given a transcriptome data which provides the genome-wide gene expression values in patients with different subtypes of cancer, the supervised learning can accurately predict patients into different clinical groups [8, 9, 10], and the unsupervised learning can identify new disease subtypes by clustering samples upon the similarities among their gene expressions [11, 12]. Enormous effort has been put into research and development of

machine learning methods for mining omic data, but mainly focused on single-omic data and does not take full advantage of multi-omic data.

In a complex biological system, molecules usually function coordinately, rather than independently, with each other through the functional links among them. The coordinate functional ways of molecules thus form a network in which the nodes are the molecules and the edges are the functional links between corresponding molecule pairs. A molecular network provides a comprehensive representation of the biological system, which allows us to investigate and understand the biological characteristics based on the network properties [13, 14]. On the basis of network theory, statistical modelling and machine learning methods are capable of detecting hidden patterns from the molecular network [15, 16, 17]. Different data sources and prior knowledge can be incorporated into molecular networks as additional attributes of nodes and edges to be investigated based on the functional context using proper models. Molecular networks therefore offer a flexible framework for multi-omic data representation and integration.

With the advances of high-throughput technologies and the reduce of genomic experimental costs, there will be a boost of generation of multi-omic data. The rapid growth of the amount of multi-omic data provides great opportunities for enhancing our understanding of molecular biological systems through data integration. However, it also brings big challenges for multi-omic data integration such as need of large computational resources, data heterogeneity of different types of omic data and inefficient computational models for the high-dimensional data structure. It is therefore necessary to develop more effective and efficient approaches for multi-omic data integration based on more powerful methods such as advanced machine learning techniques and network theory.

1.2 The hypothesis

The hypotheses for this thesis are as follows:

1. Complex biological systems can be represented by networks where nodes are molecules and edges are the functional links between molecules.

2. Multi-omic data can be integrated into molecular networks as attributes of nodes and edges.

3. Machine learning methods can be applied to recognise underlying patterns from molecular networks.

1.3 Research problems

On the basis of the above hypotheses, the main aim of this thesis is to develop network-based approaches for multi-omic data integration using machine learning methods. Specifically, we aim to provide solutions for four generalized biological problems by integrating corresponding multi-omic data, of which the former two are unsupervised learning problems and the latter two are supervised learning problems:

1. How to identify condition responsive gene functional modules based on transcriptome and interactome data?

2. How to identify translational regulated gene functional modules based on transcriptome, translatome and interactome data?

3. How to select feature genes for scalar responses based on transcriptome and interactome data?

4. How to classify patients based on epigenome and transcriptome data?

1.4 Contributions

The main task of multi-omic studies is to perform data mining through integration of multiple heterogeneous and high-dimensional omic data with respect to specific biological research problems. The biggest challenge for this task is how to use appropriate computational models to integrate different heterogeneous and high-dimensional omic data taking into account the relationships between the different levels of omic data. This thesis builds a bridge between computer science and biology by contributing to the multi-omics field in terms of methodologies. We develop efficient approaches using appropriate machine learning models for data mining through multi-omic data integration. These approaches are capable of transforming the multiple heterogeneous and high-dimensional omic data into underlying biological insights. Specifically, in this thesis, we propose four approaches aiming to address the aforementioned research problems, respectively, which incorporate principles from machine learning and network theory, as illustrated in Figure 1.1. Chapter 3 addresses the first research problem, in which we propose an approach for gene module detection by integrating multi-condition transcriptome data and interactome data using network overlapping module detection method. Chapter 4 addresses the second research problem, in which we propose an approach for gene module detection by integrating transcriptome, translatome, and interactome data using multilayer network. Chapter 5 addresses the third research problem, in which we propose an approach for feature selection by integrating transcriptome and interactome data using network-constrained regression.

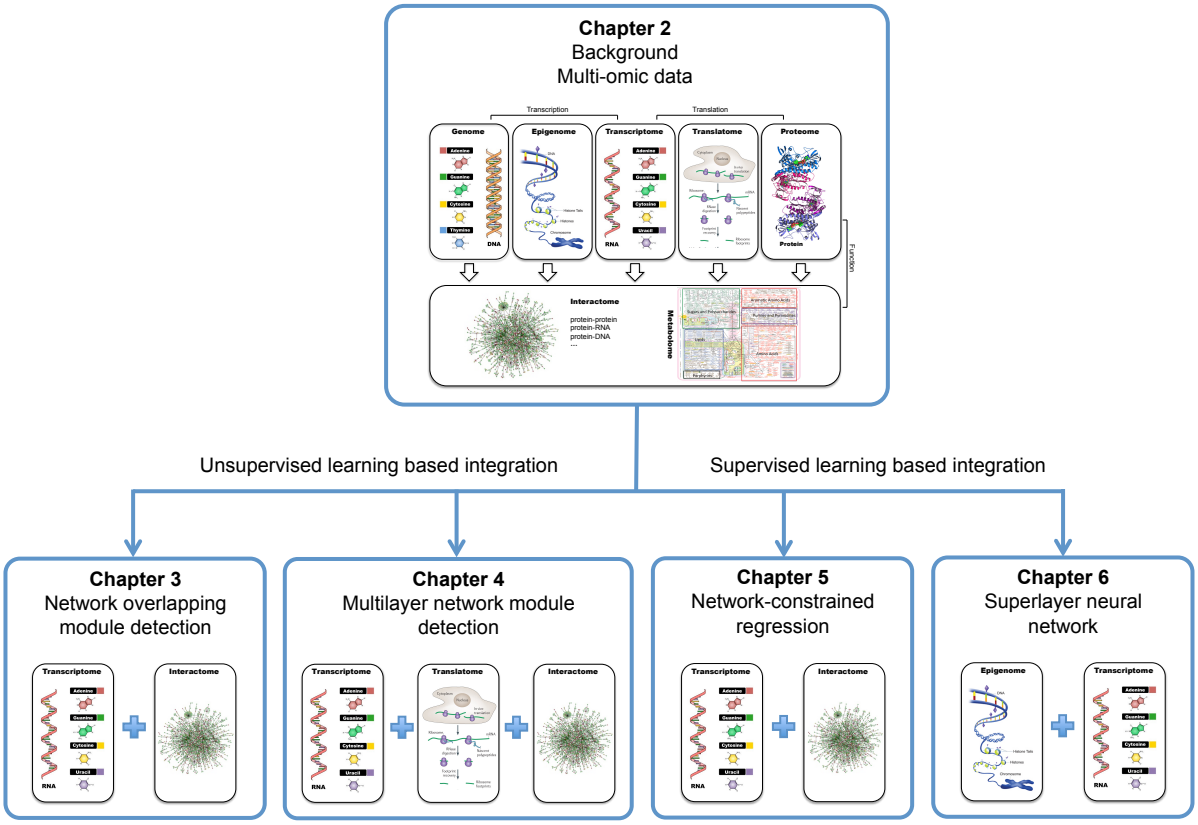


Figure 1.1: Multi-omic data integration approaches proposed in the thesis. Chapter 3 proposes an approach for gene module detection by integrating multi-condition transcriptome and interactome data using network overlapping module detection method. Chapter 4 proposes an approach for gene module detection by integrating transcriptome, translatome, and interactome data using multilayer network. Chapter 5 proposes an approach for feature selection by integrating transcriptome and interactome data using network-constrained regression. Chapter 6 proposes an approach for classification by integrating epigenome and transcriptome data using neural networks.

Chapter 6 addresses the fourth research problem, in which we propose an approach for classification by integrating epigenome and transcriptome data using neural networks.

Besides the contributions to the methodology side, this thesis also contributes to the biological discovery side. The research studies conducted in this thesis are involved in an EU collaborative project, Marie Curie Initial Training Network (ITN) EpiHealthNet, which consists of several research groups from different fields such as biology, statistics and computer science. EpiHealthNet aims to explore molecular mechanisms during mammalian early embryonic development in order to help to improve the health of human population through the interdisciplinary collaborations within the ITN. Various types of data, including omic data, are generated by the biological groups in EpiHealthNet, and my responsibility is to perform data mining on the omic data and transform them into the information that biologists need. Using the proposed approaches, we have successfully assisted our collaborators in transforming their in-house generated omic data into meaningful biological

insights through the integrative analysis with molecular networks. In Chapter 3 and 5, we present the works that are in collaboration with our partners in EpiHealthNet, Prof Daniel Brison from The University of Manchester, UK and Dr Giovanna Lazzari from Avantea, Italy. Applying the proposed approaches on the in-house generated transcriptome data of human early embryos provided by Daniel and Giovanna, we successfully discover several meaningful biological insights which help them to understand the underlying mechanisms related with human early embryonic development. Furthermore, besides the omic data provided by EpiHealthNet, in Chapter 4 and 6, we apply the proposed approaches on public human cancer omic data and mine interesting biological clues related with the mechanisms of tumorgenesis.

Publications (published, accepted, planned) related with this thesis are listed as follows:

1. Barandalla M, Shi H, Xiao H, Colleoni S, Galli C, Lio P, Trotter M, Lazzari G. Global gene expression profiling and senescence biomarker analysis of hESC exposed to H2O2 induced non-cytotoxic oxidative stress. *Stem Cell Res Ther*, 2017 Jul 5;8(1):160. (related with Chapter 3)

2. Helen Louise Smith, Adam Stevens, Ben Minogue, Sharon Sneddon, Lisa Shaw, Lucy Wood, Tope Adeniyi, Hui Xiao, Pietro Lio, Sue Kimber, Daniel Brison. Systems based analysis of human embryos and gene networks involved in cell lineage allocation. (*accepted by BMC Genomics*). (related with Chapter 3)

3. Hui Xiao, Pietro Lio. TERM: an R package for identification of translation efficiency regulated modules via network-based integration of Ribosome profiling data. (*in submission*). (related with Chapter 4)

4. Systematic study of associations between EmbryoScope time-lapse parameters and gene expression in human pre-implantation embryos. (*manuscript in preparation*). (related with Chapter 3 & 5)

5. Ioana Bica, Petar Velickovic, Hui Xiao and Pietro Lio. Multi-omics data integration using cross-modal neural networks. *The 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2018).* (related with Chapter 6)

6. Hui Xiao, Krzysztof Bartoszek, Pietro Lio. Multi-omic analysis of signalling factors in inflammatory comorbidities. *BMC Bioinformatics*, 2018 Nov 30;19(Suppl 15):439.

1.5 Thesis overview

The rest of the thesis is structured as follows:

Chapter 2 covers the background knowledge for the thesis, which provides introductions to multi-omic data, biological networks and common machine learning techniques.

Chapter 3 proposes an approach for gene module detection by integrating multi-condition transcriptome data and interactome data using network overlapping module detection method, which consists of four steps: (1) construction of gene co-expression network by evaluating co-expression correlation coefficient between each interacted gene pair based on their gene expression; (2) detection of overlapping gene modules from the co-expression network using network overlapping module detection method; (3) identification of condition-associated modules by assessing the significance of enrichment with condition-associated genes within the modules using ANOVA-GSEA; (4) selection of condition-specific feature modules and feature genes using GEL logistic regression with K-fold cross-validation. We apply the proposed approach on the transcriptome data of human pre-implantation embryos across multiple development stages and identify human embryonic development stage-specific modules and genes. Interesting biological insights are revealed from the dynamic expression patterns of the stage-specific modules and the multiple function genes located in the overlapping modules, which provides clues for understanding the potential molecular mechanisms during human pre-implantation embryonic development. To assess the stability of the modules identified by the proposed approach, we perform similar module detection studies using several common module detection methods as well as on different transcriptome data. We find that the intra-module genes are significantly overlapped between different methods and datasets.

Chapter 4 proposes an approach for gene module detection by integrating transcriptome, translatome, and interactome data using multilayer network, which consists of five steps: (1) construction of multilayer differential expression network by integrating transcriptome and translatome with interactome data respectively; (2) selection of seed genes for module detection by evaluating their degrees of differential translation; (3) detection of modules from the multilayer network using greedy search for each seed gene by minimizing the entropy-based local modularity function; (4) identification of translation efficiency (TE) regulated modules by the refinements including significance assessment, redundancy deletion and dynamic evaluation; (5) visualization of TE-regulated modules as graphs with incorporated multilayer information from the networks. We apply the proposed approach on a published ribosome profiling data of mTOR perturbed prostate cancer cells and mine several TE-regulated modules associated with mTOR perturbation. The translational regulated genes and modules downstream mTOR provide valuable clues for understanding the mTOR associated translational regulation mechanisms in prostate cancer genesis and metastasis. We develop an R package, TERM, for implementation of the proposed approach, which is capable of (1) evaluating differential translation of genes;

(2) identifying TE-regulated modules; (3) visualizing the TE-regulated modules. It is a useful tool for exploring translational regulation mechanisms by integrating transcriptome, tanslatome and interactome data.

Chapter 5 proposes an approach for feature selection by integrating transcriptome and interactome data using network-constrained regression. We develop a more efficient network-constrained linear regression method, named eGBL, by incorporating the edge weights into the GBL network-constrained penalty, which takes the advantage of weighted network. We evaluate the performance of eGBL on four simulated datasets built with different proportions of features, different magnitudes of coefficients and different signs of coefficients. We show that eGBL outperforms several common regularized regression methods and provides superior performance on feature selection. We apply eGBL to explore whether the key time-lapse parameters capable of predicting EmbryoScope blastocyst qualities are associated with transcriptional patterns. For each time-lapse parameter, we use eGBL on the transcriptome data of blastocysts to select the feature genes by fitting the linear model incorporating the human pre-implantation embryonic development network for regularization. We find scientific evidence that several selected feature genes play important roles across the stages of embryonic development. The early stage associated feature genes indicate the crucial roles of the key time-lapse parameters during the early pre-implantation embryonic development. The late stage associated feature genes account for the prediction capability of key time-lapse parameters on blastocyst qualities from the molecular level.

Chapter 6 proposes an approach for classification by integrating epigenome and transcriptome data using neural networks. We introduce two neural network models for DNA methylation and gene expression integration based on two strategies: (i) the multilayer perceptron (MLP) for series integration strategy, in which the DNA methylation and gene expression features are stacked together by samples; (ii) the superlayer neural network (SNN) for parallel integration strategy, in which the DNA methylation features and gene expression features are learned separately in superlayers but with cross-connections allowing the crosstalks between them. We train the optimal MLP and SNN on a breast cancer dataset using stratified nested 5-fold cross-validation and compare their performances on the cancer patients classification. The SNN provides superior performances and outperforms the MLP due to its capability of learning the intrinsic characteristics of the heterogeneous datasets. We compare the neuron activations between the layers before cross-connections and the ones after cross-connections in the SNN, and find that the cross-connections lead to a markedly improvement on discriminating the two classes of samples in the latter layer. We recommend the parallel integration strategy (i.e. the SNN) for the neural network based integration of DNA methylome and transcriptome data.

Chapter 7 concludes the thesis. We summarize the main contributions of the thesis and discuss future directions of the research field.

Chapter 2

Background

2.1 Multi-omics

2.1.1 Central dogma

The genetic information carried by DNA can be transmitted to offspring, which is the basis of the inheritance of phenotypic traits. Genes are the functional subunits of DNA which can encode other functional molecules such as RNAs and proteins. The flow of the genetic information from gene to gene products follows the "central dogma" of molecular biology [18] (Figure 2.1), in which DNA is firstly transcribed into RNA ("transcription") and then RNA is translated into protein ("translation"). The process, by which the information contained within a gene becomes a useful product (mRNA or protein), is called gene expression. The expression levels of genes are regulated by a very complicated system of mechanisms at both the transcription and translation levels. In biological systems, the different molecules such as DNA, RNA and protein do not work independently but in coordinate ways to fulfil some specific functions by interacting with each other through various biological reactions such as DNA-protein binding, RNA-protein binding, protein-protein physical interactions, metabolic reactions and so on. The molecules and the functional links among them which are involved in a biological process are considered as a biological pathway. From the perspective of system biology, biological systems are highly dynamic, which consist of multiple pathways and functional molecules.

2.1.2 Multi-omic data

The advent of advanced high-throughput technologies in molecular biology, such as microarray and next generation sequencing, provides opportunities to measure the whole genomes at different molecular levels in biological systems. Such genome-wide experiments of the molecular levels produce various omic data including genome, epigenome, transcriptome, translatome, proteome, metabolome, and interactome (Figure 2.2). Each type of omic

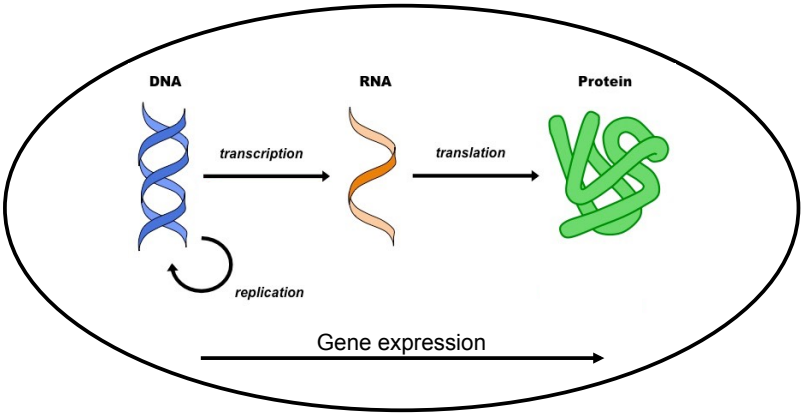


Figure 2.1: Central dogma of molecular biology.

data has its unique characteristics and provides a comprehensive view of functionality at the corresponding molecular level. In order to gain a global view of molecular mechanisms at the system level, it is critical not to understand the single omic data separately, but to integrate the multiple omic data. Multi-omics refers to the integrative study of the following multiple omic data.

Genome is the complete set of DNA, which holds all genetic information of an organism [19]. The term "-omics" refers to the study of corresponding omic data. Genomics is the study of the genome of an organism. The main aim of genomics is to determine the whole sequence of DNA and study its structure, function, evolution, and editing of the genome. The DNA is the fundamental knowledge of all other omic data. Therefore, genomics is of great importance not only in omics fields but also in other research fields such as medicine and biotechnology.

Epigenome is the complete set of the reversible chemical changes to the DNA and histone proteins of an organism [20], which is heritable to its offspring. Unlike the static genome, epigenome can be dynamically affected by environmental conditions [21]. Epigenetic changes can lead to alterations in chromatin structures, which will, in turn, affect the function of the genome. Two main types of epigenetic changes are DNA methylation and histone modification, which have been proved to play important roles in regulating gene expression [22]. Epigenomics focuses on genome-wide identification and characterization of epigenetic modifications.

Transcriptome is the complete set of RNA molecules in a cell or a collection of cells [23]. It usually refers to the total RNAs, but sometimes just the messenger RNAs (mRNAs), which depends on the experiment design. By keeping only the mRNAs, the transcriptome is an expression of the genome, which captures the expressed genes at

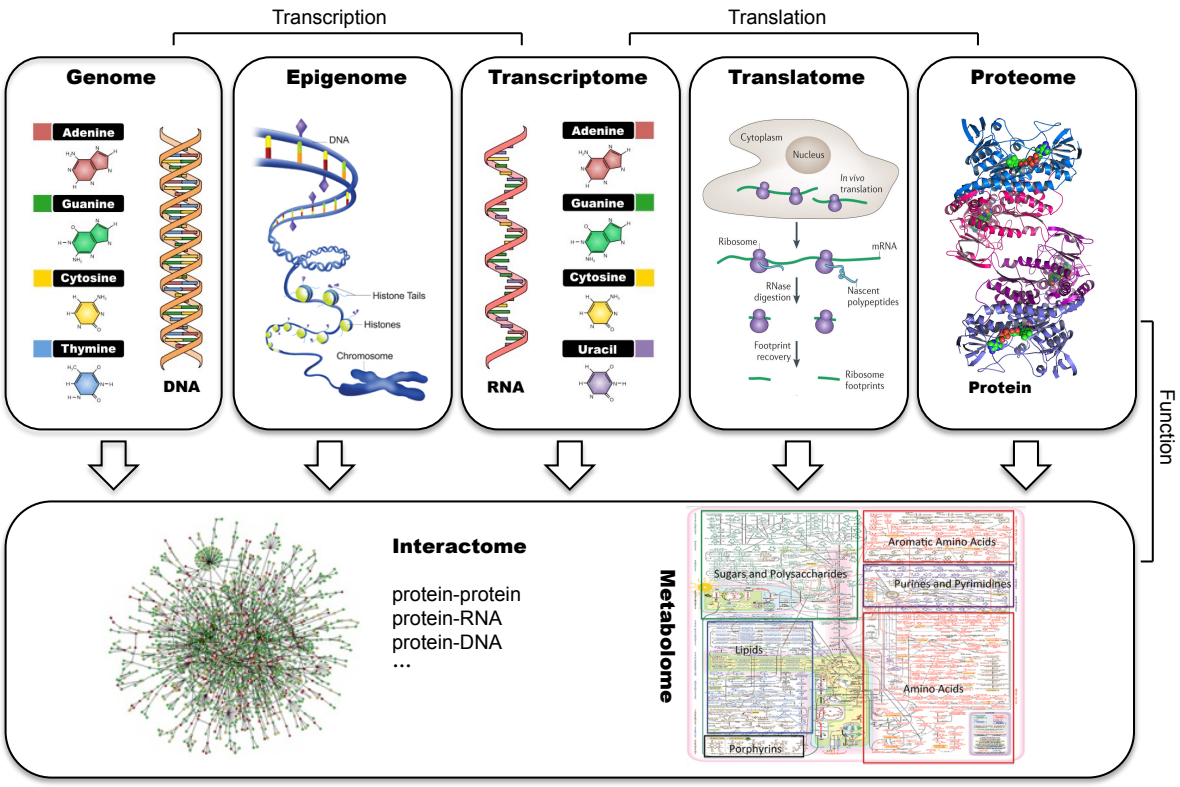


Figure 2.2: Overview of multi-omic data.

transcription level in the given condition. Transcriptome is a dynamic system which is responsive to external environmental conditions, where the gene expression changes with the alterations in conditions. The main aim of transcriptomics is to identify genes that exhibit significant difference in expression between different conditions (e.g. disease vs. normal, different tissues, multiple time points), usually referred to as differentially expressed genes (DEGs), and thus to infer the potential regulatory patterns which will provide clues for understanding the related mechanisms.

Translatome is the complete set of the mRNA fragments that are being translated in a condition in a single cell [24]. Translatome data can be obtained by using ribosome profiling techniques [25]. It measures the total ribosome protected mRNAs fragments (RPFs) as ribosome is the factory of protein synthetic in cells. Translatome is the intermediate layer between transcription and translation. Translatome data are not studied independently, but in combination with the matched transcriptome data. Translatomics aims to infer the discordance between the changes in transcriptome and the changes in translatome. The strong discordance will suggest the potential regulation mechanisms which control the expression from transcription to translation.

Proteome is the complete set of proteins expressed in a cell, tissue, or organism [26]. Similar with the transcriptome, the proteome is also an expression of the genome, but at the translation level, which captures the expressed proteins. The proteome also actively changes in response to the external environmental conditions. Besides the protein expressions, the proteomics also involves understanding the potential patterns that proteins function and interact with each other.

Metabolome is the complete set of all low molecular weight metabolites that are produced by cells during metabolism, and provides a direct functional readout of cellular activity and physiological status [27]. Compared to the above omic data, metabolome is not directly involved in the information flow of the central dogma. Metabolomics is an emerging discipline which aims to profile all small molecular metabolites present in an organism. It provides a tool for understanding how mechanistic biochemistry links to cellular phenotypes.

Interactome is the complete set of molecular interactions in a cell [28]. An interactome can be intuitively represented by a network, where nodes are molecules such as genes, RNAs and proteins, and edges indicate functional relationships between molecule pairs. The functional links are defined from different data sources, such as physical interactions from protein-protein interaction network, and protein-RNA binding from gene regulation network. The interactomics aims to discover the potential patterns in the molecular interactome from the network perspectives by studying the topological properties which usually suggest the potential biological roles. It is needed to be noted that the molecular interactome in a cell is a static network as it is a collection of molecular interactions under various conditions. But biological processes in cells are highly dynamic systems where the interactions turn on/off in response to the temporal and spatial changes in cells. Therefore, capturing the dynamic responsive patterns will help to understand the underlying mechanisms.

2.2 Networks in molecular biology

Networks are widely used in the field of molecular biology as an intuitive representation of a biological system, where nodes/vertices are molecules such as genes, RNAs and proteins, and edges indicate functional links between molecule pairs. Molecular networks provide different functional information of the biological system corresponding to different types of interactions between molecules.

- Gene co-expression networks are constructed by looking for pairs of genes which show similar expression patterns across biological conditions (e.g. disease vs. normal,

different tissues, multiple time points), where the activation levels of two co-expressed genes rise and fall together across conditions.

- Protein-protein interaction networks represent physical interactions between proteins such as building of a protein complex which is a group of multiple proteins stably interacting with each other and the activation of one protein by another protein.
- Metabolic networks show how metabolites are transformed, for example, to produce energy or to synthesize specific substances such as carbohydrates, glycans, proteins and nucleotides which are essential for biological systems.
- Signal transduction and gene regulatory networks describe how genes can be activated or repressed, and therefore contain information about which mRNAs or proteins are produced in a cell at a particular time.

In a biological system, the above networks crosstalk with each other and form a comprehensive complex molecular interaction network which provides the fundamental function context of molecular mechanisms.

2.2.1 Network properties

Molecular interaction network is presented in a graph with vertices/nodes referring to molecules and edges presenting interactions among molecules. A node can be characterized according to its topological properties in a network, which suggests that the topological characteristics may indicate its biological roles. The most widely used network properties for inferring associated biological patterns are described as follows:

2.2.1.1 Scale-free property

In a network, the degree of a node is the number of connections it has. The degree distribution is the probability distribution of the degrees over the whole network. It is reported that the degree distributions of most biological networks follow a power-law distribution where the degree k following $P(k) = ck^{-\gamma}$, with $2 < \gamma < 3$ [29]. Power-law networks are often known as scale-free networks since alterations in the constant c do not change the power-law exponent. One important characteristic of the scale-free network is that there are many nodes with few interactions and few nodes that have many interactions in the network. Random removal of a node in a scale-free network will not disturb its fundamental structure because the chance that a random failure would delete a highly connected node is very small and the removal of small degree nodes does not have a strong effect on integrity of network, which indicates the robustness of the scale-free network. Since molecular interaction networks represent a complex biological system, the scale-free

characteristic can also explain the stability of the biological system which is characterized by its capacity to recover to stable conditions or steady states after a random perturbation of its robustness (e.g. stimulations to the biological system or environment changes) [30].

2.2.1.2 Hubs

In a scale-free network, highly connected nodes are called hubs. Hubs have a significantly larger number of links in comparison with other nodes in the network, which suggests that hubs play central roles in the network locally or globally. Hubs are responsible for exceptional robustness of network [31]. Removal of hubs will result in destruction of the network. Because small nodes are predominantly linked to hubs, the integrity of the network will fall apart relatively fast by the removal of hubs especially the largest ones.

The topological importance of hubs in molecular interaction network will suggest their crucial roles in biological processes, e.g., essential genes are likely to be hubs in molecular networks [13], as well as the disease genes [14]. Dysfunction of the hub molecules will lead to a strong interruption for the biological system which might result in death or disease genesis. A large number of efforts have been done in studying hub genes/proteins to discover new biomarkers or drug targets.

2.2.1.3 Cliques and modules

A clique, is a graph or subgraph in which every node is connected to every other nodes, e.g., a clique of size three corresponds to a triangle. A maximal clique is a clique that cannot be further enlarged given its neighbours in the network. Large cliques in a sparse network might be a potential sign of modules existing in the network. A module, also called a community or a cluster, is usually defined as a tightly connected subnetwork in a network, which is densely connected by internal edges but loosely connected to the outside nodes. The emergency of modules suggests an important characteristic of scale-free network which is defined as modularity.

In biological systems, molecules do not work independently but in a coordinate way by crosstalking to each other, and a set of molecules that are involved in the same biological process are likely to form a coherent functional module. Consequently, a functional module can be defined as a group of genes or their products which are related by one or more functional interactions such as co-regulation/co-expression/co-occurrence in a protein complex/a metabolic or signalling pathway. An important property of a module is that its function is separable from other modules [32] and that its members have more relations among themselves than with members of other modules, which is reflected in the network topology as a tightly connected network module [33].

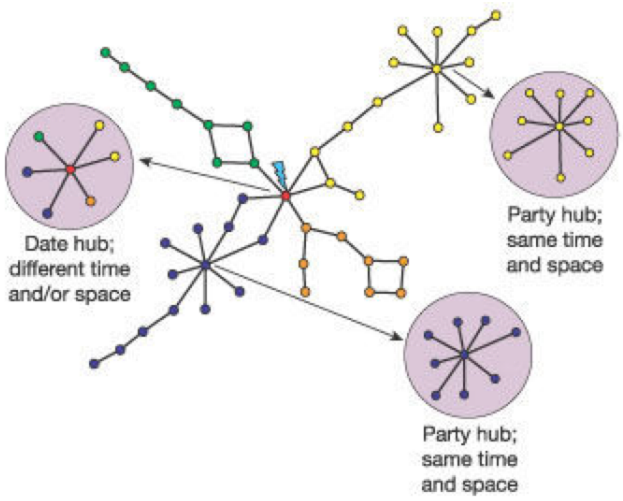


Figure 2.3: Date and party hubs illustrated by Han et al. [34].

2.2.2 Dynamic network

According to the biological roles of network properties, a biological system can be considered as a functionally organised modular network consisting of functional modules and single molecules. Biological systems are not static processes but dynamic procedures where molecules are responsive to different temporal or spatial environments, e.g. many genes are multi-functional that they involve in different biological pathways corresponding to different conditions, and different biological pathways are responsive in different biological systems. Therefore, the modular network can be considered as a dynamic network organised by dynamic crosstalks among molecules and modules corresponding to different conditions.

Han et al. [34] discovered two types of hubs in protein-protein interaction network according to their dynamic co-expression patterns with their partners (as shown in Figure 2.3): party hubs which interact with most of their partners simultaneously and date hubs which interact with their different partners at different times or locations. From the network perspectives, party hubs are likely to be hubs located within modules, while date hubs are usually the ones located between modules. On the basis of such hypothesis, the modularity of a molecular interaction network can be considered as dynamic re-wiring between date hubs and modules responsive to different conditions. The dynamic patterns can be captured by combining the molecular interaction network with different omics data, e.g. transcriptome.

Dynamic molecular interaction network provides us a flexible framework to integrate omic data with molecular network to identify responsive molecular patterns by using data mining technologies such as statistical learning and machine learning.

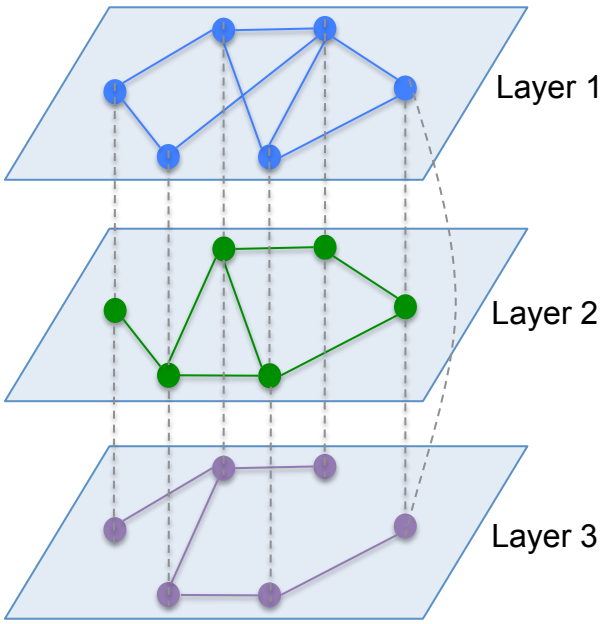


Figure 2.4: Illustration of a simple multilayer network. The multilayer network consists of three layers with nodes in blue, green and purple respectively. The solid lines represent the within-layer links and the dashed lines represent the between-layer links.

2.2.3 Multilayer network

Multilayer network is an emerging domain in the field of network science [35, 36]. A multilayer network consists of several layers of network, which include the same nodes in each layer network, illustrated in Figure 2.4. The networks are connected by both within-layer links (links within a same layer) and between-layer links (links between different layers). The multilayer network has advantages in modelling complex systems as it is capable of describing the properties of a specific aspect by the within-layer links as well as capturing the complexities between different aspects by the between-layer links.

Multilayer network has been introduced in molecular biology to understand the mechanisms from different molecular levels in an integrative way instead of aggregating them into a single network [37, 38], which helps to uncover new knowledge that are ignored in the aggregated network. Multilayer network also provides a flexible and efficient framework for multi-omic data integration, as in a multilayer framework, each oimc data can be described by a layer of network and the correlations among different omic data can be captured by the between-layer links. Recently, many efforts have been made for multi-omic data integration based on multilayer network, e.g., inference of epigenetic functional modules from a multiple layer networks constructed based on gene expression and DNA methylation data [39], and identification of cancer driver genes via community detection from multilayer networks built by integrating multi-omic data [40].

2.3 Machine learning

2.3.1 Regularized linear regression

Regularized linear regression is a supervised machine learning technique, which fits a linear model using regularization techniques to penalize the coefficients of the linear model. This section provides the background knowledge of linear regression, regularized/penalized regression and network-constrained regression.

2.3.1.1 Linear regression

Linear regression is a statistical model developed for understanding the relationship between a dependent scalar variable and a number of independent variables, which has been widely used in machine learning for supervised learning to predict a quantitative response variable from the predictor variables.

Let $y = (y_1, ..., y_n)^T$ be a quantitative response variable which contains a vector of n observations. Let $x_i = (x_{i1}, ..., x_{ip})$ be the vector of p predictor variables corresponding to the observed response y_i, where $i = 1, ..., n$. According to the assumption that the relationship between response variable and predictor variables is linear, it is modelled as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \epsilon_i = \beta_0 + x_i\beta + \epsilon_i, \quad i = 1, ..., n \tag{2.1}$$

where $\beta = (\beta_1, ..., \beta_p)^T$. $\beta_1, ..., \beta_p$ are the regression coefficients corresponding to the p predictor variables, and β_0 is the intercept of the linear model. ϵ_i is an unobserved random variable of the systematic error for the linear model between y_i and x_i, which cannot be predicted or reduced. The equations of n observations are stacked together and the linear model can be written as:

$$y = \beta_0 + X\beta + \epsilon \tag{2.2}$$

where $X = (x_1, ..., x_n)^T$ and $\epsilon = (\epsilon_1, ..., \epsilon_n)^T$.

Statistical estimation and inference in linear regression model focuses on β. A regression coefficient β_j can be interpreted as the partial derivatives of the response variable with respect to the corresponding predictor variable. It can be negative or positive, which assesses the degree of change in the response variable for every 1-unit of change in the predictor variable. If β_j is positive, the interpretation is that for every 1-unit increase in the predictor variable, the response variable will increase by the value of β_j. If β_j is negative, the interpretation is that for every 1-unit increase in the predictor variable, the response variable will decrease by the absolute value of β_j.

2.3.1.2 Ordinary least squares estimation

The main task of linear regression is to estimate and infer the coefficients β of the linear model. A large number of methods have been developed to solve this problem. Ordinary least squares (OLS) [41] is one of the most basic and common solution to estimate and infer β for a linear model.

OLS aims to estimate the unknown coefficients in a linear model by minimizing the loss function $L(\beta)$ defined by the sum of the squared residuals (SSR) which is the differences between the observed response variables y in the given dataset and those $\hat{y}$ predicted by the linear model. Suppose β is a candidate vector of values for the coefficients. The residual for the ith observation is quantified as $(y_i - \hat{y_i})$, which assesses the distance of fit between the actual data and the model. The sum of squared residuals (SSR) given β, $S(\beta)$, is a measure which evaluates the overall model fit:

$$S(\beta) = \sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \sum_{i=1}^{n}(y_i - x_i\beta)^2 = (y - X\beta)^T(y - X\beta) \tag{2.3}$$

Thus, the loss function $L(\beta)$ is defined as $L(\beta) = S(\beta)$. The solution for coefficient estimation for the linear model is to find the OLS estimator $\hat{\beta}_{OLS}$ for β which minimizes $L(\beta)$, that is, $\hat{\beta}_{OLS} = \arg\min L(\beta)$. Since $L(\beta)$ is a quadratic function of β, the vector $\hat{\beta}_{OLS}$ which gives the global minimum can be found via matrix calculus by differentiating with respect to the vector β and setting equal to zero:

$$\begin{aligned} 0 = \frac{dL(\beta)}{d\beta} &= \frac{d}{d\beta}\left(y^Ty - \beta^TX^Ty - y^TX\beta + \beta^TX^TX\beta\right)\Big|_{\beta=\hat{\beta}_{OLS}} \\ &= -2X^Ty + 2X^TX\hat{\beta}_{OLS} \end{aligned} \tag{2.4}$$

By assumption that matrix X has full column rank, X^TX is invertible. Thus, the OLS estimator $\hat{\beta}_{OLS}$ for β is given by [42]:

$$\hat{\beta}_{OLS} = \arg\min_{\beta} L(\beta) = (X^TX)^{-1}X^Ty \tag{2.5}$$

2.3.1.3 Bias-variance trade-off

In order to verify whether the OLS estimator $\hat{\beta}$ have the optimum values, we can check if the predicted response value $x_i\hat{\beta}$ is close to the actual value $x_i\beta$ and how well the linear model fits other independent datasets (e.g., future observations).

The mean squared error (MSE) is used to evaluate the closeness between the estimated $\hat{\beta}$ and the true β, which calculates the squared distance of $\hat{\beta}$ to β:

$$MSE(\hat{\beta}) = \mathrm{E}[||\hat{\beta} - \beta||^2] = \mathrm{E}[(\hat{\beta} - \beta)^T(\hat{\beta} - \beta)] \tag{2.6}$$

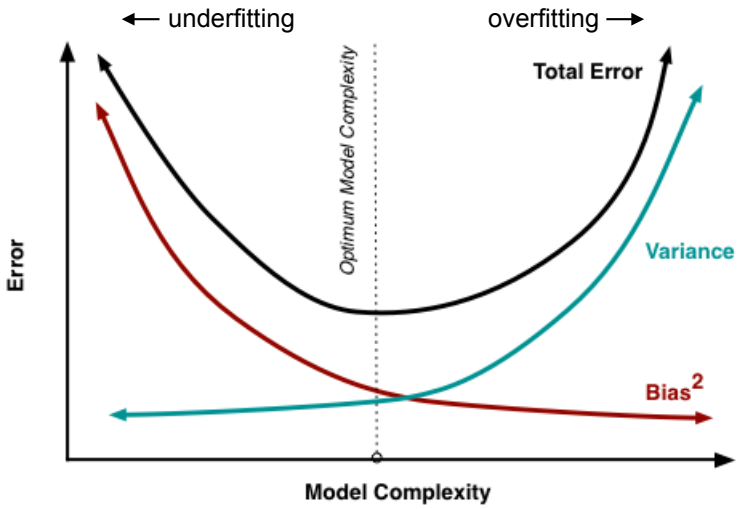


Figure 2.5: Bias-variance trade-off. This figure is adapted from [43].

where E is the true expectation.

The predicted error is calculated to evaluate how well the estimated model fits on future observations, which can be explained as the difference between the predicted $\hat{y}$ and the actual y from the new dataset. The expected squared predicted error is calculated as follows:

$$ERR = \mathrm{E}[(y - \hat{y})^2] = (\mathrm{E}[\hat{y}] - y)^2 + \mathrm{E}[(\hat{y} - \mathrm{E}[\hat{y}])^2] + \epsilon \tag{2.7}$$

The ERR can be deposed as: $ERR = Bias^2 + Variance + IrreducibleError$. The bias is an error from erroneous assumption in the learning algorithm, which simply means how far away is estimated values from actual values. The variance is an error from sensitivity to small fluctuations in the training set which is a measure of variations in the predicted values. The irreproducible error is the inherent uncertainty around the mean, which cannot be predicted or reduced. Therefore, the predicted error is affected by the bias and variance of the model. Dealing with these two components will help in reducing MSE which in turn will reduce the predicted error of the estimation.

The decomposition illustrates a trade-off between bias and variance of the model. Figure 2.5 illustrates how bias and variance change as the complexity (number of predictors) of the model increases. As the complexity increases, variance increases and bias decreases. Linear regression exhibiting low variance and high bias in a model will result in underfitting of the data, which means that the model is unable to find the underlying patterns within the dataset. By contrast, a model with high variance but low bias will result in overfitting, which suggests the model captures not only the underlying patterns but also the noises and outliers in the dataset. In order to obtain a perfect model, we need to find an optimum point balancing the trade-off between bias and variance to improve the generalization capability of the model, which is shown as marked by the dotted line in Figure 2.5.

2.3.1.4 Regularized linear regression

In practice, the big data (e.g., omic data) studied by machine learning are usually high-dimensional data, which means in the dataset, the number of features (predictors) greatly exceeds the number of observations (samples). OLS estimation on such dataset will result in overfitting of the linear model. Moreover, in OLS estimation, β is estimated without any constrain, and therefore, the values of β can explode which is susceptible to a very high variance.

As shown in Figure 2.5, to overcome overfitting of a model, the goal is to find the optimum point for bias-variance trad-off by moving the trade-off line more towards left-hand side. In this case, a small increase in bias can result in a big decrease in variance, which will result in a substantial decrease in predicted error.

One of the most common methods to avoid overfitting is to reduce the model complexity using regularization, which introduces a penalty term to the coefficients β constraining their magnitudes in OLS estimation. The objective of regularized regression is to solve:

$$\min_{\beta} S(\beta) \text{ subject to } P(\beta) \leq t, \tag{2.8}$$

where $S(\beta)$ is the OLS loss function which is given by Equation 2.3, and $P(\beta)$ denotes the penalty term added for regularization. t is a predefined free parameter which determines the magnitude of regularization. Consequently, the loss function for the regularized regression becomes $L(\beta) = S(\beta) + P(\beta)$ and the estimator $\hat{\beta}$ is given by:

$$\hat{\beta} = \arg\min_{\beta} L(\beta) = \arg\min_{\beta}(S(\beta) + P(\beta)) \tag{2.9}$$

The most commonly used regularization technique controlling the magnitude of a numeric vector is called L_p-norm, which is defined as follows:

$$||x||_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p} \tag{2.10}$$

If $p = 1$, L_1-norm is defined as the sum of absolute value of the vector. If $p = 2$, L_2-norm is defined as the root of sum of squares. When $p \to \infty$, L_∞-norm approaches the infinity norm which is defined as the maximal absolute value of the vector. L_1-norm and L_2-norm are widely used for regularized regression. L_2-norm offers a smooth solution for coefficient estimation by shrinking the magnitudes of the coefficients, and L_1-norm offers a sparse solution by shrinking the coefficients and enforcing the irrelevant coefficients to 0.

A large number of regularized regression methods have been proposed by assigning different penalties to coefficients to overcome the overfitting of linear models in high-dimensional data, and several most common methods are introduced as follows:

Ridge Ridge regression [44] is a regularized linear regression method with the L_2-norm based regularization. The penalty function in Ridge regression is defined as:

$$P(\beta) = \lambda||\beta||_2^2 = \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.11}$$

where λ is the shrinkage parameter which controls the magnitudes of coefficients β. The overall fit of the Ridge regression model can be evaluated by the following loss function:

$$\begin{aligned} L(\beta) &= \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \\ &= (y - X\beta)^T (y - X\beta) + \lambda||\beta||_2^2 \end{aligned} \tag{2.12}$$

Thus, the Ridge regression estimator $\hat{\beta}_{Ridge}$ is given by $\hat{\beta}_{Ridge} = \arg\min_{\beta} L(\beta)$.

Ridge regression aims to avoid overfitting by penalizing large values of coefficients β. The L_2-norm based penalty introduces the smooth solution by shrinking large values of β in order to reduce the mean squared error and the predicted error of the linear model. The penalty function $P(\beta)$ is applied to the coefficients $\beta_1, ..., \beta_p$ but not to the intercept β_0 as β_0 is simply a constant of the mean value of the response variable. λ is a tuning parameter controlling the amount of shrinkage of the values of β, which will always be greater than 0. When $\lambda = 0$, Ridge regression is equivalent to standard linear regression and the coefficient estimation procedure is the same with OLS estimation. The higher value of λ, the greater shrinkage will be performed on the values of β.

One important property of Ridge regression is that it shrinks the coefficients of less important predictors approaching 0 but not becoming 0, which means that Ridge regression cannot filter out irrelevant predictors by enforcing their coefficients equal to 0 but reduce their impacts in the model. Consequently, Ridge regression improves prediction error by parameter shrinkage to reduce overfitting, but does not perform variable selection.

Lasso Lasso [45], standing for Least Absolute Shrinkage and Selection Operator, is a regularized regression method with the L_1-norm for regularization. The penalty function in Lasso regression is defined as:

$$P(\beta) = \lambda||\beta||_1 = \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.13}$$

where λ is the shrinkage parameter similar with the λ parameter in Ridge regression. The overall fit of the Lasso regression model can be evaluated by the following loss function:

$$\begin{aligned} L(\beta) &= \sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= (y - X\beta)^T(y - X\beta) + \lambda||\beta||_1 \end{aligned} \tag{2.14}$$

Thus, the Lasso regression estimator $\hat{\beta}_{Lasso}$ is given by $\hat{\beta}_{Lasso} = \arg\min_{\beta} L(\beta)$.

Lasso regression performs the L_1-norm regularization, which offers the sparse solution with both regularization and selection for coefficient estimation. Similar as Ridge regression, Lasso regression also shrinks the values of coefficients according to the impacts of the predictors. But the key difference with Ridge regression is that Lasso regression enforces the coefficients of irrelevant predictors to 0 and removes them from the linear model. The remaining non-zero predictors are therefore selected as the features for the model, which achieves the goal of variable selection.

Although Lasso regression is capable of feature selection, there are still some shortcomings with the practical applications on the high-dimensional omic data. When the number of predictor variables p is greater than the number of observations n (i.e., $p > n$), Lasso regression can select at most n features, even if there might be more than n associated features. Moreover, for a group of highly correlated variables, Lasso tends to select only one variable from the group and ignore the others, which might miss some important features in the data.

ElasticNet In high-dimensional datasets, especially when $p \gg n$, some groups of predictor variables tend to be strongly correlated among themselves within the same group, which refers to group effect. Taking such group effect into account in regression, a group of highly correlated predictors are supposed to have similar regression coefficients. But Lasso tends to select only one variable from the group and ignore the others. To overcome such limitations, Zou and Hastie introduced the elastic net [46], referred to as ElasticNet hereafter, which adds the Ridge penalty to Lasso. The penalty function in ElasticNet regression is defined as a combination of L_1-norm and L_2-norm:

$$P(\beta) = \lambda_1||\beta||_1 + \lambda_2||\beta||_2^2 = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \tag{2.15}$$

where λ_1 and λ_2 are the shrinkage parameters for Lasso regularization and Ridge regularization, respectively. Let $\alpha = \lambda_1/(\lambda_1 + \lambda_2)$, the penalty function is thus equivalent to:

$$P(\beta) = (1-\alpha)||\beta||_1 + \alpha||\beta||_2^2 = (1-\alpha)\sum_{j=1}^{p}|\beta_j| + \alpha\sum_{j=1}^{p}\beta_j^2 \tag{2.16}$$

α is a tuning parameter ranging from 0 to 1, which controls the trade-off between Lasso regularization and Ridge regularization. When $\alpha = 0$, ElasticNet regression is equivalent to Lasso regression. When $\alpha = 1$, it becomes Ridge regression. The overall fit of the ElasticNet regression model can be evaluated by the following loss function:

$$\begin{aligned} L(\beta) &= \sum_{i=1}^{n}(y_i - x_i\beta)^2 + (1-\alpha)\sum_{j=1}^{p}|\beta_j| + \alpha\sum_{j=1}^{p}\beta_j^2 \\ &= (y - X\beta)^T(y - X\beta) + (1-\alpha)||\beta||_1 + \alpha||\beta||_2^2 \end{aligned} \tag{2.17}$$

Thus, the ElasticNet regression estimator $\hat{\beta}_{ElasticNet}$ is given by $\hat{\beta}_{ElasticNet} = \arg\min_{\beta} L(\beta)$.

In ElasticNet penalty, the combination of L_1-norm and L_2-norm offers advantages of both Lasso and Ridge. ElasticNet regression introduces both sparse solution and smooth solution for coefficient estimation. It is capable of enforcing coefficient sparsity and overcomes the limitation on the number of selected variables. More importantly, ElasticNet regression performs variable selection encouraging the grouping effect. For a set of strongly correlated independent variables in the dataset, ElasticNet regression simply forms them into a group and shrinks their coefficients into similar levels. The entire group tends to be selected into or out of the model together. ElasticNet regression is particularly useful for high-dimensional data when the number of predictors is much bigger than the number of observations.

Group Lasso Ridge, Lasso and ElasticNet regression perform regularizations to estimate coefficients for individual predictor variable. ElasticNet regression has taken group effect into account for variable selection, but the highly correlated predictors are formed into groups automatically when estimating coefficients. However, in many regression problems, predictors are not distinct but arise from common underlying patterns, such as the predefined groups. In these cases, the goal focuses on selecting important groups and estimating their effects. The aforementioned regression methods can still be used for such cases by considering each predictor individually, but they are inefficient to deal with the prior grouping structure of predictor variables. Ignoring the existing structures of variables might result in biased or insufficient results for feature selection. To address these limitations, Yuan and Lin [47] proposed the group lasso regression method in order to allow predefined groups of variables to be selected into or out of a model together, so that all the members of a particular group are either included or not included.

Suppose that the p predictor variables are divided into L groups. Let p_l be the number

of predictors in group l, where $\sum_{l=1}^{L} p_l = p$. Let X_l be the matrix of the predictors in group l with the corresponding coefficient vector β_l. Group Lasso incorporates the grouping structure into the regularization and the estimator $\hat{\beta}_{gLasso}$ is given by minimizing the loss function:

$$\hat{\beta}_{gLasso} = \arg\min_{\beta} L(\beta) = \arg\min_{\beta} \left(||y - \sum_{l=1}^{L} X_l \beta_l||_2^2 + \lambda \sum_{l=1}^{L} \sqrt{p_l} ||\beta_l||_2 \right) \tag{2.18}$$

where λ is the shrinkage parameter, and $||\beta_l||_2$ denotes the L_2-norm of the coefficients of group l.

Group Lasso performs regularization like Lasso at the group level, which shrinks the coefficients of the predictors within a group to a same value and enforces the coefficients for irrelevant groups to 0 by tuning the parameter λ. The groups with non-zero coefficients are considered to be associated with the response variable, and all the predictors corresponding to each non-zero group are selected into the model. Therefore, Group Lasso performs variable selection at group level. If the size of each group is one, Group Lasso is equivalent to standard Lasso.

Sparse-Group Lasso Group Lasso produces sparsity of coefficients at group level, but does not yield sparsity within a group as it treats all the predictors within the group equally. However, in practice, the predictors within a same group have different impacts on the response variable. It is therefore meaningful to select both groups and the important predictors within the groups. To achieve this goal, Simon et al. [48] proposed a regularized regression method, named Sparse-Group Lasso, which offers the bi-level sparse solution resulting in both the group-wise sparsity and the within-group sparsity, by combining Lasso penalty to Group Lasso penalty. The estimator of Sparse-Group Lasso $\hat{\beta}_{sgLasso}$ is given by:

$$\hat{\beta}_{sgLasso} = \arg\min_{\beta} \left(||y - \sum_{l=1}^{L} X_l \beta_l||_2^2 + (1-\alpha)\lambda \sum_{l=1}^{L} \sqrt{p_l} ||\beta_l||_2 + \alpha\lambda ||\beta||_1 \right) \tag{2.19}$$

where $\beta = (\beta_1, ..., \beta_p)$ is the coefficients vector of all predictors in the dataset, β_l is the coefficients vector of predictors belonging to a specific group l, λ is the shrinkage parameter and α is the tuning parameter which controls the trade-off between Lasso penalty and Group Lasso penalty. When $\alpha = 0$ Sparse-Group Lasso is equivalent to Group Lasso regression and when $\alpha = 1$, it becomes standard Lasso regression. On the basis of the bi-level sparsity produced by Sparse-Group Lasso, the coefficients of all predictors in the irrelevant groups are enforced to 0 as well as the irrelevant predictors in the relevant groups. The remaining non-zero predictors in the relevant groups are selected as the feature predictors in the feature groups.

Group Exponential Lasso (GEL) Breheny proposed a more efficient solution, Group Exponential Lasso (GEL) [49], for producing bi-level sparsity, which is capable of group selection as well as important predictor selection from feature groups. The GEL penalty is defined based on an exponential function as follows:

$$P(\beta) = \sum_{l=1}^{L} \left(\frac{\lambda^2}{\theta} \left\{ 1 - \exp\left(-\frac{\theta ||\beta_l||_1}{\lambda} \right) \right\} \right) \tag{2.20}$$

where λ is the regularization parameter and θ is the tuning parameter. The GEL penalty allows the penalization on a predictor in a group decay exponentially as the importance of the group grows, and the rate of such decay is controlled by parameter θ. It is reported that the diminishing rate of penalization will lead to nearly unbiased estimator $\hat{\beta}$ given a large enough sample size [50], which accounts for the outperformance of GEL in comparison with the classic regularized methods such as Lasso that introduces significant bias toward zero for large numbers of regression coefficients. Consequently, GEL will be well suited for feature selection in high-dimensional dataset in the case $p \gg n$.

2.3.1.5 Network-constrained regression

The aforementioned regularized regression methods deal with the problem of variable selection in high-dimensional data with an important assumption that the predictor variables in the dataset are independent among each other. Although ElasticNet and Group Lasso based methods have taken into account the group effects such as inherent correlations and pre-defined grouping information among the predictors, they still treat each predictor individually in the group. However, for most cases of high-dimensional data in practice, the predictors are not independent but correlated to each other. Such correlation structures can be defined based on the prior knowledge from different aspects. For instance, different types of molecular interaction networks provide various functionally correlated structure among the genes in omics data such as co-expression correlation, co-regulation correlation and physical interaction in proteins (see details in Chapter 2.2). A generalization structure among predictors can be intuitively given by a network, where the nodes represent predictors and the edges denote the relationships between correlated predictor pairs that pre-defined from the prior knowledge. Incorporating the information from these networks into linear regression is of great importance, which will capture underlying patterns in the dataset by feature selection. To address this problem, a specific type of regularized regression framework, network-constrained regression, has been proposed for fitting linear models and achieving variable selection by incorporating the network structure as the constraint for regularization. Network-constrained regression methods are developed based on two prior assumptions:

- The hub predictors in the network are supposed to have larger coefficients than the lower degree predictors. On the basis of network properties, the topological importance of hubs usually indicates their critical roles in practice, which suggests the stronger associations between the hub predictors and the responsive variable. Highly connected predictors in the network, therefore, tend to have larger coefficients.

- Two predictors that are linked in the network are supposed to have similar degree-scaled coefficients. The edge between the two predictors indicates a correlated relationship between them, which suggests that they both have impacts on the response variable. Their impacts should be similar because the two linked predictors could be considered as a group following the pre-defined group structure. But according to the first assumption, the impact of a predictor is also proportional to its degree in the network. Therefore, the degree-scaled coefficients for the two linked predictors should be similar.

Recently, several network-constrained regression methods have been proposed to incorporate the graph information into regularization for fitting linear models. In this section, we introduce some commonly used network-constrained regression methods.

On the basis of the linear model defined in Chapter 2.3.1.1, lets introduce a weighted graph $G = (V, E, W)$ to represent the network, where V is the set of vertices/nodes corresponding to the p predictor variables, $E = i \sim j$ is the set of edges denoting that the correlated predictors i and j are linked in the network, and W is the weights of the edges, where $w(i, j)$ denotes the weight for the corresponding edge $e = (i \sim j)$. The edge weight can be considered as a measure which evaluates the strength of the correlated relationship between two predictors. Let $d_i = \sum_{i \sim j} w(i, j)$ be the degree of vertex i. If i is an isolated vertex in the network, let $d_i = 0$.

Grace Grace, standing for Graph Constrained Estimation, is the first network-constrained regression model proposed by Li et al. [51]. It performs a network-constrained regularization by combining a graph-based penalty with the Lasso penalty, which is defined as follows:

$$P(\beta) = \lambda_1 \sum_{k=1}^{p} |\beta_k| + \lambda_2 \sum_{i \sim j} \left(\frac{\beta_i}{\sqrt{d_i}} - \frac{\beta_j}{\sqrt{d_j}} \right)^2 w(i, j) \tag{2.21}$$

where the first term is Lasso penalty inducing the sparse solution and the second term induces the smooth solution, which is similar as ElasticNet that combines Lasso and Ridge penalties to achieve both sparsity and smoothness for coefficients. Let $\alpha = \lambda_2/(\lambda_1 + \lambda_2)$, then $P(\beta)$ can be written in the generalization style as EalsticNet:

$$P(\beta) = (1 - \alpha) \sum_{k=1}^{p} |\beta_k| + \alpha \sum_{i \sim j} \left(\frac{\beta_i}{\sqrt{d_i}} - \frac{\beta_j}{\sqrt{d_j}} \right)^2 w(i, j) \tag{2.22}$$

where $\alpha \in [0, 1)$ is a tuning parameter which controls the trade-off between the Lasso penalty and the graph-based penalty. When $\alpha = 0$, Grace becomes the standard Lasso. The network-constrained estimator $\hat{\beta}_{Grace}$ is derived by minimizing the loss function:

$$\begin{aligned}\hat{\beta}_{Grace} = \arg\min_{\beta} \quad & (y - X\beta)^T (y - X\beta) \\ & + (1 - \alpha) \sum_{k=1}^{p} |\beta_k| + \alpha \sum_{i \sim j} \left(\frac{\beta_i}{\sqrt{d_i}} - \frac{\beta_j}{\sqrt{d_j}} \right)^2 w(i, j)\end{aligned} \tag{2.23}$$

Grace performs a two-item ElasticNet-like penalty taking network structures into account. The first item produces the sparsity for coefficients which enforces the coefficients of irrelevant predictors to 0 in order to achieve the goal of variable selection. The second penalty term provides smooth solution motivated by the two prior assumptions. It shrinks the coefficients over the network by penalizing the weighted sum of squares of the difference of the scaled coefficients between the linked vertices in the network. The coefficients are scaled by the degrees of the corresponding vertices ($\beta_i/\sqrt{d_i}$) in the regularization, which allows the vertices with larger degrees (e.g. hubs) in the network to have larger coefficients. For a pair of linked predictors in the network, this penalty item cannot shrink their coefficient to similar magnitudes. But by introducing the weight factor $w(i, j)$ into the penalty, it is capable to reduce the difference between the two coefficients according to the weight of the corresponding edge. Thus, Grace can enforce the similar scaled coefficients for two highly correlated predictors (i.e., $\beta_i/\sqrt{d_i} \approx \beta_j/\sqrt{d_j}$).

Although Grace provides solution for enforcing $\beta_i/\sqrt{d_i} \approx \beta_j/\sqrt{d_j}$, it might fail when β_i and β_j have opposite signs which will be reasonable in practice, e.g., the opposite signs of two linked nodes/genes in gene co-expression networks usually suggest the potential regulation between the two genes.

GBL Pan et al. [52] proposed an alternative but more efficient penalty for network-constrained regression, which performs a form of grouped penalty for each edge in the network. We refer to this method as GBL hereafter because the authors suggested the implementation with modified generalized Boosted Lasso (GBL) algorithm [53]. The penalty function is defined as follows:

$$P(\beta) = \lambda \sum_{i \sim j} 2^{1/\gamma'} \left(\frac{|\beta_i|^\gamma}{w(i)} + \frac{|\beta_j|^\gamma}{w(j)} \right)^{1/\gamma} \tag{2.24}$$

where parameters $\lambda > 0$, $\gamma > 1$ and γ' satisfies $(1/\gamma) + (1/\gamma') = 1$. $w(i)$ denotes a weight function which is used to scale the coefficient of each predictor. The authors proposed three types of weight function $w(i)$ for vertex i based on its degree in the network: $w(i) = d_i^{(\gamma+1)/2}$, d_i, or d_i^γ.

The group-like penalty of GBL shrinks the coefficients on each edge over the network incorporating the two prior assumptions. It considers each pair of linked predictors in the network as a group. Similar as the Group Lasso penalty, it is capable to shrink the scaled coefficients of two linked predictors to similar levels and to enforce the coefficients of irrelevant predictor pair to 0. Scaling the coefficients in the penalty allows the predictors with larger degrees (e.g. hubs) in the network to have larger coefficients. Specifically, if $\gamma = 2$ and $w_i = w_j = 1$, the GBL penalty becomes the Group Lasso penalty, which intuitively accounts for its capability of group shrinkage that achieves the goal of the first priori assumption.

When parameter γ is determined, the multiplier $2^{1/\gamma'}$ becomes a constant and the penalty therefore exclusively depends on λ. By comparing the performances with different γ, the author suggested that a large γ will result in stronger shrinkage on coefficients and better performance for variable selection, and setting $w(i) = d_i^{\gamma}$ will reduce the bias in predicted errors. Consequently, the penalty function can be simplified based on these suggestions as follows:

$$P(\beta) = \lambda \sum_{i \sim j} \left[\left(\frac{|\beta_i|}{d_i} \right)^{\gamma} + \left(\frac{|\beta_j|}{d_j} \right)^{\gamma} \right]^{1/\gamma} \tag{2.25}$$

In particular, when $\gamma \to \infty$, the penalty becomes

$$P(\beta) = \lambda \sum_{i \sim j} \max \left(\frac{|\beta_i|}{d_i}, \frac{|\beta_j|}{d_j} \right) \tag{2.26}$$

Through the comparisons with Lasso, ElasticNet and Grace, Pan et al. [52] found that GBL outperforms them in variable selection, but suffers a stronger bias in the predicted error. Although the GBL penalty given by $\gamma \to \infty$ provides better performance than the other smaller γ, the predicted error is still not impressive.

2.3.2 Artificial neural networks

Deep Learning is an emerging subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. An artificial neural network, abbreviated as NN, initially inspired by neural networks in the brain [54, 55], which is defined as "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs" [56].

The basic structure of a neural network consists of layers of interconnected computing neurons, which can be classified in to three categories: the input layer, the hidden layer and the output layer, as shown in Figure 2.6. It receives data from the input layer, which

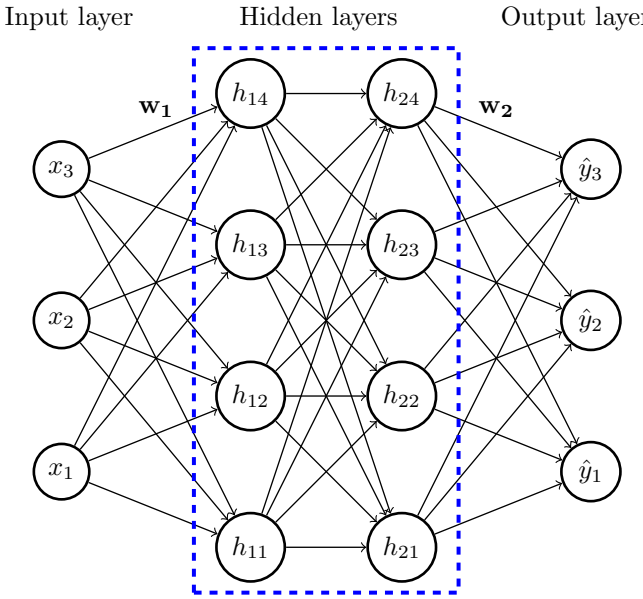


Figure 2.6: Basic structure of a neural network (NN). A basic NN consists of a input layer, several hidden layers and an output layer. Each node represents a neuron and arrows indicate the interconnections among neurons. The which consists of a input layer, the hidden layers and an output layer. The blue dashed rectangle indicates the hidden layers.

are then transformed in a non-linear way through the multiple hidden layers, before final outputs are computed in the output layer. Neurons in a hidden or output layer are connected to all neurons in the previous layer. The depth of a neural network is defined by the number of the hidden layers, and the width refers to the maximum number of neurons in one of its layers.

2.3.2.1 Artificial neuron

The fundamental unit of a neural network is an artificial neuron, illustrated in Figure 2.7. Each neuron computes a linear combination, i.e. weighted sum, of its inputs $x = (x_1, ..., x_n)$, and then applies a non-linear activation function f to calculate its output y:

$$y = f\left(b + \sum_{i=1}^{n} x_i w_i\right) = f(b + xw) \tag{2.27}$$

where $w = (w_1, ..., w_n)$ is the vector of weights corresponding to the input x, and b denotes the intercept of the linear combination. The activation function f can be defined by the following common ways:

- logistic function

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-(b+xw)}} \tag{2.28}$$

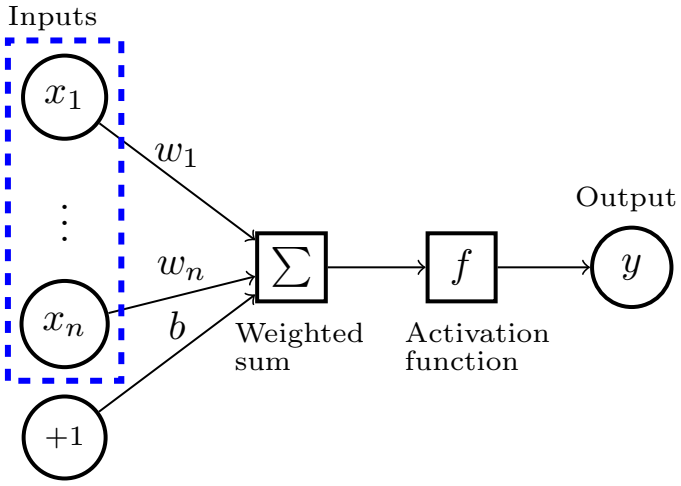


Figure 2.7: Illustration of an artificial neuron.

- hyperbolic tangent (tanh)

$$f(x) = \tanh(b + xw) \tag{2.29}$$

- rectified linear unit (ReLU)

$$f(x) = \max(0, (b + xw)) \tag{2.30}$$

- softmax function

$$f(x) = P(y = j|x) = \frac{e^{b_j + xw_j}}{\sum_{k=1}^{K} e^{b_k + xw_k}} \tag{2.31}$$

The first three functions can be used for both hidden layers and the output layer, while the softmax function is usually used for the output layer. The choice of activation functions depends on the goal of the problem. The logistic function provides an output representing the probability for binary values 0 and 1, which is used for binary classification problem. The softmax function provides an output representing a categorical distribution, i.e., a probability distribution over K different possible outcomes, which is used for multi-class classification problem. The hyperbolic tangent can result in a symmetric output ranging from -1 to 1. In practice, the most widely used activation function is the ReLU as it allows faster learning compared to others [57].

2.3.2.2 Neural network architectures

Several neural network architectures have been developed for specific applications, which depends on the way how the neurons are arranged, such as the convolutional neural network (CNN) for images [58] and the recurrent neural network (RNN) for sequential data [59]. Here, the neural network architectures that are commonly applied in biology

will be introduced, including the multilayer perceptron (MLP) [60], the recurrent neural network (RNN) and the Long Short-Term Memory Units (LSTM) [61].

MLP The multilayer perceptron is the most basic neural network, in which a sequence of layers are fully connected with at least one hidden layer. The neural network illustrated in Figure 2.6 is a MLP, where the neurons between two adjacent layers are all-against-all connected, but no connections exist between neurons within the same layer. MLP is a class of feedforward artificial neural network, in which connections between the neurons do not form a cycle. Each neuron in a hidden layer takes the outputs from the previous layer as its inputs and computes its output which is be used as an input to the neurons in the next layer.

The MLP is trained using the supervised learning technique backpropagation with gradient descent optimization, which learns the weights between neurons by minimizing the error between the predicted output and the actual observation defined by a loss function. It repeats a three-step cycle including propagation, backpropagation and weight updating:

1. In the propagation step, the neural network takes the input data from the input layer and propagate forward layer by layer until reaching the output layer. The errors are calculated for each neuron in the output layer according to the loss function.

2. In the backpropagation step, the resulting errors are propagated from the output layer back through the network, until each neuron has an associated error value that reflects its contribution to the original output.

3. In weight updating step, the gradient of the loss function, calculated using these error values, is fed to the optimization method to update the weights.

After repeating this cycle for a sufficiently large number of times, the network will usually converge to some state where the prediction error is smaller than expectation. In this case, we would say that the network has learned a certain target function $\hat{F}(x; w, b)$, which can be used to make predictions for future data that are unlabelled.

RNN Unlike MLP analysing all the elements of the input vector x simultaneously, a recurrent neural network (RNN) processes the inputs as a sequence of time-steps $x = (x_1, x_2, ..., x_T)$, where x_t is the input at time-step t. RNN is a class of neural network architecture, different from feedforward neural network, which allows for cycles, illustrated in Figure 2.8.

At each time-step, RNN applies the same operation:

$$h_t = \sigma(w_{hx}x_t + w_{hh}h_{t-1} + b_h) \tag{2.32}$$

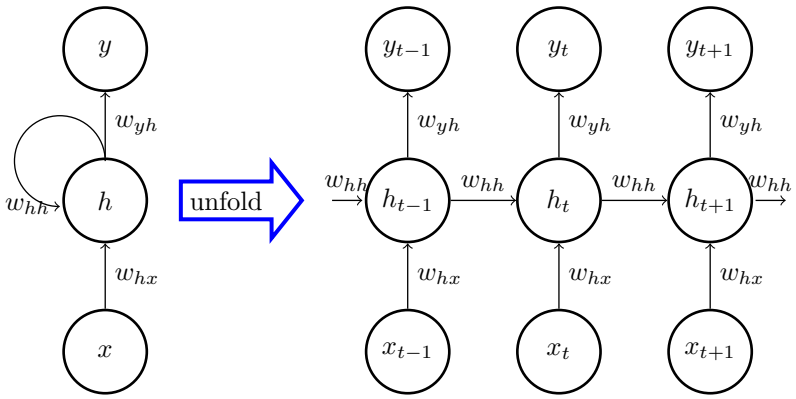


Figure 2.8: Recurrent neural network (RNN).

where σ is a non-linear activation function. h_t is a vector of hidden state of the RNN, which will be sequentially updated based on the current input x_t and the previous hidden state h_{t-1}. It is also referred to as the memory of the neural network as it memorizes the input sequence $(x_1, ..., x_t)$ up to the time-step t. The final hidden state h_T memorizes the whole input sequence. An important characteristic of the RNN is that the weights w_{hx}, w_{hh} and the bias b_h are shared across all the time steps. The output y_t at time-step t depends on the hidden state h_t, and therefore, the whole previous sequences:

$$y_t = f(w_{yh}h_t + b_y) \tag{2.33}$$

f is an activation function, which is chosen according to the aim of the task, such as the logistic function to model binary outputs and the softmax function to model categorical outputs. Th output of RNN can be either the single y_T or a vector of the sequence of outputs $y = (y_1, ..., y_t)$. The RNN parameters $w_{hx}, w_{hh}, b_h, w_{yh}, b_y$ are also trained by using backpropagation with gradient descent optimization, which is similar with training a MLP.

LSTM RNN has been considered difficult to train because of its long computation paths. An incorrect parameter initialization can lead to exploding or vanishing gradients. Several advanced RNN architectures have been developed to address these problems such as the long short-term memory (LSTM) units. The core idea of LSTM is using additional gates to update the memory of the network only at certain time steps, which will help keep the gradients stable.

The basic LSTM block has much more internal structure than the standard RNN, illustrated in Figure 2.9. The centre of the LSTM block is a memory cell which maintains the information over the time steps. At each time step, the current information is obtained from the inputs of the LSTM block through a tanh activation function:

$$v_t = \tanh(w_v x_t + u_v h_{t-1} + b_v) \tag{2.34}$$

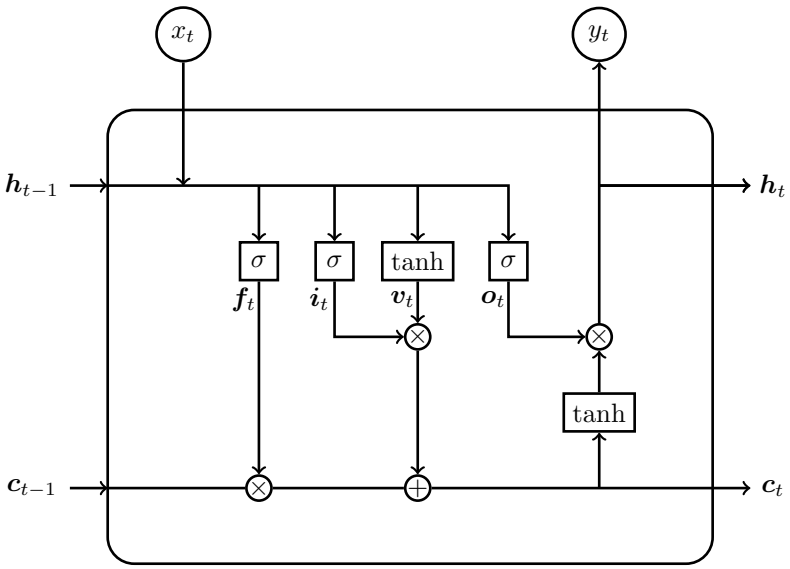


Figure 2.9: Long Short-Term Memory Unit (LSTM).

where w_v, u_v are the weights and b_v is the bias for the LSTM block.

The current state of the memory cell is calculated from the current inputs and the previous states of memory cell, which is modulated by the following logistic gates:

- The forget gate computes a linear function of its inputs, followed by a logistic activation function. If the gate is on (i.e., outputting 1), the memory cell remember its previous value, otherwise, the cell forgets the previous value.

$$f_t = \sigma(w_f x_t + u_f h_{t-1} + b_f) \tag{2.35}$$

- The input gate has the same linear-then-logistic function as the forget gate, which controls whether the memory cell receive the inputs from other LSTMs in the network. If the gate is on, the summed inputs are passed through a tahn activation function and then added to the memory cell.

$$i_t = \sigma(w_i x_t + u_i h_{t-1} + b_i) \tag{2.36}$$

- The output gate performs the linear-then-logistic function as well, which controls whether pass on the LSTM output to the rest of the network. If the gate is on, the value of the memory cell is passed through a tanh activation function and then passed on to the rest of the network.

$$o_t = \sigma(w_o x_t + u_o h_{t-1} + b_o) \tag{2.37}$$

where $w_f, w_i, w_o, u_f, u_i, u_o$ are the weights and b_i, b_f, b_o are the biases of the corresponding

gates. Based on the status of the forget and input gates, the current states of the memory cell is updated following:

$$c_t = v_t \otimes i_t + c_{t-1} \otimes f_t \tag{2.38}$$

The behaviours of the memory cell responsive to different states of the gates are summarized in Table 2.1.

Table 2.1: LSTM memory cell operations modulated by forget and input gates.

Forget gate	Input gate	Memory cell behaviour
1	0	keep the previous value
1	1	add the input value to the previous value
0	0	clear the previous value
0	1	replace the previous value by the input value

Finally, the output of the LSTM block is controlled by the status of the output gate:

$$h_t = \tanh(c_t) \otimes o_t \tag{2.39}$$

2.3.3 Support vector machine

The support vector machine (SVM) [62] is a popular and powerful machine learning method. It is a supervised learning technique which can be used for both classification and regression. In its standard form, a SVM model performs binary classification, which maps the data into a higher-dimensional space where the two classes are separated by a hyperplane. The goal of the SVM model is to maximise the gap between the separating hyperplane and it thus results in minimised expected generalization error.

Let (x_i, y_i) denote the sample i $(i = 1...n)$ in a labelled data, where x_i represents a vector of real numbers (referred to as features) and $y_i \in \{0, 1\}$ presents the labels of the two classes. The goal is to find a discriminant function f mapping the input vectors x onto labels y which minimizes the misclassified number $(f(x) \neq y)$. A linear classifier is constructed based on the linear discriminant function f as follows:

$$f(x) = wx + b \tag{2.40}$$

where w is the weight vector corresponding to the features in x and b is the bias. The space is divided into two sets according to the sign of $wx + b$. The linear classifier is defined as

$$h(x) = \begin{cases} 1, & \text{for } wx + b \geq 0 \\ 0, & \text{for } wx + b < 0 \end{cases} \tag{2.41}$$

The optimal linear discriminant function can be estimated by minimizing the objective

function

$$\min_{w} \frac{1}{2}||w||^2 \tag{2.42}$$

subject to specific constraints

$$\begin{cases} wx \geq 1, & \text{if } y = 1 \\ wx \leq -1, & \text{if } y = 0 \end{cases} \tag{2.43}$$

With these constraints, the model ensures that the selected the linear discriminant function has the largest distance from the closest data points, while at the same time the classifier minimises the classification errors on future unlabelled data. If all the samples can be classified correctly by a linear classifier, we call the data linearly separable.

In practice, most data are not linearly separable in the original feature space. One solution to address this problem is using the kernel functions mapping the original features into a higher-dimensional space where the transformed data are linearly separable. Four commonly used kernel functions include: the linear kernel, the polynomial kernel, the radial basis function kernel and the sigmoid tanh kernel.

Linear kernel The linear kernel is usually used when the data is close to being linearly separable, which represents a simple scalar product of two feature vectors x_i and x_j:

$$K(x_i, x_j) = x_i^T x_j \tag{2.44}$$

Polynomial kernel The polynomial kernel describes the similarity of samples in a feature space over polynomials of the original features:

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \quad \gamma > 0 \tag{2.45}$$

where x_i and x_j are two feature vectors, and γ, r and d are kernel parameters.

Radial basis function (RBF) kernel The radial basis function kernel, also called the Gaussian kernel, is a polynomial kernel with infinite degree. Its features are all possible monomials of the input original features without degree restriction:

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2), \quad \gamma > 0 \tag{2.46}$$

where x_i and x_j are two feature vectors, and γ is the kernel parameter.

Sigmoid tanh kernel The sigmoid tanh kernel provides the tanh of a scaled and shifted scalar product:

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r), \quad \gamma > 0 \tag{2.47}$$

where x_i and x_j are two feature vectors, and γ, r and d are kernel parameters.

2.3.4 Random forest

Random forest [63] is an ensemble learning method for various machine learning tasks such as classification, regression and so on. It constructs a multitude of decision trees and outputs the class that is the label of the classes (classification) or the mean prediction (regression) of the individual trees.

2.3.4.1 Decision tree

Decision tree [64] is a popular technique for various machine learning tasks. It is a tree-like structure that consists of four main parts: a root node, internal nodes, leaf nodes and branches. The root node is the starting point of the tree. Each internal node represents a test on an attribute (e.g. a feature in the data). Each branch represents the outcome of the test (e.g., conditions of the tested feature). Each leaf node represents a class label, i.e., the decision that has been taken after testing all attributes. The paths from the root node to the leaf node define the classification rules, also called decision rules. The generalized form of the rules follows:

$$if \text{ condition 1 } and \text{ condition 2 } and...and \text{ condition n } then \text{ outcome.}$$

The most widely used methods for training decision trees are greedy algorithms. A tree can be learned by splitting the population of the data into subsets based on the test of a feature. The process is repeated on each derived subset in a recursive way (referred to as recursive partitioning). The recursive partitioning is completed when the subset at a node has all the same class labels, or when the splitting does not contribute to the performance of the classification.

Compared with other classification methods, decision trees have several significant advantages. Decision trees are simple to understand and interpret due to the intuitive decision rules. They work with low quality data because they do not require the parametric distributions of the input data and are insensitive with the missing data. They also work with different types of features at the same time, e.g., scalar variables, categorical variables and boolean variables. For large datasets, decision trees provide high efficiency because we only need to construct one tree for the data and during each prediction, the maximum of the tests is no more than the tree depth.

However, decision trees also have some disadvantages. They are unstable because a small change in the data can lead to a large change in the tree structure. Consequently, decision trees exhibit high variance when they are learned by different training and test sets of the same data, which results in overfitting on the training data. For data including

categorical variables with different numbers of levels, decision trees tend to be biased in favour of the ones with more levels. They also suffer low classification accuracy for the data with correlated features.

2.3.4.2 Random forest

Random forests, also called random decision forests, are a popular ensemble method that can be used to build predictive models for both classification and regression tasks. Ensemble methods use multiple learning models to gain better predictive results. In the case of a random forest, the model creates an entire forest of random uncorrelated decision trees to achieve the best predictions.

A random forest consists of an arbitrary number of simple decision trees, which are used to determine the final outcome. Each simple tree is growing based on a random subset of features chosen independently (with replacement) from the original data. The selected feature subsets follow the same distribution for all trees in the forest. All the trees are capable of producing an outcome for the same task of the random forest. For classification tasks, the ensemble of simple trees vote for the most popular class. The random forest defines a margin function that measures the degree to which the average number of votes for the correct class exceeds the average vote for any other class present in the response variable. For regression tasks, the outcomes of all simple trees are averaged to obtain an estimate of the response variable.

Because the random forest method is based on decision trees, it keeps all the advantages of decision trees. Furthermore, it overcomes the overfitting limitations of decision trees by using tree ensembles which can lead to significant improvement in predictions for new unlabelled data.

2.3.5 Classification evaluation

2.3.5.1 Evaluation strategy

For the classification evaluation on a dataset, the classifier is learned on a subset of total samples (referred to as training set), but is validated on another independent subset of samples (referred to as test set). There is not overlapping between the training set and the test set, which avoids the overfitting of the classifiers. Three strategies are mainly used for classification evaluation: the hold-out validation, the K-fold cross-validation, and the stratified nested K-fold cross-validation.

Hold-out validation For the hold-out validation, the samples of the data are randomly separated into two non-overlapping subsets. One subset is used as the training set and the other is used as the test set. A classifier is trained on the training set. A prediction is then

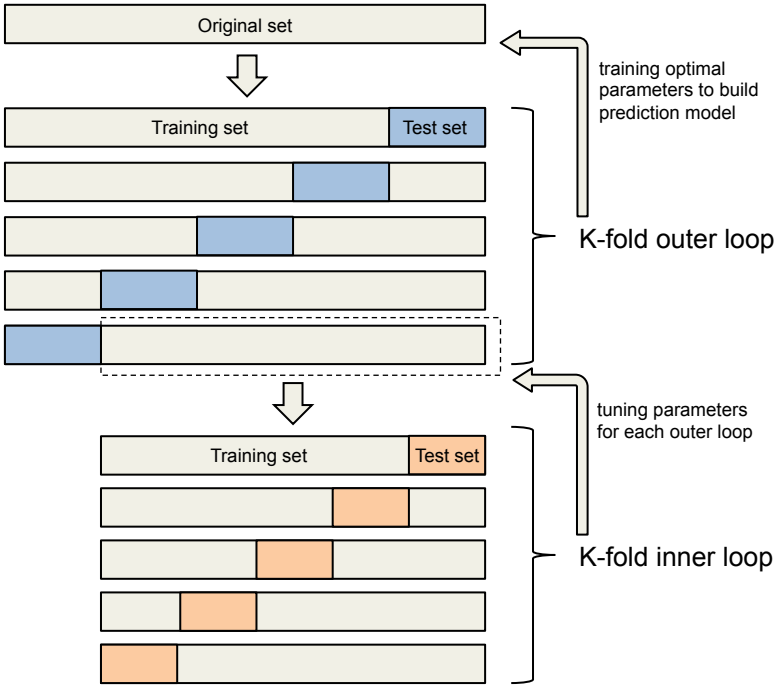


Figure 2.10: Stratified nested K-fold cross-validation.

performed on the test set using the trained classifier. The performance of the classifier is assessed by evaluation measures on the test set. Hold-out validation is usually used for the data with large number of samples.

K-fold cross-validation The samples of the data are equally separated into K folds. $K-1$ folds are used as training set to train a classifier. Then, the remaining fold is used as the test set where a prediction is performed by the trained classifier. The procedure is implemented iteratively for K times until every fold has been used as test set. The performance of the classier is assessed by the summary of evaluation measures on K test sets. Specifically, when setting K as the sample size of the data, the cross-validation is referred to as leave-one-out cross-validation.

Stratified nested K-fold cross-validation Stratified nested K-fold cross-validation is based on K-fold cross-validation. It consists of two loop procedures, the outer loop and the inner loop, illustrated in Figure 2.10. The original sample set are equally separated into K folds. In an outer loop step, $K-1$ folds are used as training set and the remaining fold is used as test set. An optimal classifier is learned through an inner loop of a nested K-fold cross-validation on the training set. Then the test set is used to evaluate the performance of the optimal classifier. The outer loop is implemented iteratively for K steps until each fold has been used as test set. The final optimal classifier is selected from the outer loop step which provides the best performance on the corresponding test set.

2.3.5.2 Evaluation measure

For bi-class classification problems, the prediction performance of classifiers can be evaluated by several common measures. The actual labels for test samples are separated into two classes: actual positive and actual negative. The predicted labels for the same test samples are also separated into two classes: predicted positive and predicted negative. The confusion matrix for a bi-class classifier is given by Table 2.2.

Table 2.2: Confusion matrix for a classification.

Classes	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Several common evaluation measures for classification are defined based on the confusion matrix:

- *accuracy*
  The measure *accuracy* is the proportion of the total samples which are predicted correctly:
  $$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.48}$$
- *precision*
  The measure *precision* is the proportion of the predicted positive samples which are correct:
  $$precision = \frac{TP}{TP + FP} \tag{2.49}$$
- *recall*
  The measure *recall* is the proportion of the actual positive samples which are identified correctly:
  $$recall = \frac{TP}{TP + FN} \tag{2.50}$$
- F-score
  The measure F-score is the harmonic mean of *precision* and *recall*:
  $$F\text{-score} = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.51}$$
- False discovery rate (FDR)
  The measure FDR is the proportion of the predicted positive samples which are not correct:
  $$FDR = \frac{FP}{TP + FP} \tag{2.52}$$

2.3.6 Clustering

In contrast to the aforementioned supervised learning methods, unsupervised learning methods are methods in which we draw references from datasets consisting of input data without labelled responses. Generally, it is used as a process to find underlying structures in an unlabelled dataset.

Clustering is a type of unsupervised machine learning technique, which aims to group unlabelled data points into inherent subsets or clusters. The task of clustering is to divide data points into a number of groups based on similarity and dissimilarity between them. Data points within a same group should be as similar as possible and data points in one group should be as dissimilar as possible from data points in another group. One of the most common clustering method is hierarchical clustering which aims to build a hierarchy of clusters from the unlabelled data points. Strategies for hierarchical clustering mainly fall into two types [65]: the "bottom-up" strategy which begins with each data point as a singleton cluster and successively merges pairs of clusters as moving up the hierarchy, and the "top-down" strategy which starts from a whole cluster consisting of all data points and splits clusters recursively as moving down the hierarchy.

2.3.7 Network clustering

Network clustering is a type of clustering technique which aims to discover clusters from a network structured unlabelled dataset. The dataset can be viewed as a network where nodes represent data points and edges represent the predefined correlated relationships between data points. The task is to discover the highly connected subnetworks or modules through clustering over the network.

Biological network analysis has become an important field of bioinformatics and network science because network properties provide clues to understand potential mechanisms of biological systems. Significant efforts have been made on studying the topology and structure of molecular networks to infer hidden patterns from the network. As aforementioned, modularity is an important feature of biological networks which suggests the existence of modules in the network. A group of nodes that are tightly connected among each other within the group form a module. A gene module in biological networks is most likely to be a function unit in a biological process or pathway [66]. Detection of the functional modules from biological networks will provide important clues for understanding underlying mechanisms. This task can be addressed by applying network clustering methods to biological networks.

Biological network clustering algorithms can be broadly categorized as topology-free methods and graph-based methods. Topology-free clustering methods use traditional clustering techniques (e.g. hierarchical clustering) on the basis of distances between nodes

which do not take into account the topology of the network. Graph-based clustering methods incorporate the topology of the network by applying specialized clustering techniques. The common techniques that have been successfully used for biological network clustering based on topology generally fall into four categories [67]: local neighbourhood density search (LD), flow simulation (FS), link clustering (LC) and cost-based local search (CL).

Local neighbourhood density search (LD) The LD methods are based on local optimization strategies designed to discover dense subnetworks from a network. In a dense subnetwork, each node is connected to many other nodes. The LD methods aim to maximize the local density of each subnetwork. One of the most typical methods of LD category is MCODE [68]. MCODE aims to detect dense and connected modules by weighting nodes based on their local neighbourhood density. To address this task, the k-core concept is applied. A k-core is defined as a subnetwork in which each node has a degree of at least k. The highest k-core of a network is the most densely connected subnetwork with the highest k. The weight of each node is defined based on its topological relationships with the highest k-core. MCODE selects a certain number of nodes with highest weights as seeds. For each seed, MCODE considers it as an initial cluster and recursively merges neighbouring nodes into the cluster if their weights are above a fixed threshold until there are no neighbouring nodes can be added.

Flow simulation (FS) The FS methods discover subnetworks from biological network by mimicking the spread of information on the network using random walk [69] or biological knowledge for passing information between genes. One of the most well-known methods of FS category is MCL [70]. In a network, a random walk (or flow) means that the direction walking (or flowing) from a node to its neighbouring nodes is assigned by chance. MCL simulates many random walks (or flows) within a network according to the strength of flows, i.e., strengthening the flow where the flows are strong and weakening it where the flows are weak. By repeating such simulations, several subnetworks reveal with strong internal flow, separated by boundaries without flows between subnetworks.

Link clustering (LC) The LC methods consider the edges of a network rather than the nodes as the data points that are to be clustered. They group the set of edges based on the similarities between edges. The nodes associated with the edges within a cluster are assigned as a subnetwork. The LC methods allow to discover overlapping subnetworks because the edges of a node can be grouped into different clusters and the node is thus assigned into different subnetworks. One of the most typical LC methods is LinkComm [71]. LinkComm performs the bottom-up clustering to group edges of a network into topologically related clusters. It applies hierarchical clustering to build a dendrogram

based on similarities between edges which are estimated taking into account the size of both the intersection and the union of their neighbourhoods. The dendrogram is cut with the best partition density to obtain the edge clusters. The nodes associated with the edges within each cluster form a subnetwork.

Cost-based local search (CL) The CL methods divide the input network into connected subnetworks by a cost function that guides the search toward a best partition of the network. The methods of this category are flexible because we can define the cost functions based on practical requirements. The CL methods, therefore, are the widely used category. Two common CL methods for community detection from biological network is ModuLand and OCG. ModuLand [72] is a family of integrative methods for detecting overlapping network modules as hills of an influence function-based centrality-type community landscape and including several widely used modularization methods as special cases. OCG [73] is a recent CL method which decomposes the input network into overlapping clusters. OCG first covers the network with initial overlapping classes that are considered as leaves of a tree, and then fuses the classes progressively and hierarchically in a bottom-up way by maximizing a cost function defined based on the overlapping modularity. It stops when no further fusions can produce a gain in modularity.

Chapter 3

Network overlapping module detection for transcriptome and interactome integration

3.1 Introduction

Transcriptomic profiling technologies such as microarray [74] and RNA-seq [23] assess genome-wide gene expression of a cell in different conditions. Transcriptome data reveal the dynamic expression patterns of genes responsive to external environments, which helps to systematically understand the underlying molecular mechanisms. The most important hypothesis for transcriptomics is that genes involving in the same biological processes tend to exhibit similar expression patterns which are referred to as co-expression [75]. A group of highly co-expressed genes, therefore, are usually suggested to be a potential functional module which is responsive to a specific dynamic environment or is regulated by a same molecular regulator. Such groups of co-expressed genes are defined as functional gene modules which provide interesting clues for exploring the underlying molecular regulation mechanisms. Many methods have been proposed to identify functional gene modules based on gene co-expression using probabilistic graphical models, hierarchical clustering and network clustering methods [76, 77, 78]. However, the co-expressed functional gene modules inferred from transcriptome data suffer large false positives because of the high noise existing in the gene expression measured by the high-throughput technologies.

To reduce the noise in the inferred modules, interactome data have been integrated with transcriptome data for detection of functional gene modules. Interactome data provide a comprehensive reference of functional links among genes by the accumulating collection of molecule interactions discovered in multiple conditions, which form a large and highly connected network composed of nodes denoting genes and edges representing interactions between them. Combining with transcriptome data, a co-expressed gene pair which are

linked in the network usually indicates a condition responsive functional interaction. Therefore, a highly connected subnetwork composed of co-expressed interactions is considered as a reliable functional gene module associated with the condition. Several methods have been developed to detect functional gene modules through the integration of transcriptome and interactome data [79, 80, 81, 82, 83, 84].

In spite of the success of previous methods in functional module detection by integrating transcriptome and interactome data, there are still some limitations: (i) most of the previous methods have been developed for case-control transcriptome data but not taken into account the multi-condition data; (ii) when selecting condition responsive modules, most of the methods consider a module as a whole unit but ignore the impact of different genes within the module.

We proposed an approach to identify condition-specific responsive functional gene modules by integrating transcriptome and interactome data using network overlapping module detection method, which is capable of not only identifying the responsive functional modules but also selecting important genes within the modules.

3.2 Multi-omics

3.2.1 Multi-omic data

3.2.1.1 Transcriptome

The transcriptome data studied in this chapter consist of the transcriptomes of human embryos encompassing multiple pre-implantation development stages.

Human pre-implantation embryonic development is a crucial period of individual life, which refers to the time from final maturation of the oocyte following by the fertilization through the development of the early embryo before the implantation in the uterus [85]. The pre-implantation development encompasses a series of consecutive stages as shown in Figure 3.1, including mature oocyte, fertilized oocyte, 2-cell embryo, 4-cell embryo, 8-cell embryo, morula and blastocyst [86].

The transcriptomes of 15 human embryos, encompassing a range of important pre-

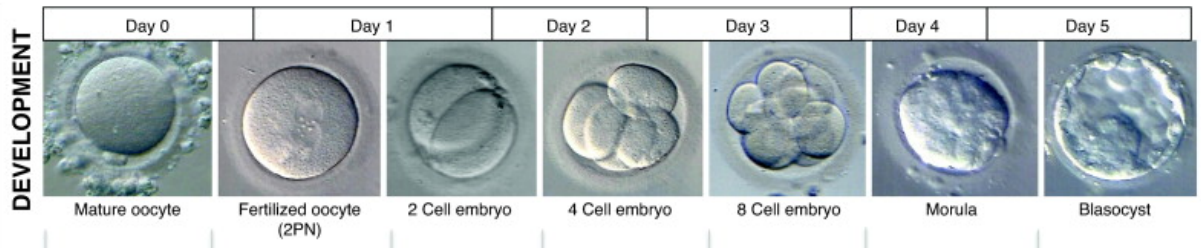


Figure 3.1: Human pre-implantation embryonic development stages illustrated in [86].

implantation development stages, consist of four oocytes, four 4-cell embryos, three 8-cell embryos and four blastocysts. The raw Affymetrix microarray data are provided by our collaborator in EpiHealthNet ITN, Prof Daniel Brison from The University of Manchester, which can be accessed from the Gene Expression Omnibus (GEO) with the accession number GSE110693. Genome-wide gene expression values of each embryo are obtained by pre-processing the raw data using RMA [87] and mas5call [88], which are implemented in R using the affy package [89].

3.2.1.2 Interactome

Human gene interactome data are downloaded from PathwayCommons (version 8) [90], which is a collection of public available human pathway data. The interactome data provide comprehensive functional interactions, such as biochemical reactions, complex assembly, transport and catalysis events, and physical interactions, among molecules including proteins, DNA, RNA, small molecules and complexes. All the molecules in the interactome data are mapped to the corresponding genes. After removing the duplicated interactions and self-interactions, the remaining interactions form the gene-gene interaction network.

3.2.2 Problem definition

Pre-implantation embryonic development is a crucial period of human life [85]. It is easily affected by the abnormal factors to maternal environment such as overstress, unhealthy diet and in vitro fertilization (IVF) [86], which has been reported to have long-term impact on adult's health [91]. Understanding the pre-implantation development will help to decrease the risk of health problems in the later life. However, due to limited resources, the molecular basis of human pre-implantation embryonic development is poorly understood. During the pre-implantation development, human embryos undergo the dramatic changes of gene expression patterns in response to the series of development stages. The goal of this study is to identify the underlying patterns associated with mechanisms of embryonic development by integrating the transcriptome and interactome data.

In this chapter, the transcriptome data provide the dynamic information of gene expression during embryo development across four stages. Following the aforementioned hypothesis in Chapter 3.1, a group of highly co-expressed genes tend to involve in same functions related with embryonic development. The interactome data from PathwayCommons provide a comprehensive functional context of genes, which forms a large gene-gene interaction network. This network is static as it is a collection of gene functional interactions observed from various experimental conditions, e.g., different cell lines, different tissues, different diseases and so on. From the development point of view, this network can be

considered as an aggregation of gene functional interactions from different development periods including the embryonic development stages and different ages of adults. In this comprehensive static interaction network, if two interacted genes are co-expressed across multiple embryonic development stages, the gene pair is responsive to embryonic development. Consequently, all dynamic co-expressed gene pairs responsive to embryonic development can be extracted from the static network by integrating the transcriptome data of embryos from series of development stages. All the responsive gene pairs form a dynamic co-expressed gene-gene interaction network which is associated with human embryonic development. In this co-expression gene network, if a module contains the genes not only highly functionally connected but also significantly co-expressed, the module is considered as a co-expressed gene module which tends to be associated with embryonic development. Identifying key genes and gene modules from the dynamic co-expression network will help to understand the mechanisms of embryonic development. The research problem can be considered as an unsupervised learning task to detect functional gene modules from the co-expression network. The task can be solved by applying efficient network module detection methods to identify gene modules from co-expression network and using feature selection methods to select key genes and modules associated with specific development stages.

3.3 Methodology

The task of this study is to identify key genes and gene modules associated with human embryonic development by integrating the transcriptome data of embryos and the human gene interactome data from PathwayCommons database. The task can be solved by dividing into four subtasks: Firstly, constructing a co-expression network using the transcriptome and interactome data; Then, detecting gene modules from the co-expression network; Next, identifying significant modules associated with embryonic development; Finally, selecting feature genes and modules associated with specific embryonic development stage from the significant modules.

A number of approaches have been developed for identifying experiment condition related functional gene modules through co-expression network clustering analysis [80]. The common pipelines for most existing approaches are generally solve the first three up-mentioned subtasks in three steps [80]. In the first step, individual relationships between genes are estimated based on correlation measures between each gene pair. In the second step, the co-expression correlations are used to construct a network where nodes represent genes and edges represent the strength of the co-expression relationships. In the third step, co-expression gene functional modules are identified from the co-expression network by using available clustering techniques. These co-expressed modules can subsequently

be correlated to the samples corresponding to a specific condition (e.g., disease status or tissue type).

Some frequently used pipelines and tools for identifying such co-expressed modules includes WGCNA [92], CoXpress [93], DiffCoEx [94] and DINGO [95]. All of them first identify modules co-expressed across all conditions, and then find the modules associated with specific conditions. WGCNA is one of the most widely used method for co-expression analysis. It first constructs a co-expression network using default Pearson Correlation Coefficient (PCC) or a custom distance measure, and then performs hierarchical clustering using tree cutting techniques to identify co-expression gene modules. CoXpress identifies modules in which the genes are co-expressed in one condition and evaluates whether the genes are also co-expressed under other conditions. DiffCoEx uses a similar approach to WGCNA to identify and cluster differentially co-expressed genes. It identifies modules of genes that have the same different partners between two different conditions. DINGO is a recent tool that clusters genes based on their differential expression levels in a group of samples (e.g., under a particular condition) compared with the baseline co-expression estimated across all samples.

The up-mentioned co-expression analysis methods are developed to identify co-expression gene modules only using gene expression data, which usually lead to high false positives in the inferred modules because the curated functional relationships such as protein-protein interactions and gene-gene functional links between genes have not been taken into account. In order to integrate such functional information into module identification, many methods have been developed to integrate gene expression with molecular functional interaction network to infer co-expressed interacted gene functional modules [96]. For example, two recent methods, COSINE [97] and BMRF [98], are developed for identification of gene modules by integrating bi-class condition gene expression data (i.e., case vs. control) and gene interaction network. COSINE identifies a single optimal subnetwork by a genetic algorithm maximizing the scoring function with two measures for both nodes (genes) and edges (co-expressed genes) in change of the expression pattern. BMRF models the gene expression data and interaction data with Markov Random Field and searches subnetworks with maximum posteriori estimation using the bagging aggregation scheme algorithm. Besides, some methods have been developed to address the same tasks for multi-class condition gene expression data. For example, one of the most recent methods of this category is developed by Shen et al. [99], which extracts gene modules from multiple time points gene expression data and gene interaction network. It identifies the active time points of genes by constructing a co-expression network based on Connected Affinity Coefficient and Pearson Correlation Coefficient measures.

Regarding the up-mentioned typical state-of-art methods for identification of co-expression gene functional modules, the co-expression analysis methods such as WGCNA

[92], CoXpress [93], DiffCoEx [94] and DINGO [95], are developed to identify co-expression gene modules based on the differential expression of genes between two classes of conditions, and thus they are well suited the bi-class condition gene expression data analysis. For the multi-class condition gene expression data, these methods can be used indirectly by transforming the multi-class condition tasks into bi-class condition tasks, i.e., comparing one class of condition against all other classes of conditions. However, these methods only use gene expression data without the combination with gene functional interaction data, which may lead to high false positives in the identified gene modules. Consequently, these co-expression only methods are not suitable for the tasks in this chapter. For the methods that identify gene functional modules by integrating gene expression data with gene functional interaction data such as COSINE [97] and BMRF [98], they are developed to work only with bi-class condition gene expression data, and they are therefore not applicable to tasks in this chapter as the embryonic development gene expression data include multiple conditions/stages. The method proposed by Shen et al. [99] suits the multi-class condition tasks, but it is not capable of achieving the goal of the last subtask in this chapter which aims to select both condition-specific genes and modules. Since the existing approaches are not established comprehensively to address all the aforementioned four subtasks of this chapter as a whole pipeline, we aim to develop a novel computational framework which is capable of addressing all the subtasks by employing common methods that have been successfully used to address each subtask in the field of computational biology.

In order to identify functional gene modules associated with human embryonic development, we propose an approach for gene module detection by integrating multi-condition transcriptome data and interactome data using network overlapping module detection method. It consists of four steps to address the aforementioned four subtasks respectively. The flowchart of the proposed computational framework is shown in Figure 3.2, including four steps: (1) construction of gene co-expression network, (2) detection of overlapping modules, (3) identification of condition-associated modules, and (4) selection of condition-specific modules and genes. Detailed methodologies for each step are described in the following sections.

3.3.1 Construction of gene co-expression network

For the first subtask of constructing the co-expression network, the common strategy is to evaluate the co-expression between each interacted gene pairs in the interactome data and only the gene pairs that are significantly co-expressed are kept to form the co-expression network. The co-expression for each gene pair is estimated by the Pearson Correlation Coefficient (PCC) which evaluates the linear correlations between two scalar variables. It is a parametric hypothesis test that requires the two scalar variables both following normal

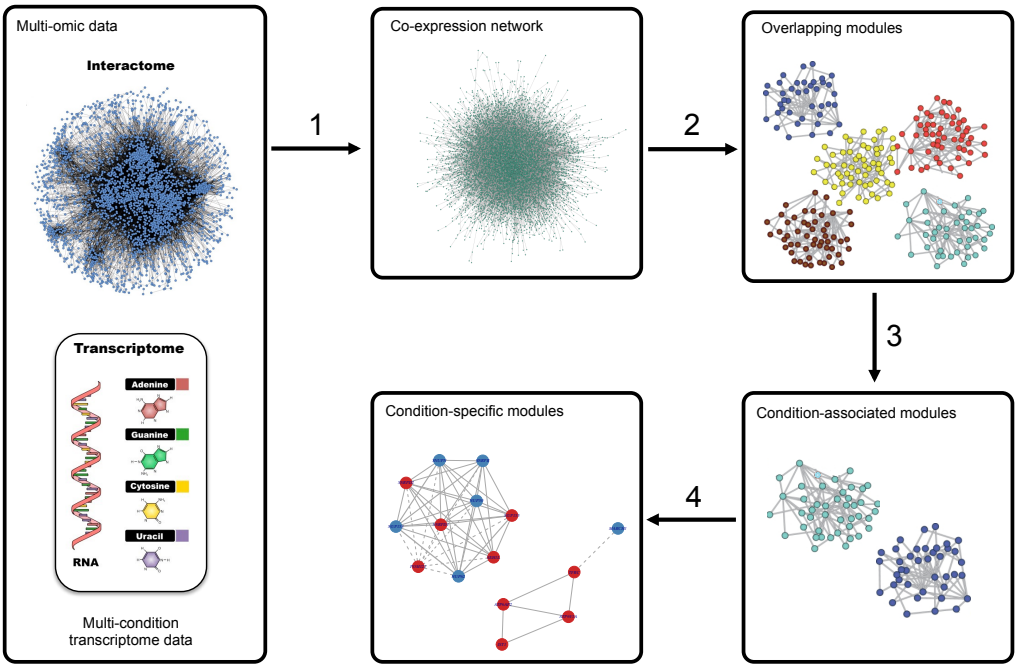


Figure 3.2: Flowchart of proposed approach to network overlapping module detection for transcriptome and interactome integration. It consists of four steps: (1) construction of gene co-expression network by evaluating co-expression correlation coefficient between each interacted gene pair based on their gene expression; (2) detection of overlapping gene modules from the co-expression network using network overlapping module detection method; (3) identification of condition-associated modules by assessing the significance of enrichment with condition-associated genes within the modules using ANOVA-GSEA; (4) selection of condition-specific feature modules and feature genes using GEL logistic regression with K-fold cross-validation.

distributions. PCC is the most commonly used method for gene co-expression estimation because the processed gene expression abundances usually follow normal distributions. The co-expression PCC of a gene pair estimates the concordance of the changing patterns between the two genes across all samples. A higher PCC suggests the two genes are co-regulated responsive to the experiment conditions, and thus they are likely involved in the same biological processes and perform similar functional roles. In the first step of the approach, we construct a gene co-expression network by integrating gene expression data and gene interaction data using PCC to evaluate co-expression correlations between genes.

Let X be a matrix of the multi-condition gene expression data with p genes in rows and n samples in columns. Let $x_i = (x_{i1}, ..., x_{in})$ be a vector of expression values of gene i across the n samples. Let $y = (y_1, ..., y_n)$ be a vector of the conditions corresponding to the samples, which consists of k different classes of conditions. In the human embryonic development study, the samples are the embryos and y is the vector of the development stages corresponding to the embryos, including "oocyte", "4cell", "8cell" and "blastocyst".

Let $G = (V, E)$ be a simple connected graph with p vertices and m edges ($|V| = n, |E| =$

m), which represents the network of human gene interactome including m functional links among p genes.

A co-expression network across multiple conditions (i.e., human embryonic development stages) is constructed based on human gene-gene interaction network by evaluating the correlation between the expression of each linked gene pair. Given an edge $e(i,j)$ in G which linked gene i and gene j, let $x_i = (x_{i1}, ..., x_{in})$ and $x_j = (x_{j1}, ..., x_{jn})$ be the vectors of expression values in all samples for gene i and j respectively. Let $\overline{x_i}$ and $\overline{x_j}$ be the average of x_i and x_j. The co-expression correlation w_{ij} of edge $e(i,j)$ is calculated by Pearson Correlation Coefficients (PCC) between x_i and x_j:

$$w_{ij} = \frac{\sum_{k=1}^{n}(x_{ik} - \overline{x_i})(x_{jk} - \overline{x_j})}{\sqrt{\sum_{k=1}^{n}(x_{ik} - \overline{x_i})^2}\sqrt{\sum_{k=1}^{n}(x_{jk} - \overline{x_j})^2}} \tag{3.1}$$

The statistical significance p_{ij} of w_{ij} is obtained by the p-value evaluated by the hypothesis test of PCC. The co-expression network, which is responsive to human embryonic development, is constructed by keeping the edges with significant w_{ij} whose $p_{ij} \leq 0.05$. The graph G for the co-expression network is redefined by the m_c edges in the co-expression network, where $|E| = m_c$.

3.3.2 Detection of overlapping modules

On the basis of the co-expression network constructed from the first step, the second subtask is to detect gene modules from the co-expression network. It can be solved by employing network module detection algorithms. Network module detection is a well-studied subfield in the fields of both network science and computational biology. Numerous network clustering algorithms have been developed to identify co-expression gene modules from gene expression data [100, 101], which can be generally categorized as topology-free methods and graph-based methods (see details in Chapter 2.3.7). The topology-free clustering methods do not take into account the topological characteristics of nodes in the network, which may ignore some important information because the critical topological roles of genes in biological networks usually suggest important functional roles of the genes (see details in Chapter 2.2.1). On the contrary, graph-based clustering algorithms incorporate the network topology by applying specialized clustering techniques which are generally fall into four main categories: local neighbourhood density search (LD) such as MCODE [68], flow simulation (FS) such as MCL [70], link clustering (LC) such as LinkComm [71], and cost-based local search (CL) such as ModuLand [72] and OCG [73]. Details of these methods have been described in Chapter 2.3.7. The main limitation of MCODE and MCL is that they do not allow the participation of a node to more than one modules. But in practice, there is not clear boundaries for biological pathways, and they are therefore overlapping to each other. Several multi-function genes

may take part in multiple biological functions and belong to multiple modules. Thus, a biological network with a modular structure will contain multiple overlapping modules. LinkComm, ModuLand and OCG are capable of identifying overlapping modules through network clustering. In terms of implementation, LinkComm may become computationally expensive for large dense network [67]. ModuLand and OCG provide a Cytoscape plug-in and an R package, respectively, for implementation. To address this subtask, the method OCG is chosen because we implement the proposed approach in R and OCG has provided an easy-to-use R package. Theoretical details of the OCG algorithm [73] are described in the following part.

3.3.2.1 Modularity definition

Newman et al. [102] proposed a measure of modularity to evaluate the overall modular structure of a network, which quantifies the excess of within-group edges relative to the number of edges expected for a random partition into node groups having the same number of members.

Let $G = (V, E)$ be a simple connected graph with n vertices and m edges ($|V| = n, |E| = m$). Given a strict partition P of G, in which V are divided into p non-overlapped groups: $P = V_1, V_2, ..., V_p$, let e_{ij} be the percentage of edges having one end in group V_i and the other in group V_j: $e_{ij} = |E \cap (V_i \times V_j)|/m$. The probability for a random edge to have one end in V_i is equal to:

$$a_i = e_{ii} + \frac{\sum_{j \neq i} e_{ij}}{2} \tag{3.2}$$

The modularity M of partition P is defined as:

$$M(P) = \sum_{i=1}^{p} (e_{ii} - a_i^2) \tag{3.3}$$

An equivalent measure has been proposed to extend the modularity M to a partition with overlapped node groups [103]. To avoid the confusion between non-overlapped partition and overlapped partition, we refer to an overlapped partition as a cover of G. Let R be a cover of G, defined by a binary relation α: $V \times V \rightarrow \{0, 1\}$, where $\alpha_{ij} = 1$ if both node i and node j belong at least once to a common group and 0 otherwise. The modularity Q of cover R is defined as:

$$Q(R) = \sum_{i \neq j} (2mA_{ij} - d_i d_j)\alpha_{ij} \tag{3.4}$$

where d_i and d_j are the degrees of i and j in G and A is its incidence matrix ($A_{ij} = 1$, if $(i, j) \in E$).

3.3.2.2 Module detection

The module detection method OCG aims to find the optimal cover R, that is, to find the matrix α_{ij}. OCG starts with an initial cover of the network and then iteratively merges the gene groups according to a greedy strategy step by step.

The initial cover of the network is built based on three optional overlapped group systems:

- Edges: The initial cover consists of all edges of the network, where each edge is considered as a group. The modularity of the initial cover is maximal and the merging process starts by establishing cliques. The modularity function increases as long as there is at least one edge connected two groups.

- Maximal cliques: The initial cover is built with list of the maximal cliques calculated from the network, which cannot be further enlarged given its neighbours. The maximal modularity Q_{max} of overlapped group system has been obtained. Any group fusion will contribute to Q decrease until $Q_{min} = \sum_{i=i}^{n} d_i^2$.

- Centred cliques: For each node $i \in G$, a clique is built using a greedy polynomial algorithm [73]. As long as a clique is produced, nodes adjacent to i are added in decreasing order of their relative degree. The resulting clique, containing i, is not necessarily maximal because a larger one containing i could exist.

At each merging step, the joined two groups are those provide the highest modularity average gains which is defined as the modularity gain divided by the number of newly joined vertex pairs:

$$\Delta Q = (Q(R_m) - Q(R_0))/(\sum \alpha(R_m) - \sum \alpha(R_0)) \tag{3.5}$$

where R_0 is the cover of G before the group fusion, and R_m is the cover after group fusion.

The merging process is stopped when reaching the expected number of modules, or reaching the maximal size of modules, or reaching no gains to increase ΔQ, which depends on the initial cover system.

At the end, a filtering step is added to refine the modules. The contribution of each node to the modularity of the modules is measured. When negative, the node is transferred to the module where its contribution is the highest. This refinement permits eliminating loosely assigned elements and further improving the modularity value.

3.3.3 Identification of condition-associated modules

In this step, we aim to address the third subtask of identifying condition-associated modules which are significantly related with all classes of conditions across all samples. The modules

produced by the second step are inferred from the co-expression network based on the topological characteristics using OCG algorithm. In this step, we identify modules with potential biological insights by evaluating their associations with embryo development. The association of each module is evaluated by the over-representation of genes associated with the overall embryo development stages. We apply gene set enrichment analysis (GSEA) [104], a most widely used method, to achieve this task. We choose GSEA because it takes into account not only the numbers of interested genes but also the magnitudes of gene expressions across all the development stages.

GSEA tests for enrichment of a predefined gene set S among the N background genes. GSEA first assesses the correlation of each gene with the experiment conditions of all samples, and ranks the background genes according to their correlations. Based on the ranked background gene list, it calculates an enrichment score (ES) for the gene set S by evaluating its over-representation of the correlations among the background genes. The significances for the over-representation of the modules are evaluated by calculating the empirical p-values using permutation strategy.

In this step, each module is considered as a predefined gene set S, and the background genes are the total genes in the co-expression network. GSEA was originally developed to work with bi-class gene expression data which contains only two types of experiment conditions. But in this study, the experiment conditions (i.e., development stages) are multiple, we therefore modify the original GSEA method by incorporating the statistical method analysis of variance (ANOVA) to suit the gene expression data with multiple development stages in this study. The modified GSEA method is referred to as ANOVA-GSEA hereafter. Details of ANOVA-GSEA are described as follows:

3.3.3.1 Rank background gene list

Suppose there are p genes in the co-expression gene network. Based on their expression abundances, the correlation of each gene with the development stages of all samples (i.e., embryos) is evaluated by one-way ANOVA F-test. For each gene, ANOVA assesses whether the expected expression abundances of a gene within a stage class differ from other classes of stages. If there are differences between the four classes of development stages (oocyte, 4-cell, 8-cell, blastocyst), it suggests that the gene is associated with embryo development because it is regulated between different stages during embryo development. The p background genes are ranked according to the F-statistics assessed by ANOVA. Larger values of F-statistics represent stronger correlations of the genes with embryo development.

3.3.3.2 Calculate enrichment score

Let L be the ranked background gene list of the p genes. As suggested by GSEA, an enrichment score ES is calculated to evaluate the degree to which a module is over-represented at the extremes of the ranked gene list L. For a given module S that contains s genes, the enrichment score $ES(S)$ evaluates the fraction of genes in S ("hits") weighted by their F-statistics and the fraction of genes not in S ("misses") present up to a given position i in L as follows:

$$\begin{aligned} P_{hit}(S,i) &= \sum_{\substack{j \in S \\ j \leq i}} \left(|F_j| / \sum_{j \in S} |F_j| \right) \\ P_{miss}(S,i) &= \sum_{\substack{j \notin S \\ j \leq i}} \frac{1}{p-s} \end{aligned} \tag{3.6}$$

Then, $ES(S)$ is the maximum deviation from zero, $ES(S) = max_i |P_{hit}(S,i) - P_{miss}(S,i)|$, which depends on both the weights of the correlations and the positions of the genes in S relative to all of the genes in L.

3.3.3.3 Assess the significance of modules

For each module, the significance of the observed score ES is assessed by comparing it with the set of scores ES_{NULL} computed with the permutations of experiment conditions for the gene expression data. Specifically, the original stage labels are assigned randomly to embryos. Then the background genes are sorted based on correlations to the permuted labels, and $ES(S)$ is re-computed. The permutation step is repeated 100 times to create a null distribution of enrichment scores ES_{NULL}. The empirical p-value of a module is estimated from ES_{NULL} using the positive or negative portion of the distribution corresponding to the sign of the observed ES.

By setting a cutoff for the p-value, the significant modules with p-values lower than the cutoff are identified as associated with human pre-implantation embryonic development. The significance suggests general correlation between a module and the overall development stages, but the specific correlation to a certain stage is not implicated. Consequently, in the next step, we aim to identify the modules associated with a specific stage, referred to as condition-specific modules, from these significant modules.

3.3.4 Selection of condition-specific modules and genes

Through the first three steps, we address the first three subtasks and identify gene functional modules significantly associated with human early embryonic development. Since this is a multi-class condition gene expression dataset which includes four embryonic development

stages, we aim to further select key modules and key genes within the modules that are associated with each specific stage. This subtask can be addresses by applying the bi-level feature selection techniques such as regularized linear regression. As introduced in Chapter 2.3.1.4, Group Exponential Lasso (GEL) [49] is a common regularized linear regression method which performs bi-level feature selection. It is capable of group selection as well as important predictor selection from feature groups. By applying GEL regression method, we can select the condition-specific modules as well as the key genes within the modules. Because the significant modules identified from the third step are overlapping, the GEL regression method can not be applied directly. Besides, the multi-class condition is a multinomial response variable, so the feature gene and module selection can not be achieved by linear regression but by logistic regression instead. There is not well-established method to perform logistic regression with GEL regularization on overlapping feature groups, we therefore propose a regularized logistic regression framework to select modules that are associated with a specific condition (i.e., condition-specific modules) from the significant modules which are identified from last step.

The proposed logistic regularization regression framework performs group variable selection on the significant modules with the conditions as the response variable. Each gene is considered as a predictor variable and each module is considered as a group. Group Exponential Lasso (GEL) penalty [49] is used for the regularization. Thus, we refer to the proposed framework as GEL logistic regression. The GEL penalty provides a solution of bi-level sparsity on regression coefficients. The GEL logistic regression is therefore capable of selecting both the modules and the important genes within the modules. The GEL logistic regression is implemented as follows.

3.3.4.1 GEL logistic regression

The vector of response variable y is a multinomial variable with multiple conditions, e.g., the embryonic development stages (oocyte, 4-cell, 8-cell and blastocyst). Binomial logistic regression is performed for each condition to select the condition-specific modules by transforming y into a binary vector. For example, the binary response vector for a specific condition, e.g. the oocyte stage, is defined as:

$$y_{oocyte} = \begin{cases} 1 & \text{if } y_i = \text{oocyte}, \, i = 1, ..., n \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

Given a binary response vector y of a condition, a binomial logistic regression model describes the relationship between the log odds of probability of "success" response and the predictors. Let $p_i = \Pr(y_i = 1|x_i)$ be the probability of "success" of y_i given the

predictors $x_i = (x_{i1}, ..., x_{ip}), i = 1, ..., n$, the logistic model is given by:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + x_i\beta \tag{3.8}$$

where β_0 is the intercept, and $\beta = (\beta_1, ..., \beta_p)^T$ is the vector of coefficients for the p predictors. After the transformation, p_i is given by:

$$p_i = \frac{e^{(\beta_0 + x_i\beta)}}{1 + e^{(\beta_0 + x_i\beta)}} = \frac{1}{1 + e^{-(\beta_0 + x_i\beta)}} \tag{3.9}$$

$$e^{(\beta_0 + x_i\beta)} = \frac{p_i}{1-p_i} \tag{3.10}$$

Logistic regression estimates the coefficients using maximum likelihood estimation (MLE) by maximizing the log-likelihood:

$$\begin{aligned} LL(\beta) &= \sum_{i=1}^{n} y_i \log p_i + \sum_{i=1}^{n} (1-y_i)\log(1-p_i) \\ &= \sum_{i=1}^{n} \left\{ y_i(\beta_0 + x_i\beta) - \log(1 + e^{(\beta_0 + x_i\beta)}) \right\} \end{aligned} \tag{3.11}$$

In order to achieve the goal of group variable selection, the coefficients are estimated by the regularized regression which adds the GEL penalty as well as the Ridge penalty to the negative log-likelihood. The former provides the bi-level sparsity and the latter provides the smoothness for the coefficients. The penalty function is defined as:

$$P(\beta) = \alpha \sum_{l=1}^{L} \left(\frac{\lambda^2}{\theta} \left\{ 1 - \exp\left(-\frac{\theta ||\beta_l||_1}{\lambda} \right) \right\} \right) + (1-\alpha) \sum_{j=1}^{p} \beta_j^2 \tag{3.12}$$

where, λ and θ are the regularization parameters and α is the tuning parameter ranging from 0 to 1, which controls the trade-off between the two penalty terms. The estimator $\hat{\beta}$ is given by:

$$\hat{\beta} = \arg\min_{\beta}(-LL(\beta) + P(\beta)) \tag{3.13}$$

To overcome the overlapping in groups for regularized logistic regression, the original coefficients $\beta = (\beta_1, ..., \beta_p)$ for the p predictors is decomposed according to the overlapping groups. For example, suppose that there are four predictors x_1, x_2, x_3, x_4 belonging to three groups S, where $S_1 = \{x_1, x_2\}, S_2 = \{x_2, x_3\}, S_3 = \{x_1, x_3, x_4\}$. Since the predictor x_1 is belonging to both group 1 and group 3, β_1 is thus decomposed into $\beta_{11} + \beta_{13}$. Consequently, the original $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)$ is decomposed as $\tilde{\beta} = (\beta_{11}, \beta_{13}, \beta_{21}, \beta_{22}, \beta_{32}, \beta_{33}, \beta_4)$. Correspondingly, the predictors in each observation are also duplicated to match $\tilde{\beta}$ as

$\tilde{x}_i = (x_{i1}, x_{i1}, x_{i2}, x_{i2}, x_{i3}, x_{i3}, x_{i4}), i = 1, ..., n$. Based on the coefficient decomposition, the aim becomes to estimate $\tilde{\beta}$. The decomposed coefficients $\tilde{\beta}$ of L groups are estimated by regularized regression using the GEL penalty and Ridge penalty by:

$$\begin{aligned} &\text{minimize } \left\{ -\sum_{i=1}^{n} \left(y_i(\beta_0 + \tilde{x}_i\tilde{\beta}) - \log(1 + e^{(\beta_0 + \tilde{x}_i\tilde{\beta})}) \right) \right\} \\ &\text{subject to } \left\{ \alpha \sum_{l=1}^{L} \left(\frac{\lambda^2}{\theta} \left\{ 1 - \exp\left(-\frac{\theta ||\tilde{\beta}_l||_1}{\lambda} \right) \right\} \right) + (1-\alpha) \sum_{j=1}^{\tilde{p}} \tilde{\beta}_j^2 \right\} \leq t \end{aligned} \tag{3.14}$$

3.3.4.2 Tuning parameters for coefficient estimation

The parameters $(\alpha, \lambda, \theta)$ for the best fitted logistic regression model are selected by evaluating the prediction errors using K-fold cross-validation (see details in Chapter 2.3.5). The prediction output $\hat{y}_i$ of an estimated logistic regression model is a probability of "success". By setting a cutoff of 0.5 for the probability, $\hat{y}_i$ can be classified as either "success" or "failure" as:

$$\hat{y}_i = \begin{cases} 1 & \text{if } \hat{y}_i > 0.5 \\ 0 & \text{if } \hat{y}_i \leq 0.5 \end{cases} \tag{3.15}$$

Thus, the prediction error is defined as the percentage of misclassified samples out of all samples:

$$PE = \frac{\#\text{ of incorrectly classified samples}}{\text{total sample size}} \tag{3.16}$$

The parameters $(\alpha, \lambda, \theta)$ that result in the minimum PE are chosen as the optimal parameters, which are used to fit the best model on the whole data. The coefficients estimated based on this best fitting model are the final results for variable selection.

For each condition, the coefficients are estimated based on the corresponding best fitting model. The genes with non-zero β are selected as condition-specific genes which are associated with the condition, and their corresponding modules are selected as the condition-specific modules. In the embryonic development study, the stage-specific genes and modules are selected for each development stage from the stage-associated modules identified in the third step using the proposed GEL logistic regression.

3.4 Results

3.4.1 Embryonic development stage-specific modules

By applying the proposed approach to the transcriptome data of human embryos, a gene co-expression network across human pre-implantation embryonic development is constructed, which consists of 116124 edges among 11269 genes. The degree of the network

follows power-law distribution, $P(d) = cd^{-\alpha}$, with estimated $\alpha = 2.09$, which suggests that the co-expression network is a scale-free network. Following the proposed approach, 2531 overlapping modules are detected from the network. There are 347 significant modules selected as embryonic development stage-associated modules by ANOVA-GSEA. Using GEL logistic regression with 5-fold cross-validation, 42 modules are identified as embryonic development stage-specific modules which contain at least one stage-specific genes within the module. The largest module consists of 42 genes, while the smallest one includes five genes (defined by arbitrary cutoff). Table 3.1 summarizes the numbers of stage-specific genes and modules corresponding to each embryonic development stage.

Table 3.1: Numbers of embryonic development stage-specific genes and modules.

Stage	# genes	# modules
Oocyte	25	14
4-cell embryo	41	16
8-cell embryo	107	17
Blastocyst	81	14

3.4.2 Functionality of stage-specific modules

To assess the functional coherence of the identified stage-specific modules, we perform the Gene Ontology (GO) [105] function enrichment for each module, which is a most popular strategy for functional annotation in the filed of computational biology. GO provides a system of classifications of gene functions, where genes are assigned to a set of predefined terms depending on their functional characteristics. If the genes of a module are significantly enriched in several function terms in GO, it suggests that the module represents a coherent function unit related with those GO function terms. For each stage-specific module, an enrichment analysis is performed to find which GO biological process terms are over-represented for the module.

The enrichment is calculated by hypergeometric test which is a common statistical method used to evaluate the significance of the enrichment. Given a module containing m genes and a GO biological process term with k genes, the module and the GO term are overlapping with n genes. The significance of the representation of the GO term in the module is calculated as:

$$p = P(n|m, k, N) = 1 - \sum_{i=0}^{n-1} \frac{\binom{k}{i}\binom{N-k}{m-i}}{\binom{N}{m}} \tag{3.17}$$

where N is the number of total genes in GO and $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ is the binomial coefficient. To address the multiple statistic testing problem, the p-values are adjusted by BH for the FDR correction [106]. A significant p-value, e.g. $p \leq 0.05$, suggests that the GO term is

over-represented for the module, which indicates the functional coherence for the module. The enriched GO biological process term thus characterizes the functions of the module.

Through the function enrichment analysis, all the 42 stage-specific modules are shown enriched with at list one GO biological processes, which suggests that the modules are coherent functional units related with the corresponding biological processes.

3.4.3 Case study of stage-specific modules

The proposed approach is capable of capturing the dynamic expression patterns of the modules across the multiple development stages. Figure 3.3 shows the case study of a stage-specific module that is identified associated with oocyte, 4-cell embryo, 8-cell embryo and blastocyst stages. This module consists of 10 genes as shown in Figure 3.3A, which is involved in several crucial biological processes during the pre-implantation embryonic development such as "mitotic nuclear envelope disassembly", "viral transcription" and "spliceosomal snRNP assembly" (Figure 3.3B). It is a highly dynamic module that the genes exhibit different expression patterns corresponding to different stages, illustrated in Figure 3.3C-F.

The module successfully captures the dynamic roles of several nucleoporins (Nups) such as NUP50, NUP62, NUP153 and NUP155. Nucleoporins are the key components of the nuclear pore complex (NPC), a large multiprotein assembly embedded within the nuclear envelope that mediates all transport between the nucleus and the cytoplasm [107]. The transport of specific macromolecules across the nuclear envelope mediated by NPC is critical for embryonic development, cell growth and differentiation [108]. The key components of NPC have been reported to play dynamic and diverse roles during embryonic development in many species. In this module, NUP62, NUP153, and NUP155 are selected as oocyte stage-specific genes, which suggests that they might be associated with the early stage of embryonic development. NUP50 is identified as a feature gene associated with the last three stages but not the first one, which suggests that it might be associated with the late stage of embryonic development. We perform curated literature search and find several pieces of evidence for such characteristics:

- Smitherman et al. found that the loss of Nup50 leads to embryonic death during late gestation in mouse [109], which suggests that Nup50 is associated with the late embryonic development stages.
- NUP62 was found to associate directly with a similar set of actively transcribed genes which were predominantly involved in the development and cell cycle in normal drosophila embryonic cells [110, 111].
- Jacinto et al. found that depletion of Nup153 in mouse embryonic stem cells (mESCs) causes the de-repression of developmental genes and induction of early differentiation

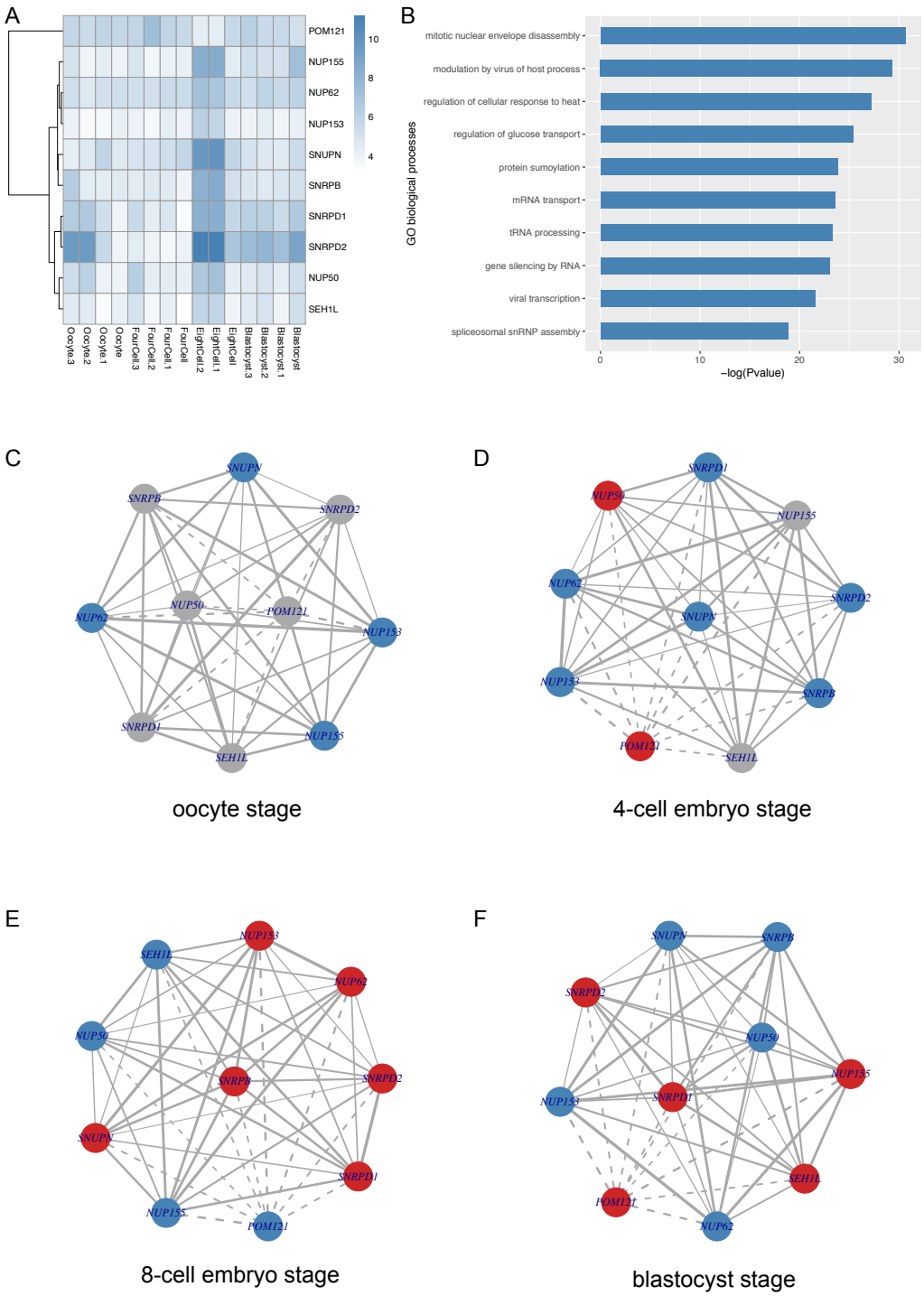


Figure 3.3: Case study of a stage-specific module that is identified associated with oocyte, 4-cell embryo, 8-cell embryo and blastocyst stages. (A) Heatmap of gene expression in the module; (B) Enriched GO biological processes for the module; (C-F) Dynamic expression pattern of stage-specific genes within the module for the oocyte, 4-cell embryo, 8-cell embryo and blastocyst stage, respectively. Nodes represent genes and edges represent co-expressed interactions. Nodes in red represent stage-specific genes positively correlated with the stage and nodes in blue represent stage-specific genes negatively correlated with the stage. Edges in solid lines represent positive co-expression and edges in dashed lines represent negative co-expression.

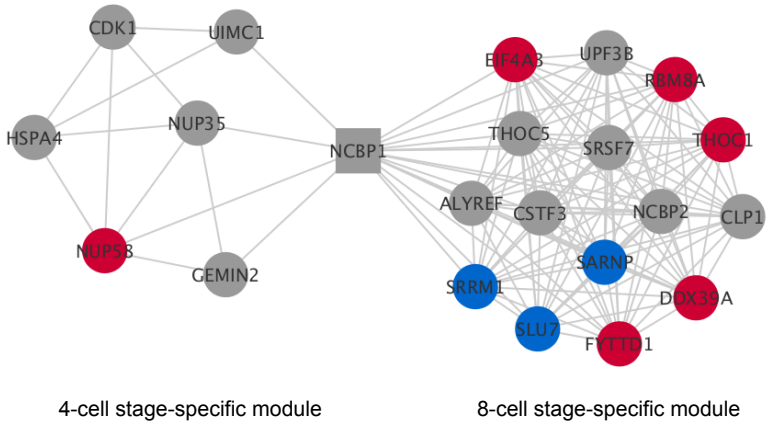


Figure 3.4: An example of multi-function gene NCBP1. Figure legends are the same as Figure 3.3C-F.

[112], which suggests that Nup153 positively regulated the developmental genes in mouse embryos. In agreement with their discovery, NUP153 is identified negatively correlated with the first two stages (oocyte and 4-cell embryo), but exhibits an activation in the subsequent 8-cell embryo stage. Scientific evidence shows that the activation of human embryo genome happens at 8-cell stage [113], in which many developmental genes are activated during this process. As a positive regulator of many developmental genes, the activation of NUP153 at 8-cell stage will result in the activations of these genes.

- NUP155 has been found to be required in early embryo in vertebrate [114]. Moreover, Franz et al. found that in the early stage embryos of C. elegans, depletion of Nup155 caused 100% embryonic lethality and the separation of the pronuclei from the centrosomes [115].

Besides, scientific evidence has shown that SNRPD2 and SEH1L are related with late stage of embryonic development. SNRPD2 has been found up-regulated during the activation of human embryo genome at the 8-cell stage [116], which is successfully captured by the dynamic module showing that SNRPD2 is positively correlated with the 8-cell stage and the blastocyst stage but not the first two stages. SEH1L has been reported as a trophoblast-specific genes which is related with the late development of embryos [117]. In the identified module, SEH1L shows positive correlation only with the blastocyst stage.

The proposed approach allows overlapping between modules, which is capable to capture the multi-function genes. In practice, genes usually have complex roles in biological processes that a gene might involve in multiple biological processes. Identifying these multi-function genes will help understand the coordination between biological processes. Figure 3.4 gives an example of a multi-function gene NCBP1 which belongs to two modules

corresponding to 4-cell stage and 8-cell stage respectively. NCBP1 is located between the two modules as a bridge, which suggests that it might have the potential role during the 4-cell to 8-cell embryonic development. The gene EIF4A3 is a proved marker of the human 8-cell embryo [113], the co-expressed functional link between NCBP1 and EIF4A3 also suggests the importance of NCBP1 during embryonic development.

3.4.4 Reproducibility of stage-associated modules

Because module detection is an unsupervised learning task, it is difficult to assess the accuracy of the results as there are not prior labels for the gene functional modules. However, in order to evaluate whether the embryonic development associated modules are reproducible using the other module detection methods, we perform a comparative study for embryonic development associated module detection on different datasets using several module detection methods as follows:

- OCG: module detection using the proposed approach on our in-house generated data.
- MCL: module detection using Markov Cluster Algorithm (MCL) [118] on our in-house generated data.
- ModuLand: module detection using ModuLand [119] on our in-house generated data.
- WGCNA: module detection using Weighted Gene Co-expression Network Analysis (WGCNA) [77] on our in-house generated data.
- MCL_seq: module detection using MCL on a published RNA-seq data of human pre-implantation embryo transcriptomes [120].

Table 3.2: Overlapping of intra-module genes between different module detection methods. The diagonal shows the numbers of total intra-module genes identified by each method; the lower triangular table shows the numbers of overlapping genes between the corresponding methods; the upper triangular table shows the significance of the numbers of overlapping genes assessed by hypergeometric test.

Method	OCG	MCL	ModuLand	WGCNA	MCL_seq
OCG	871	9.96E-07	7.47E-45	1.21E-07	6.52E-16
MCL	103	1072	5.62E-258	6.56E-21	2.74E-05
ModuLand	550	950	5837	4.73E-50	1.62E-03
WGCNA	33	59	181	201	1.24E-09
MCL_seq	279	276	1284	79	3012

The overlapping of intra-module genes, which are the genes within all embryonic development associated modules, identified by different methods are shown in Table 3.2.

We observe that the overlapping of intra-module genes are significant between all the methods. The proposed approach has more overlapping intra-module genes with MCL and ModuLand than WGCNA. The reason might be that MCL and ModuLand incorporate the gene-gene interaction network for module detection similar as the proposed approach, but WGCNA identifies co-expression gene modules from the transcriptome data exclusively. Such differences in the overlapping of intra-module genes suggest that the interactome data might introduce intrinsic information of gene functional patterns.

3.5 Implementation

The data sources of the transcriptome and interactome data studied in this chapter are described in Chapter 3.2.1. We implement the proposed approach in R to study human embryonic development by integrating the transcriptome and interactome data. The processed datasets used in this study and the R codes for implementing all the four steps of the proposed approach can be accessed from `https://github.com/bioinfoxh/condition-specific_module_detection`.

3.6 Summary

In this chapter, we propose an approach for gene module detection by integrating multi-condition transcriptome data and interactome data using network overlapping module detection method, which consists of four steps: (1) construction of gene co-expression network by evaluating co-expression correlation coefficient between each interacted gene pair based on their gene expression; (2) detection of overlapping gene modules from the co-expression network using network overlapping module detection method; (3) identification of condition-associated modules by assessing the significance of enrichment with condition-associated genes within the modules using ANOVA-GSEA; (4) selection of condition-specific feature modules and feature genes using GEL logistic regression with K-fold cross-validation.

We apply the proposed approach to the transcriptome data of human pre-implantation embryos across multiple development stages and identify human embryonic development stage-specific modules and genes. Interesting biological insights are revealed from the dynamic expression patterns of the stage-specific modules and the multiple function genes located in the overlapping modules, which provides clues for understanding the potential molecular mechanisms during human pre-implantation embryonic development. To assess the stability of the modules identified by the proposed approach, we perform similar module detection studies using several common module detection methods as well as on different transcriptome data. We find that the intra-module genes are significantly

overlapped between different methods and datasets.

The proposed approach provides an efficient computational pipeline based on network clustering for transcriptome and interactome data integration in the field of multi-omics. It extends the state-of-art approaches by providing a comprehensive and flexible computational framework, which is capable of identifying both condition-specific modules and intra-module condition-specific genes by integrating multi-class condition transcriptome data and interactome data. We believe the proposed approach is a useful tool for the field of multi-omics, which helps researchers to mine the underlying molecular mechanisms through integration of transcriptome and interactome data.

This approach could be further improved in two directions. Firstly, the proposed approach only works with unweighted co-expression network. But in practice, gene co-expression network is a weighted network, where the edge weights indicate the strength of co-expression correlations between the interacted genes in the network. A larger weight usually suggests stronger co-functional roles between the genes. Thus, the edge weights provide important functional information. Taking into account such information will lead to more coherent and dynamic functional modules detected from a weighted co-expression network. This issue can be addressed by employing effective network module detection algorithms which are capable of extracting modules from weighted network. Secondly, the proposed approach offers advantage of detecting overlapping modules which is capable of capturing genes with multiple functions. However, it will result in strong redundancy in the detected modules, which might bring confusions for interpreting the modules. This limitation can be solved by simply deleting redundant modules in the results or improving the module detection algorithm by controlling the overlapping rate between modules during the module search procedure.

Chapter 4

Multilayer network module detection for transcriptome, translatome and interactome integration

4.1 Introduction

Regulation of gene expression occurs not only at transcription level but also at translation level. Translational regulation has been proved to play a crucial role in essential biological processes [121, 122]. Aberrant regulation of translation has been shown to be involved in the genesis of many human diseases, which may be potential targets for disease therapy [123, 124]. Compared with the understanding of transcriptional regulation, our knowledge on translational control of gene expression is relatively limited.

The development of high-throughput Ribosome Profiling technology (Ribo-seq) [25] has provided great opportunities for exploring translational mechanisms of gene expression. Ribo-seq experiment is usually accompanied by RNA-seq of the matched biological sample, which reveals the genome-wide translation efficiency by comparing the ribosome protected fragments (RPFs) from translatome with the mRNAs from transcriptome [125, 126]. Alterations in translation efficiency, that is, the changes in RPF abundances between two conditions are discordant with the changes in mRNA abundances, suggests the underlying regulation of translation. To characterize the potential mechanisms associated with such translational control, the main challenges involve two tasks: identification of differentially translated gene (DTG) whose difference in RPF abundances cannot be explained by the difference in corresponding mRNA abundances and detection of functional patterns that are associated with the differential translation. Several methods have been proposed to address the first problem, including Babel [127], Xtail [128] and Riborex [129]. However, no tools have been developed for the second purpose yet and it is therefore still a challenging task to mine the underlying functional patterns from transcriptome and translatome data.

Recently, the multilayer network framework has been successfully used to characterize responsive functional patterns in complex biological systems, e.g. inference of epigenetic functional modules from a multiple network constructed based on gene expression and DNA methylation data [39], detection of dynamic pathways on multiple co-expression gene networks during multi-stage progression of diseases [130], and identification of cancer driver genes via community detection from multilayer networks built by integrating multi-omic data [40], which shows that the multilayer network framework is more efficient in capturing characteristics of biological patterns from multi-omic data compared with the aggregated single-layer network. Therefore, multilayer network would be potential useful for integrating the transcriptome and translatome data with protein-protein interactome data to explore the functional patterns such as gene modules which are associated with translation regulation mechanisms. Efficient methods and tools are in urgent need to achieve the goal.

4.2 Multi-omics

4.2.1 Multi-omic data

4.2.1.1 Transcriptome and translatome

The transcriptome and translatome datasets studied in this chapter are derived from a published ribosome profiling data of human prostate cancer cell lines PC3 in response to mTOR signalling perturbation [131]. mTOR, the mammalian target of rapamycin kinase, is a master regulator of protein synthesis that couples nutrient sensing to cell growth and cancer. To explore the crucial role of mTOR in prostate cancer, Hsieh et al. [131] performed a genome-wide ribosome profiling of PC3 cells with the treatment of PP242, an inhibitor to mTOR, to study the downstream translational regulation mechanisms that result in prostate cancer development. The matched transcriptome and translatome data are measured by RNA-seq and Ribo-seq on two original PC3 cells and two PP242-treated PC3 cells. In this study, we use the processed RNA-seq data and Ribo-seq data (i.e., the mRNA read counts and the RPF read counts) downloaded from Xiao et al. [128].

4.2.1.2 Interactome

Human interactome data used in this chapter are the curated protein-protein physical interactions downloaded from STRING database (v10.5) [132]. The proteins are mapped to corresponding genes for the integration with transcriptome and translatome data. Redundant interactions and self-interactions are removed from the interactome data.

4.2.2 Problem definition

The transcriptome and translatome reflect the genome-wide responses of translation efficiency to the mTOR perturbation by capturing the transcription level (mRNA) expression and the translation level (RPF) expression. On the basis of the network introduced by the interactome data, the alterations in mRNA and RPF expression, can be exhibited in network structure respectively, which form a multilayer network consisting of two layer networks corresponding to the transcription level and the translation level respectively. A set of genes, whose expression pattern is discordant between the two layers, are defined as a translation efficiency (TE) regulated module. To identify such TE-regulated modules, the problem can be considered as an unsupervised learning task to detect the functional gene modules from the multilayer network. A functional gene module in a multilayer network can be defined as a set of genes whose connectivity within them is stronger than the outgoing connectivity across all layers of networks. On the basis of such definition, the problem can be solved by efficient methods for multilayer network module detection.

4.3 Methodology

Since the research problem of this chapter is defined as a task of inferring gene modules from multilayer biological networks, it can be solved by using multilayer network module detection algorithms. Multilayer network is an emerging subfield in the filed of network science, but it has been successfully applied to biological studies, including but not limited to, inferring gene modules from multilayer networks by integrating multi-omic data. The multilayer network module detection algorithms developed for multi-omic study can be classified into two categories:

- For the first category, the strategy of the algorithms is to perform module detection in each layer of the multilayer network separately and then merge the signal layer modules together. A state-of-art algorithm of this category is the consensus clustering algorithm [40]. Cantini et al. proposed the consensus clustering algorithm to identify cancer related gene modules by integrating different layers of genomic information including transcription factor co-targeting, microRNA co-targeting, protein-protein interaction and gene co-expression networks [40]. The consensus clustering algorithm first applies state-of-art community detection methods in each layer to get single layer gene modules and then uses consensus clustering algorithm to merge the single layer modules in an iterative way until the consensus modules converge.

- For the other category, the strategy of the algorithms is to perform module detection by integrating the information across all layers of the multilayer network when searching for modules. A state-of-art algorithm of this category is the M-module

algorithm [130]. Ma et al. proposed a clustering algorithm, M-module, to identify modules from multilayer weighted networks [130]. They define a multilayer network module, named M-module, as a group of nodes whose within-group connectivity is stronger than the between-group connectivity across all layer networks. The algorithm performs greedy search to infer an M-module from the multilayer network using seed expansion strategy by optimising its modularity quantified by graph entropy. The M-module algorithm has been successfully applied to identify gene module associated with disease progression from multiple transcriptome data [130] and to identify epigenetic regulated gene modules from DNA methylome and transcriptome data [39].

Although the consensus clustering algorithm and the M-module algorithm are both efficient for multi-omic studies, there are still some limitations for the former. In the consensus clustering algorithm, the fundamental modules are identified from single layer network separately. It assumes that the different layers are independent between each other and ignores the between-layer relationships during the module detection. But in real multi-omic studies, different layers of the multilayer network are not independent but might crosstalk to each other, e.g., DNA methylation plays regulatory roles on gene expression. On the contrary, the M-module algorithm integrates the information across all layers of the multilayer network during module inference, which takes into account the potential between-layer impacts.

In this chapter, the goal is to identify translational regulation related gene modules from the multilayer network constructed by integrating the mRNA expression measured by RNA-seq and the RPF expression measured by Ribo-seq. The mRNA layer and the RPF layer are correlated because there are critical translational regulation mechanisms for translating mRNAs to proteins through the ribosome protected fragments (RPFs). Considering such correlated relationship will help to mine the potential regulation mechanisms from the transcriptome and translatome data. Consequently, the M-module algorithm is the more appropriate solution to achieve the goal of this chapter.

We propose an approach, based on the M-module algorithm, to identify translation efficiency regulated (TE-regulated) modules from the multilayer differential expression network constructed by integrating transcriptome, translatome and interactome data, which consists of five steps, illustrated in the flowchart in Figure 4.1: (1) construction of multilayer network, (2) selection of seed genes, (3) greedy search for modules, (4) refinement of modules, and (5) visualization of modules.

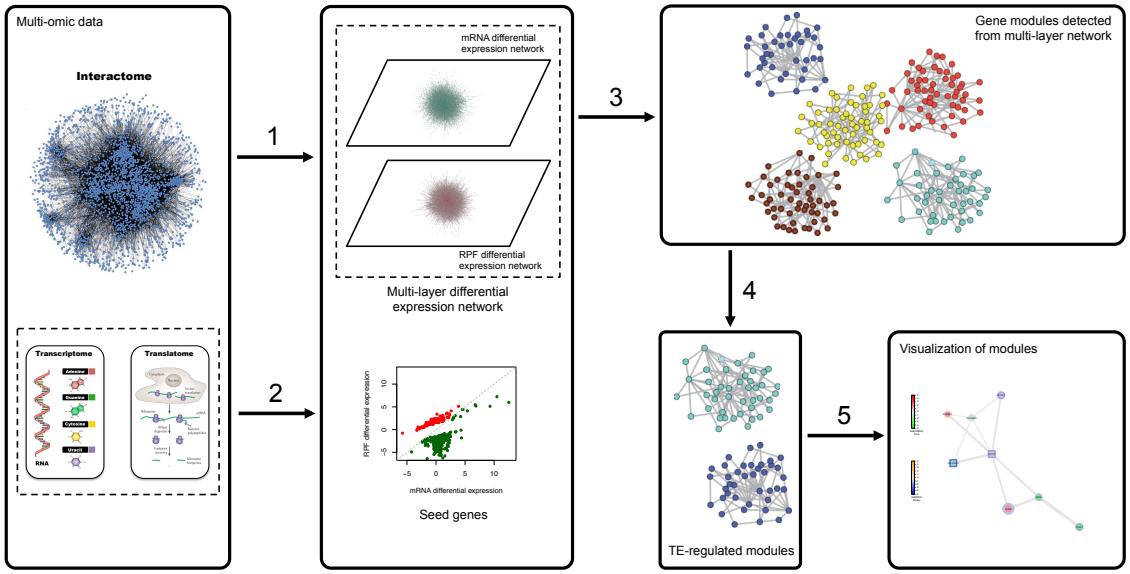


Figure 4.1: Flowchart of proposed approach to multilayer network module detection for transcriptome, translatome and interactome integration. It consists of five steps: (1) construction of multilayer differential expression network by integrating transcriptome and translatome with interactome data respectively; (2) selection of seed genes for module detection by evaluating their degrees of differential translation; (3) detection of modules from the multilayer network using greedy search for each seed gene by minimizing the entropy-based local modularity function; (4) identification of translation efficiency (TE) regulated modules by the refinements including significance assessment, redundancy deletion and dynamic evaluation; (5) visualization of TE-regulated modules as graphs with incorporated multilayer information from the networks.

4.3.1 Construction of multilayer network

In this approach, we use the multilayer network framework to integrate the transcriptome and translatome data from the same ribosome profiling data. We build a multilayer network which contains two layers: one layer representing the the transcription level (mRNA) expression derived from the transcriptome data and the other representing the translation level (RPF) derived from the translatome data. In the following part of this section, we describe the details for constructing the multilayer network.

The statistics t-values and p-values of mRNA differential expression between two conditions for each gene are obtained by use of negative binomial generalized linear models proposed by DESeq2 [133] which is a common tool for RNA-seq data analysis. Following the statistical models proposed by DESeq2, for the gene i in sample j, its mRNA read (a fragment of the mRNA sequence) count m_{ij} is modelled following a negative binomial

distribution with fitted mean μ_{ij} and a gene-specific dispersion parameter α_i:

$$\begin{aligned} m_{ij} &\sim \mathrm{NB}(\mu_{ij}, \alpha_i) \\ \mu_{ij} &= s_j q_{ij} \\ \log_2(q_{ij}) &= x_j \beta_i \end{aligned} \tag{4.1}$$

where s_j denotes a sample-specific size factor and q_{ij} is a parameter proportional to the expected true abundance of mRNA reads for sample j. The coefficient β_i gives the log2 transformed fold changes for gene i, and the vector x_j indicates the treatment conditions for sample j. The expected read counts follows:

$$\log(\mathrm{E}(m_{ij})) = x_j \beta_i + \log(s_j) \tag{4.2}$$

Hypothesis testing is done to test whether the expression levels differ between conditions. Let β_{g1} and β_{g2} indicating gene-specific coefficients for case and control groups respectively, and the differential expression is evaluated by testing:

$$\begin{aligned} H0 &: \beta_{g1} = \beta_{g2} \\ H1 &: \beta_{g1} \neq \beta_{g2} \end{aligned}$$

On the basis of protein-protein interaction (PPI) network provided by the interactome data, the mRNA differential expression network is constructed by assigning a weight to each edge using the t-values of mRNAs. Specifically, for a given edge $E(i, j)$ that connects gene i and gene j, the weight w_{ij} evaluating the mRNA differential expression on $E(i, j)$ is estimated by the squared harmonic mean of t_i and t_j. The harmonic mean assigns higher weights to gene pairs with comparable t-values with larger absolute values because it tends to mitigate the impact of the larger t and aggravate the impact of the other smaller t. The weight w_{ij} is calculated as:

$$w_{ij} = \sqrt{max(|t_i|, |t_j|) * \frac{2min(|t_i|, |t_j|)}{|t_i| + |t_j|}} \tag{4.3}$$

where t_i and t_j are the t-values of the mRNA differential expression for gene i and gene j based on the transcriptome data.

The RPF differential expression network is developed in the same way using the translatome data.

Consequently, these two networks, the mRNA differential expression network and the RPF differential expression network, form a multilayer network.

4.3.2 Selection of seed genes

Genes in the multilayer network are ranked according to the degrees of their differential translation. Given a gene, if its difference in RPF abundances is not concordant with the difference in mRNA abundances, it is referred to as a differentially translated gene (DTG). The top-ranked DTGs are used as the seeds for module inference from the multilayer network.

Several methods have been published recently for evaluating degrees of differential translation of genes to identify significant differentially translated genes (DTGs) using ribosome profiling data. There are three recent common methods, Babel [127], Xtail [128] and Riborex [129], all of which provide public R packages for users. We compare the performances of the three methods, aiming to select an efficient one to be used in our approach to select the seeds (i.e., top-ranked DTGs). We apply the three methods, Babel, Xtail and Riborex, respectively, on the published ribosome profiling data that are mentioned in Chapter 4.2.1.1. Table 4.1 shows the numbers of significant differentially translated genes identified from the total 10559 genes by the three methods and the overlaps between them. The comparisons between Babel, Xtail and Riborex reveal that the significant DTGs identified by the three methods significantly overlap between each other. The implementations of Babel, Xtail, Riborex take 8376, 341, 7 seconds, respectively, on the same data using a single core of a MacBook Pro with 2.9 GHz Intel Core i5 and 16GB 1867 MHz DDR3 memory. Through these comparisons, we find that Babel, Xtail and Riborex provide comparable performances for the identification of DTGs, but Riborex provides the superior performance in terms of the implementation time. Therefore, we employ Riborex in our approach.

Table 4.1: Overlapping of significant differentially translated genes (DTGs) identified by Babel, Xtail and Riborex. The diagonal shows the numbers of total significant DTGs identified by each method; the lower triangular table shows the numbers of overlapping DTGs between the corresponding methods; the upper triangular table shows the significance of the numbers of overlapping genes assessed by hypergeometric test.

Method	Babel	Xtail	Riborex
Babel	540	1e-16	1e-16
Xtail	275	527	1e-16
Riborex	263	392	465

In the following part of this section, we introduce the statistical model proposed by Riborex that we use to select top-ranked DTGs. We assume a matched transcriptome and translatome dataset which includes k samples and n genes. For gene i in sample j, its mRNA read count m_{ij} is modelled following a negative binomial distribution with fitted

mean μ_{ij} and a gene-specific dispersion parameter α_i:

$$\begin{aligned} m_{ij} &\sim \text{NB}(\mu_{ij}, \alpha_i) \\ \mu_{ij} &= s_j q_{ij} \\ \log_2(q_{ij}) &= x_j \beta_i \end{aligned} \tag{4.4}$$

where s_j denotes a sample-specific size factor and q_{ij} is a parameter proportional to the expected true abundance of mRNA reads for sample j. The coefficient β_i gives the log2 transformed fold changes for gene i and the vector x_j indicates the treatment conditions for sample j.

Similarly, the RPF read count r_{ij} is modelled with fitted mean π_{ij} and a gene-specific dispersion parameter ϵ_i,

$$\begin{aligned} r_{ij} &\sim \text{NB}(\pi_{ij}, \epsilon_i) \\ \pi_{ij} &= d_j p_{ij} \\ \log_2(p_{ij}) &= x_j \lambda_i \end{aligned} \tag{4.5}$$

where d_j denotes a sample-specific size factor and p_{ij} is a parameter proportional to the expected true abundance of RPF reads for sample j. The coefficient λ_i gives the log2 transformed fold changes for gene i and the vector x_j indicates the treatment conditions for sample j.

The translation efficiency of the gene is defined as the ratio of RPF and mRNA expression levels:

$$t_{ij} = p_{ij}/q_{ij} \tag{4.6}$$

RPF read counts depends on both translation efficiency and mRNA expression level. Taking into account the mRNA level while modelling RPF read counts, the expected RPF read count for gene i in sample j can be modelled as follows:

$$\begin{aligned} \log(E(\pi_{ij})) = \log(p_{ij}) + \log(d_j) &= \log(t_{ij}) + \log(q_{ij}) + \log(d_j) \\ &= x_j \delta_i + x_j \beta_i + \log(d_j) \end{aligned} \tag{4.7}$$

where $\log(t_{ij}) = x_j \delta_i$. The coefficient δ_i gives the differential translation efficiency for gene i and the vector x_j indicates the treatment conditions for sample j.

The coefficient vectors δ_i and β_i can be estimated simultaneously by constructing the design matrix as follows:

$$\begin{aligned} x_j^{\text{mRNA}} &= (x_{j1}, ..., x_{jk}, 0, ..., 0) \\ x_j^{\text{RPF}} &= (x_{j1}, ..., x_{jk}, x_{j1}, ..., x_{jk}) \end{aligned}$$

And the corresponding coefficient vector is reorganized as:

$$\gamma_i = (\beta_{i1}, ..., \beta_{ik}, \delta_{i1}, ..., \delta_{ik})$$

With the modified design matrix, differential expression of genes can be evaluated using the established framework of DESeq2. Hypothesis testing is done on δ_i:

$$H0 : \delta_i = 0$$
$$H1 : \delta_i \neq 0$$

The DTGs are then ordered by the p-values for the use of seed genes.

4.3.3 Greedy search for modules

To infer TE-regulated modules from the multilayer network, we employ the greedy search strategy suggested by the M-module algorithm [130].

Following the M-module algorithm, a multilayer network module, named M-module, is defined as a group of nodes whose within-group connectivity is stronger than the between-group connectivity across all layer networks. The modularity of an M-module in multilayer network is quantified by a measure based on graph entropy, which evaluates the skewness of within-module connectivity versus between-module connectivity.

Let a 3-dimensional matrix $A = (a_{ijk})_{n\times n\times M}$ be the adjacent matrix of an M-layer network $G_k = (V, E_k)(1 \leq k \leq M)$ with the same n nodes but different edges in each layer, where a_{ijk} denotes the weight on the edge $E(i, j)$ in the kth network. For a given node i in an M-module C, let $I_k(i)$ denote the total weight between i and other nodes in C in the kth network G_k:

$$I_k(i) = \sum_{i\neq j, j\in C} a_{ijk} \tag{4.8}$$

Similarly, let $O_k(i)$ denote the total weight between i and nodes outside of C:

$$O_k(i) = \sum_{i\neq j, j\notin C} a_{ijk} \tag{4.9}$$

Then, the connectivity of node i to module C in the network G_k is defined as:

$$H_k(i, C) = -p_{ik}log(p_{ik}) - (1 - p_{ik})log(1 - p_{ik}) \tag{4.10}$$

where $p_{ik} = I_k(i)/(I_k(i) + O_k(i))$. Consequently, the connectivity between i and C across

all M networks is calculated as:

$$H(i,C) = \sum_{k=1}^{M} H_k(i,C) \tag{4.11}$$

The multilayer network modularity MM of module C across all networks is defined as:

$$MM(C) = \sum_{i \in C} H(i,C)/|C| \tag{4.12}$$

Given the MM function, a greedy search is performed starting with a seed to identify the subnetwork within the multilayer network whose MM is locally minimal. Specifically, starting with a seed gene, the module C is expanded by iteratively adding neighbour genes whose addition causes the maximum decrease in the MM function until no decrease is observed. The modularity score MM of the module is calculated by the final addition. The greedy search procedure is implemented for each seed gene and the resulted modules will be assessed by further refinement.

4.3.4 Refinement of modules

4.3.4.1 Selecting modules with significant modularities

The significances of modularities are calculated using a Monte Carlo (MC) randomization procedure, which permutes the node statistics around the multilayer network and recomputes modularities for the modules. The significance of the modularity score MM is assessed by comparing it with the set of scores MM_{NULL} computed with the permutations of statistics t-values of transcriptome and translatome data.

The mRNA and RPF t-values are permuted in the same order for genes and a random multilayer differential expression network is constructed. The modularity score MM for each module is re-calculated based on the random network. The permutation is repeated for 100 times, and 100 permuted MM (pMM) are obtained, which create a null distribution of the scores. Finally, the empirical p-value of a module is calculated by the lower portion of the distribution corresponding to the observed MM.

For the multiple statistic test, the FDR of each module is also calculated based on the above 100 permutations of t-values. For each module, the observed MM and the 100 permuted pMM scores are normalized to the Z-score of the null distribution, that is, minus the means of pMM and divided the standard deviation of pMM. And the corresponding normalized scores are obtained, nMM and $npMM$. Then, the FDR is calculated by controlling the ratio of false positives to the total number of modules attaining a fixed level of significance for nMM and $npMM$. The null distribution is constructed from the $npMM$ of all the modules to compute an FDR value, for a given distribution $nMM = x$.

The FDR is the ratio of the percentage of all permutations with $npMM < x$, divided by the percentage of the modules with nMM, whose $nMM < x$.

The modules that contain more than five genes and pass a significance threshold (e.g., $p \leq 0.05$) are selected as candidate modules.

4.3.4.2 Removing redundant modules

The greedy search procedure for the seed genes might result in strong overlapping between the modules. The overlapping between module C_i and module C_j is measured by the *meetmin* index which has been proven a good measure for evaluating containment by Zaki et al. [100]. The *meetmin* index is calculated as:

$$meetmin(C_i, C_j) = \frac{|C_i \cap C_j|}{min(|C_i|, |C_j|)} \tag{4.13}$$

To reduce the redundancy in the candidate modules, the smaller one of the two overlapping modules whose *meetmin* index is greater than a given threshold (e.g. 50%) is removed. The remaining functional modules are considered as TE-regulated modules, which are associated the translation efficiency regulation.

4.3.4.3 Identifying dynamic modules

To evaluate the dynamic patterns of the TE-regulated modules, we use the module dynamic score (MDS) suggested by Ma et al. [130], to assess the dynamics of a module based on its connectivities in the networks based on weight differences of the module between layers. Given a module C which contains k edges, let $w_m = (w_{m1}, ..., w_{mk})$ and $w_r = (w_{r1}, ..., w_{rk})$ be the weights of the edges in the mRNA differential expression network and the RPF differential expression network, respectively, the MDS of the module is calculated as follows:

$$MDS = \frac{\sqrt{\sum_{i=1}^{k}(w_{mi} - w_{ri})^2}}{k} \tag{4.14}$$

The statistical significance of MDS is computed in the same way as that for the modularity of the module. The empirical p-value and FDR of a MDS is calculated using the same permutation procedures for evaluating the significance of modularity. A higher MDS of a TE-regulated module, the stronger association is suggested with the translation regulation mechanisms.

4.3.5 Visualization of modules

For each TE-regulated module, we provide an intuitive way to visualize it in a graph by incorporating the multilayer information of differential expression and differential

translation of each gene into the graph. The module is thus visualized as a graph of network in which nodes (genes) are coloured and shaped with the attributes of differential expression and differential translation, and edges are assigned with the width proportional to the matched weights.

4.4 Results

4.4.1 Seed genes

The processed transcriptome and translatome data provide mRNA and RPF expression of 10559 genes in both the original prostate cancer cells and the PP242-treated prostate cancer cells. Following the proposed approach, we assess the degrees of differential translation for each gene. The top 1000 of the most significant differentially translated genes are selected as the seed genes to search for the gene modules from the multilayer network. Figure 4.2 shows the distributions of mRNA differential expression and RPF differential expression of the seed genes as well as their involved biological processes. For the seed genes, we observe obvious discordance between their expression changes in mRNA and RPF levels. As shown in Figure 4.2A, 580 seed genes (dots in blue) are down-regulated in translation, while the other 420 genes (dots in red) are up-regulated in translation. Through the function enrichment analysis (see details in Chapter 3.4.2), the up-regulated translated genes are associated with functions related with mechanisms of post-translational modification such as "peptidyl-threonine phosphorylation" and "protein autophosphorylation" (see the red bars in Figure 4.2B), which are essential procedures after protein translation. The down-regulated translated genes are associated with the translation processes and several important metabolic pathways related with protein translation (see the blue bars in Figure 4.2B). The association between the genes and the translation related functions suggests their potential roles in translational regulation, which will help to mine the underlying patterns associated with translation regulation.

4.4.2 Evaluation of TE-regulated modules

On the basis of t-values of mRNA and RPF differential expression, a multilayer network is constructed with 8865 genes and 214350 edges. After the refinement of the modules searched with 1000 seed genes, 245 modules are identified as TE-regulated modules. The smallest module consists of six genes, while the largest module includes 183 genes.

Since the proposed approach is the first method for identification of TE-regulated modules using ribosome profiling data, there is not public method or tool that could be employed as benchmark to evaluate the efficiency of the approach. In order to obtain a general idea about the effectiveness of the proposed approach, we take 10 TE-regulated

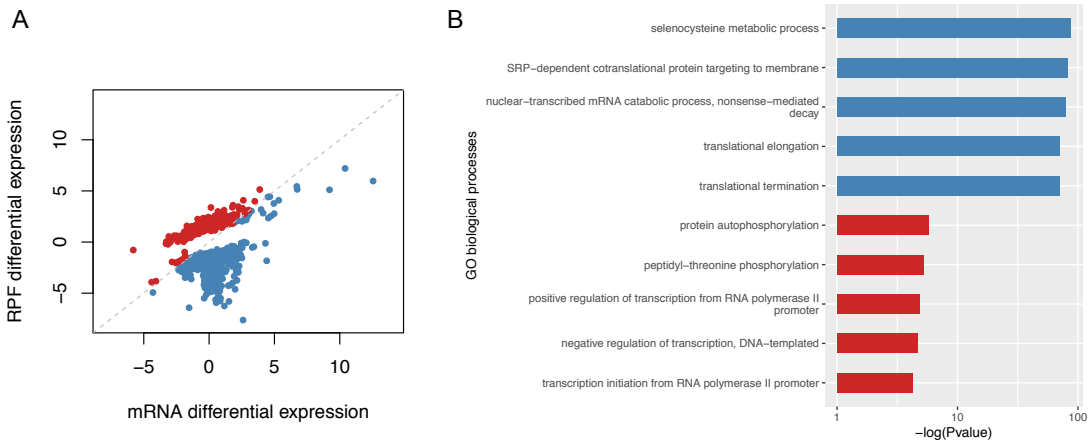


Figure 4.2: Seed genes. (A) Distribution of mRNA differential expression and RPF differential expression of the seed genes. Red dots represent the up-regulated translation and blue dots represent the down-regulated translation. (B) Top 5 Gene Ontology biological processes significantly enriched with up-regulated DTGs (red bars) and down-regulated DTGs (blue bars), respectively.

modules inferred from the top 10 significant differentially translated seed genes. Table 4.2 shows the differential expression of mRNA and RPF as well as the differential translation for each seed gene. As shown in Table 4.2, all of the top 10 seed genes are significantly down-regulated in translation efficiency (TE) in case samples with the perturbation of mTOR.

Table 4.2: Top 10 significant differentially translated seed genes of TE-regulated modules.

Seed Gene[a]	mRNA log2FC[b]	RPF log2FC[c]	TE log2FC[d]	TE p-value[e]	Module Size[f]
EIF3F	0.140	-1.232	-1.320	1.00E-08	87
PABPC4	0.322	-1.265	-1.403	1.50E-08	49
EIF4B	0.316	-1.022	-1.260	5.76E-08	16
EIF3A	0.302	-0.756	-0.964	4.00E-06	23
CHP1	0.060	-0.843	-0.925	9.00E-05	13
YBX3	0.082	-0.897	-0.974	1.03E-04	17
SPRY2	0.352	1.087	1.011	1.41E-04	6
EEF1B2	0.016	-0.976	-0.885	4.09E-04	9
RPL36	0.096	-0.741	-0.773	8.61E-04	16
RAB3A	-0.310	-0.974	-0.881	9.84E-04	7

[a] The seed genes of TE-regulated modules are ordered by the significance of differential translation.
[b] Log2 transformed fold change (FC) of mRNA expression in case vs. control.
[c] Log2 transformed fold change of RPF expression in case vs. control.
[d] Log2 transformed fold change of RPF FC vs. mRNA FC.
[e] Significance of differential translation for each seed gene.
[f] Number of genes within the TE-regulated modules inferred from each seed gene.

For each of the 10 modules, we perform the function enrichment with GO biological processes (see details in Chapter 3.4.2) to characterize its related functions, and for the corresponding seed gene, we perform curated literature search to look for scientific evidence

for its relationship with mTOR and prostate cancer. The evidence for each TE-regulated module and seed gene is listed as follows:

- EIF3F is eukaryotic translation initiation factor 3 subunit F. The TE-regulated module inferred from EIF3F is involved in the biological processes including "ubiquitin-dependent protein catabolic process" and "protein ubiquitination", which are related with the post-translational modifications of the translational machinery. EIF3F is a subunit of the largest and most complex initiation factor eIF3, which plays critical roles in translation initiation and carcinogenesis [134]. EIF3F has been proven to interact with mTOR to regulate protein synthesis [135, 136]. EIF3F has been found down-regulated in several human cancers [137, 138]. Recently, EIF3F has been found related with the genesis of prostate cancer [139].

- PABPC4 is poly(A) binding protein cytoplasmic 4. The TE-regulated module inferred from PABPC4 is involved in the biological processes including "ribonucleoprotein complex biogenesis" and "rRNA metabolic process", which are related with translational mechanisms of protein synthesis. PABPC4 has been reported to be involved in the initiation of mRNA translation [140] which may be regulated by the mTOR signalling pathway. PABPC4 has also been found overexpressed in prostate cancer cells [141], which suggests the potential relationship between PABPC4 and prostate cancer.

- EIF4B is eukaryotic translation initiation factor 4B. The TE-regulated module inferred from EIF4B is involved in the biological processes including "apoptotic signalling pathway", "cellular component organization or biogenesis", "formation of translation initiation ternary complex", "translational termination" and "translational elongation", which are related with essential translational mechanisms. It is reported that mTOR stimulates the phosphorylation and activity of EIF4B, which may promote the translation of specific mRNAs, thereby promoting cell growth, proliferation and tumour progression [142]. EIF4B has been found up-regulated in several cancers including breast, colon, head and neck, and ovarian carcinoma and non-Hodgkins lymphoma [134]. It has also been found to be related with the tumorigenesis of prostatic carcinoma [143].

- EIF3A is eukaryotic translation initiation factor 3 subunit A. The TE-regulated module inferred from EIF3A is involved in the biological processes including "regulation of metabolic process", "cellular macromolecule biosynthetic process", "cell differentiation" and "DNA damage induced protein phosphorylation", which are related with the translational regulation. EIF3A is the largest subunit of eIF3. It interacts with all other eIF3 subunits and EIF4B, which establishes a direct link

to mTOR signalling [134]. EIF3A has been found overexpressed in several human cancers, including breast, cervix, colon, lung, urinary bladder, esophagus, and oral squamous cell carcinoma [134]. Recently, Yin et al. studied the frequency of EIF3A somatic alterations in human cancers based on the analysis of catalogue of somatic mutations in cancer (COSMIC) database, and the results show that the predominate somatic mutation patterns of EIF3A in prostate cancer are the deletions in untranslated regions (UTR) [144].

- CHP1 is calcineurin like EF-hand protein 1. The TE-regulated module inferred from CHP1 is involved in the biological processes including "polysaccharide metabolic process", "phospholipid metabolic process" and "post-translational protein modification", which are related with the post-translational modifications of the translational machinery. CHP1 encodes a phosphoprotein that binds to the Na+/H+ exchanger NHE1 and serves as an essential cofactor which supports the physiological activity of NHE1 [145]. NHE1 has been found to be regulated by mTOR, which implicates the possible downstream effector role of NHE1 contributing to mTOR's effects on cell growth, proliferation and tumorigenesis [146]. NHE1 has also been found related with tumorigenesis of prostate cancer [147], which suggests the potential relationship between CHP1 and prostate cancer.

- YBX3 is Y-box binding protein 3. The TE-regulated module inferred from YBX3 is involved in the biological processes including "positive regulation of epithelial cell proliferation", "transforming growth factor beta receptor signalling pathway" and "BMP signalling pathway". YBX3 belongs to the gene family of Y-box binding protein as well as the well-established oncoprotein YBX1. The synthesis of YBX1 is activated by mTOR signalling [148] and YBX1 has also been found related with prostate cancer progression [149]. Recently, a novel gene fusion between YBX3 and STYK1 with clinical relevance has been identified through whole-genome sequencing of small-cell prostate carcinoma [150], which suggests the potential relationship between YBX3 and prostate cancer.

- SPRY2 is sprouty RTK signalling antagonist 2. The TE-regulated module inferred from SPRY2 is involved in the biological processes including "negative regulation of MAP kinase activity", "response to fibroblast growth factor" and "Ras protein signal transduction", which are related with regulatory mechanisms of translation. It is reported that the PI3K/AKT/mTOR signalling is a key pathway throughout the development of prostate cancer [151]. SPRY2 has been proven an important tumour suppressor in prostate cancer which drives PI3K/AKT/mTOR pathway through its dysfunction [152].

- EEF1B2 is eukaryotic translation elongation factor 1 beta 2. The TE-regulated module inferred from EEF1B2 is involved in the biological processes including "translational elongation". Overexpression of EEF1B2 was observed in most of cancer types [153]. Recently, Hassan et al. found that EEF1B2 was up-regulated in the high-risk group patients in the survival analysis of prostate cancer, although the difference between survival outcomes of the two groups was not significant [154].

- RPL36 is ribosomal protein L36. The TE-regulated module inferred from RPL36 is involved in the biological processes including "formation of translation initiation ternary complex", "translational termination", "translational elongation" and "cellular amino acid metabolic process", which are related with essential translational mechanisms. A recent cancer cohort study found that RPL36 was up-regulated in nine cancer clusters arising from thyroid, brain, liver, kidney clear cell, thymoma, prostate, pancreatic, pheochromocytoma and paraganglioma, and B-cell lymphoma [155], which suggests the potential relationship between RPL36 and human cancers including prostate cancer.

- RAB3A is RAB3A, member RAS oncogene family. The TE-regulated module inferred from RAB3A is involved in the biological processes including "regulation of translational elongation" and "cellular protein complex disassembly", which are related with translational regulation mechanisms. RAB3A has been found up-regulated in human cancers including insulinoma, breast cancer, hepatocellular carcinoma, and tumours derived from the neural system [156].

To summarize the above, all of the 10 TE-regulated modules are enriched in biological processes related with translational mechanisms. Solid evidence or potential evidence has been found for nine out of the 10 corresponding seed genes, excluding RAB3A, to support their relationships with mTOR or prostate cancer. Although there is not direct evidence for the relationship between RAB3A and prostate cancer, but RAB3A has been reported related with several other human cancers. Consequently, the proposed approach is effective and capable of mining the underlying functional patterns related with translational regulation mechanisms from the ribosome profiling data.

4.4.3 Case study of TE-regulated modules

We successfully mine the functional modules associated with key regulators downstream of mTOR, such as EIF4EBP1, EIF4EBP2 and YBX1, which provides clues for understanding the translation regulation mechanisms induced by mTOR related with prostate cancer development.

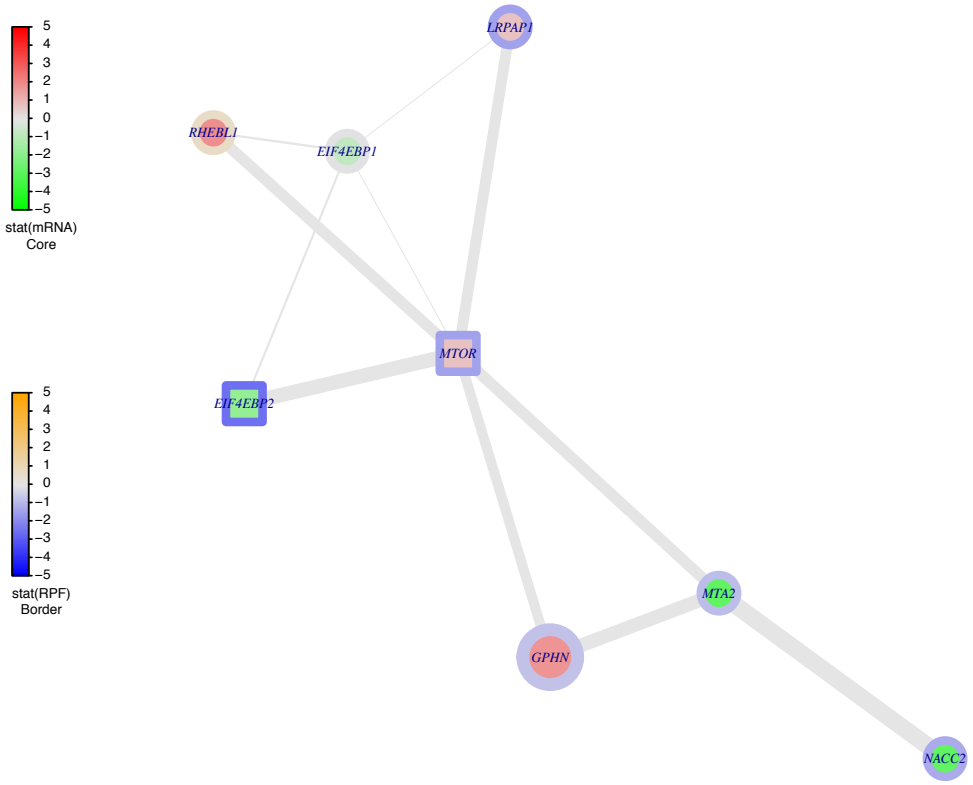


Figure 4.3: A TE-regulated module searched by the seed gene GPHN. Nodes represent genes and edges represent differential expressed interaction between the nodes linked by the edge. Node colours denote the statistics of differential expression: core and border for mRNA and RPF, respectively. Square nodes represent the significant differentially translated genes. Edge width denotes the weights of differential expression on the edges. The Node with larger size denotes the seed gene.

4.4.3.1 The mTOR module

Figure 4.3 shows a key TE-regulated module identified from the multilayer network by the seed gene GPHN, which involves the key translation regulator mTOR. In this functional module, mTOR interacts with the downstream genes EIF4EBP1 and EIF4EBP2. EIF4EBP1 is a major regulators of protein synthesis downstream of mTOR [157], which negatively regulates the translation initiation factor eIF4E. Phosphorylation of EIF4EBP1 by mTORC1 leads to its dissociation from eIF4E, which actives translation initiation [158]. As shown in Figure 4.3, mTOR interacts with both EIF4EBP1 and its homologue EIF4EBP2, thus treatment with mTOR inhibition significantly affects the activities of downstream EIF4EBP1 and EIF4EBP2. A key feature of prostate cancer metastasis is the ability of epithelial cells to migrate and invade, which can be induced by mTOR through the translational control of pro-invasion mRNAs. In order to investigate whether the translational regulator EIF4EBP1 and EIF4EBP2 control the expression of the mTOR-sensitive pro-invasion genes in these PP242 treated prostate cells, Hsieh et al. [131]

performed knockout experiments on EIF4EBP1 and EIF4EBP2 to explore the potential mechanisms. They found that the knockout of only EIF4EBP1 does not change the expression of the mTOR-sensitive pro-invasion genes, but the knockdown of both EIF4EBP1 and EIF4EBP2 reduces the effect of mTOR on the expression of pro-invasion genes. Hsieh et al. suggested that the reason for no alterations of pro-invasion gene expression on EIF4EBP1 knockout might be the complementary role of its homologue EIF4EBP2. The module in Figure 4.3 might provide an evidence for the question because mTOR exhibits a much stronger functional link with EIF4EBP2 than with EIF4EBP1. It suggests that in these prostate cancer cells, the translation control between mTOR and the pro-invasion mRNAs is mainly intermediated by the regulator EIF4EBP2 but not EIF4EBP1, and therefore, knockout of EIF4EBP1 will not affect the expression of the mTOR-sensitive pro-invasion genes. Besides, Hsieh et al. [131] showed that a key regulator of prostate cancer invasion and metastasis, MTA1, is also translational controlled by the oncogenic mTOR signalling. Although the direct interaction between mTOR and MTA1 is not revealed in the module, but we observe a strong interaction between mTOR and MTA2, a homologue to MTA1, which has also been reported significantly overexpressed in metastatic prostate cancer [159].

4.4.3.2 The YBX1 module

Another important regulator is YBX1, Y box binding protein 1, whose aberrant expression has been proven associated with cancer proliferation in numerous tissues [160]. YBX1 is a well-established oncoprotein. The synthesis of YBX1 is activated by mTOR signalling [148] and YBX1 has also been found related with prostate cancer progression [149]. In this prostate cancer cell, it has been identified as a significant differentially translated gene, which suggests that it might be associated with the translation regulation in prostate cancer. Applying the proposed approach, we mine two functional modules associated with YBX1, shown in Figure 4.4. The two modules are identified as significant dynamic modules between the two layers of the multilayer network, which suggests strong associations with the potential translation regulation mechanisms. For example, the interaction between YBX1 and LARP1 (shown in Figure 4.4A) and the interaction between YBX1 and HNRNPA1 (shown in Figure 4.4B) have been reported in cancer [161, 162]. These functional modules provide clues for exploring their potential roles in the translational control associated with prostate cancer genesis.

4.4.4 R package: TERM

We develop TERM, an R package for identification of Translation Efficiency Regulated Modules, for implementation of the proposed approach. TERM is available from `https:`

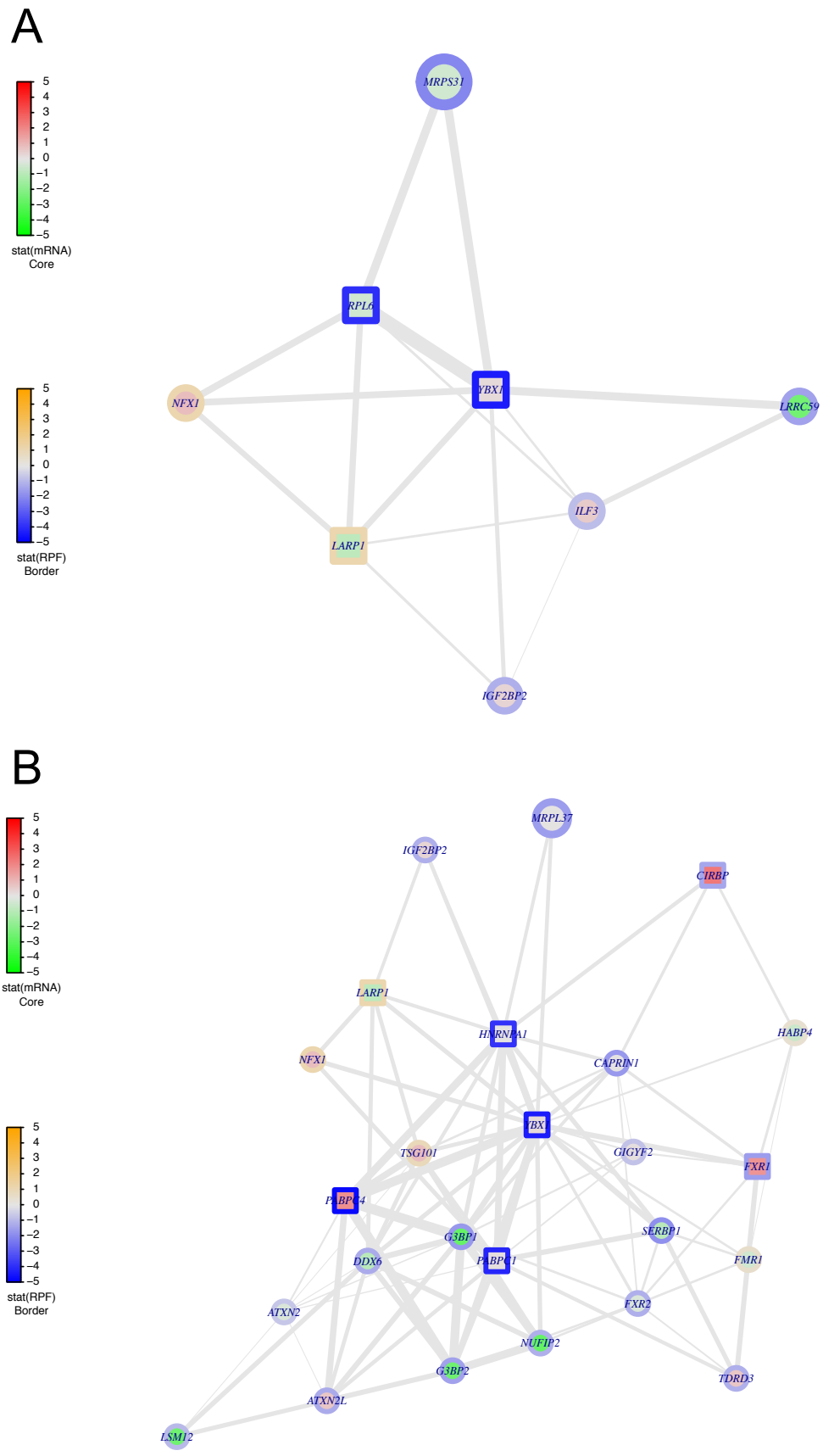


Figure 4.4: TE-regulated modules involving YBX1. Figure legends are the same as Figure 4.3.

//github.com/bioinfoxh/TERM. Different versions of the source code for TERM can be downloaded from https://github.com/bioinfoxh/TERM/releases. So far, TERM is the first tool developed for identification of TE-regulated modules by integrating RNA-seq and Ribo-seq data.

4.4.4.1 Design and implementation of TERM

Since no tools have been developed for identification of TE-regulated modules from ribosome profiling data, we implement the proposed approach as an R package, TERM, to provide the first tool for easy use for the scientific fields. TERM implements the proposed approach in R. The proposed approach aims to identify TE-regulated modules, which can be divided into three subtasks for implementation:

- Firstly, selecting top ranked differentially translated genes as seed genes (corresponding to Step 2 in Figure 4.1).
- Then, identifying TE-regulated modules using the selected seed genes through multilayer network analysis (corresponding to Step 1, 3 and 4 in Figure 4.1).
- Finally, visualizing the identified TE-regulated modules as graphs (corresponding to Step 5 in Figure 4.1).

TERM provides three R functions to achieve the up-mentioned three subtasks, respectively:

- *calculateDTE*: This R function achieves the subtask 1. It employs the R packages DESeq2 and Riborex to calculate the fold changes and the significance for the differential expression of mRNAs and RPFs, as well as the differential translation of genes. Based on the output of *calculateDTE* function, the genes can be ordered according to their significance of differential translation. Consequently, users can select top ranked genes as seed genes for practical use.
- *detectTERM*: This R function achieves the subtask 2. It implements the multilayer network module detection algorithm, M-module, to identify TE-regulated modules. *detectTERM* constructs a multilayer network by integrating RNA-seq and Ribo-seq data with gene-gene interaction network, and searches modules from the multilayer network using greedy search for the selected seed genes. Then, this function performs refinements (see details in Chapter 4.3.4) with the modules to identify TE-regulated modules.
- *plotTERM*: This R function achieves the subtask 3. It employs the R package igraph to provide an intuitive way to visualize the TE-regulated modules. For each

TE-regulated module, $plotTERM$ generates a figure to show the module in a graph by incorporating the multilayer information of differential expression and differential translation of each gene. The module is thus visualized as a graph of network in which nodes (genes) are coloured and shaped with the attributes of differential expression and differential translation, and edges are assigned with the width proportional to the matched weights (as illustrated in Figure 4.3).

Besides the up-mentioned R functions, TERM also provides another R function, named *term*, which achieves the first two sub-tasks as a whole pipeline by wrapping the functions $calculateDTE$ and $detectTERM$ together.

4.4.4.2 Installation of TERM

Before using the R package TERM, users need to install it correctly in their R environments. Because TERM is developed based on some other R packages including DESeq2, igraph, Matrix and corrplot. So these packages are prerequisite for the installation of TERM. We recommend users to install them following the instructions for "Installations of Packages" from CRAN (`https://cran.r-project.org/`). To install TERM, we advise users to download the latest version of TERM (e.g., term_1.0.tar.gz) from `https://github.com/bioinfoxh/TERM/releases`. In this way, TERM can be installed using the following command in R:

$$install.packages(\text{“}LocalPath/term\_1.0.tar.gz\text{”}, repos = NULL)$$

The $LocalPath$ is the local path where users put the downloaded file term_1.0.tar.gz in their computers.

4.4.4.3 Instructions for using TERM

TERM takes RNA-seq data, Ribo-seq data and gene-gene interaction data as the input. To use TERM correctly, the input data should be processed into the following formats:

- Raw read counts matrix of mRNA
  The raw read counts matrix of mRNA represents the read counts of genes (in rows) across all samples (in columns). It is obtained by processing the raw RNA-seq data. There are many well-established pipelines for processing RNA-seq data, which can be used to get required mRNA matrix.

- Raw read counts matrix of RPF
  The raw read counts matrix of RPF represents the read counts of genes (in rows) across all samples (in columns). It is obtained by processing the raw Ribo-seq data

for the matched samples of RNA-seq data. The required RPF matrix can be obtained in the same way as the RNA-seq data.

- Two columned matrix for gene-gene interaction data
  The gene-gene interaction data are processed into a two columned matrix, in which each line represents the interacted gene pairs.

With the correctly formatted input data, TERM can be implemented in R using the following commends for the four aforementioned functions including *term*, *calculateDTE*, *detectTERM* and *plotTERM*:

$$\begin{aligned} &term(raw\_rna, raw\_ribo, rna\_label, ribo\_label, baseLevel, raw\_ppi, \\ &\quad minCounts = 10, minCountsProportion = 1, num\_seed = 100, \\ &\quad minModSize = 10, permTimes = 100, modularity\_p = 0.05, \\ &\quad maxModOvlp = 0.5) \end{aligned}$$

$$\begin{aligned} &calculateDTE(raw\_rna, raw\_ribo, rna\_label, ribo\_label, baseLevel, \\ &\quad minCounts = 10, minCountsProportion = 1) \end{aligned}$$

$$\begin{aligned} &detectTERM(ppi, DEstat, TEstat, num\_seed = 100, minModSize = 10, \\ &\quad permTimes = 100, modularity\_p = 0.05, maxModOvlp = 0.5) \end{aligned}$$

$$\begin{aligned} &plotTERM(TEmodule, layout\_style = "layout.fruchterman.reingold", \\ &\quad gene2symbol = NULL) \end{aligned}$$

The details of the parameters for each function can be accessed using R command $help(function)$, where $function$ is the name of the function.

As mentioned before, the RNA-seq data and Ribo-seq data include 10559 genes, and the gene-gene interaction data include 214350 interacted gene pairs. Implementing the function *term* on this dataset using a single core of a MacBook Pro with 2.9 GHz Intel Core i5 and 16GB 1867 MHz DDR3 memory, it takes about 107 minutes to identify the aforementioned 245 TE-regulated modules. If the gene interactome data are larger, we would advise to use TERM on high performance servers or clusters with more advanced computing resources.

4.5 Implementation

The data sources of the transcriptome, translatome and interactome data studied in this chapter are described in Chapter 4.2.1. We develop an R package TERM for implementation of the proposed approach, and use TERM to identify gene modules related with

mTOR translational regulation in human prostate cancer by integrating the transcriptome, translatome and interactome data. The processed datasets used in this study and the source code of R package TERM can be accessed from `https://github.com/bioinfoxh/TERM`.

4.6 Summary

In this chapter, we propose an approach for gene module detection by integrating transcriptome, translatome, and interactome using multilayer network, which consists of five steps: (1) construction of multilayer differential expression network by integrating transcriptome and translatome with interactome data respectively; (2) selection of seed genes for module detection by evaluating their degrees of differential translation; (3) detection of modules from the multilayer network using greedy search for each seed gene by minimizing the entropy-based local modularity function; (4) identification of translation efficiency (TE) regulated modules by the refinements including significance assessment, redundancy deletion and dynamic evaluation; (5) visualization of TE-regulated modules as graphs with incorporated multilayer information from the networks.

We apply the proposed approach on a published ribosome profiling data of mTOR perturbed prostate cancer cells and mine several TE-regulated modules associated with mTOR perturbation. The translational regulated genes and modules downstream mTOR provide valuable clues for understanding the mTOR associated translational regulation mechanisms in prostate cancer genesis and metastasis.

We develop an R package, TERM, for implementation of the proposed approach, which is capable of evaluating differential translation of genes, identifying TE-regulated modules, and visualizing the TE-regulated modules. It is a useful tool for exploring translational regulation mechanisms by integrating transcriptome, tanslatome and interactome data.

This chapter provides an efficient approach using multilayer network clustering for transcriptome, translatome and interactome data integration in the field of multi-omics. It is the first method for identifying translational regulation related gene functional modules by integrating the transcriptome, translatome and interactome data. We believe the proposed approach and the R package TERM are valuable tools for the field of multi-omics, which helps researchers to understand the underlying translational regulation mechanisms through the data mining from ribosome profiling data combined with interactome data.

This approach could be further improved in two directions. Firstly, to gain a further understanding of translational regulation mechanisms, the proposed approach can be extended into three-layer multilayer network by adding a layer for proteome data. Since the proteome data is the final molecular level of translation, it will provide a more comprehensive view for understanding translational regulation mechanisms by integrating proteome with transcriptome and translatome data. Secondly, there is a limitation of the

proposed approach that it only works with balanced data, i.e., the sample sizes for case and control are equal in the datasets, due to the limitation of the differential translation evaluation method in the step of seed gene selection. But in practice, the datasets are usually with unbalanced sample sizes, and the approach is needed to be applicable to the unbalanced data. This issue can be addressed by employing effective methods for differential translation evaluation that can deal with unbalanced data.

Chapter 5

Network-constrained regression for transcriptome and interactome integration

5.1 Introduction

A key problem in transcriptomics research is to select genes whose expression are predictive for a phenotype or a clinical outcome. A prediction model can be built based on the selected genes to predict the outcomes for future transcriptome data. The problem can be considered as a supervised learning problem of classification or regression for predicting binomial or quantitative outcomes correspondingly. It can in general be formulated as a prediction problem with n observations of $(y_i, x_i), i = 1...n$, where y_i is the response and $x_i = (x_{i1}, ..., x_{ip})$ are the p predictors. Considering the standard linear regression model, the response y is predicted by:

$$\hat{y}_i = \hat{\beta}_0 + x_{i1}\hat{\beta}_1 + \cdots + x_{ip}\hat{\beta}_p \tag{5.1}$$

where a model-fitting procedure estimates the vector of coefficients $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p)$. Predictors with non-zero coefficients are selected as the predictive features. Coefficient estimation can therefore achieve the goals of both prediction and feature selection.

In practice, trancriptome data are high-dimensional data with $p \gg n$, which easily results in overfitting linear models. To overcome this problem, many regularized regression methods have been proposed to provide solutions for coefficient sparsity to enable feature selection from the high-dimensional data, such as Lasso [45], ElasticNet [46], Group Lasso [47], Sparse-Group Lasso [48] and Group Exponential Lasso [49] (see details in Chapter 2.3.1.4). However, these methods treat all genes equally a priori without utilizing the prior correlated structures among genes which cannot be ignored in genomic study.

It is known that in biological systems, molecules do not work independently but function in a coordinate way through interactions with each other. Molecular interaction networks formed by interactome data provide comprehensive functional context of genes. Recently, several effective regularized regression methods, such as Grace [51] and GBL [52] (see details in Chapter 2.3.1.5), have been developed to take into account such network structured prior biological knowledge for feature selection, which are referred to as network-constrained regression. Network-constrained regression performs regularization on coefficients incorporating the network correlation structure based on two assumptions: (i) hub genes are supposed to have larger coefficients due to more crucial roles in the network; (ii) two genes that are linked in the network tend to have similar degree-scaled coefficients because they are functional correlated to each other.

Network-constrained regression offers a practical way to integrate transcriptome and interactome data. Although the existing network-constrained regression methods have proven effective for variable selection in various applications, there is still room for improvement of efficiency in feature selection and prediction.

5.2 Multi-omics

5.2.1 Multi-omic data

5.2.1.1 Transcriptome

The transcriptome dataset studied in this chapter consists of Affymetrix array transcriptomic profiling of 10 blastocysts which were developed in the EmbryoScope time-lapse system [163]. The data are provided by our collaborator in EpiHealthNet ITN, Prof Daniel Brison from The University of Manchester. Genome-wide gene expression values of each blastocyst are obtained by pre-processing the raw data using RMA [87] and mas5call [88], which are implemented in R using the affy package [89]. Because the manuscript of this study is in preparation for submission, this gene expression dataset is not public at the moment, but it will be released as soon as we submit the manuscript.

During the in vitro fertilization (IVF) treatment, the pre-implantation embryos are developed in the incubators until the late blastocyst stage, and then the ones with good qualities are selected to implant into the mother's uterus to get pregnancy. EmbryoScope is a new type of incubator that maintains the necessary physiological conditions required by a living embryo while they are in the IVF laboratory. It has an incorporated time-lapse system that has a camera that continuously captures images and records them as a video of the embryonic development. Compared with the conventional incubator, the EmbryoScope time-lapse system offers the advantages that allows embryologists to monitor embryo development without taking the embryos out of the incubator. Time-lapse

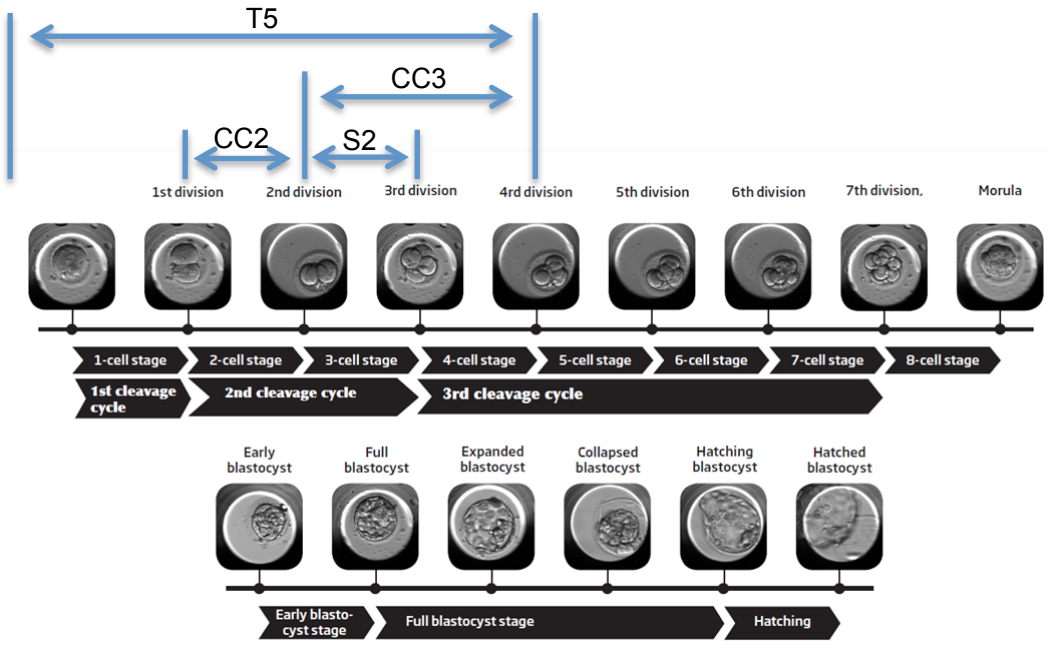


Figure 5.1: Key time-lapse parameters of human pre-implantation embryonic development. The image is adapted from [166].

Table 5.1: Key time-lapse parameters.

Parameter	Explanation
CC2	Second Cell Cycle, duration of the period as 2-blastomere embryo
CC3	Third Cell Cycle, time taken to develop from 3-blastomere embryo to a 5-blastomere embryo
S2	Synchrony in division from 2-blastomere embryo to a 4-blastomere embryo
T5	Time to 5-cell stage, an annotation point in which the embryo finishes the division to become 5 cells

technology with image analysis software allows for the tracking of specific timings between developmental events, which produces the corresponding time-lapse parameters during the embryo development. The 10 blastocysts in this study encompass a various range of qualities. For each blastocyst, four key EmbryoScope time-lapse parameters are also recoded during its development, including CC2, CC3, S2 and T5 (details are shown in Figure 5.1 and Table 5.1). These four parameters have been reported to be the most important checkpoints for human embryo development [164, 165], which could serve as predictors of blastocyst qualities and are therefore used for the clinical embryo selection during the IVF treatment.

5.2.1.2 Interactome

Human gene interactome data are downloaded from PathwayCommons (version 8) [90], which is a collection of public available human pathway data. The interactome data

provides comprehensive functional interactions, such as biochemical reactions, complex assembly, transport and catalysis events, and physical interactions, among molecules including proteins, DNA, RNA, small molecules and complexes. All the molecules in the interactome data are mapped to the corresponding genes. After removing the duplicated interactions and self-interactions, the remaining interactions form a gene-gene interaction network.

5.2.2 Problem definition

The EmbryoScope time-lapse system offers advantages of improvement of embryo selection for IVF treatment. It is reported that EmbryoScope system also result in a higher portion of good quality embryos than the conventional incubators [167]. Time-lapse parameters have been shown capable of predicting embryo development to blastocyst stage in conventional incubators, which is associated with transcriptional patterns [113]. However, such transcriptional patterns for EmbryoScope developed blastocysts have not been studied yet. Based on the transcriptomes of EmbryoScope developed blastocysts and the corresponding time-lapse parameters, we aim to identify the important genes whose expression are associated with the key time-lapse parameter. It is a supervised learning problem of feature selection, which can be solved by using linear regression methods to fit a linear model where the time-lapse parameter is the response and the predictors are the genes.

5.3 Methodology

Since the task of this chapter is feature gene selection from transcriptome data, it can be addressed by using linear regression methods. But the transcriptome data are high-dimensional data where the number of genes is too much larger than the sample size, which easily results in overfitting linear models. Besides, the genes exhibit highly dynamic correlation patterns in functions during human pre-implantation development, which suggests a correlated structure among the predictors and can not be ignored when fitting the linear model. Several regularized linear regression methods and network-constrained regression methods such as Lasso, ElasticNet, Grace and GBL have been developed to reduce the overfitting by introducing regularization and incorporating the network structured correlations between genes for coefficient estimation. But these state-of-art methods still suffer low power and high bias for feature selection. To overcome limitations of the state-of-art methods, we propose a more efficient network-constrained regression method and use it to select time-lapse parameter associated genes by incorporating the human pre-implantation embryonic development co-expression network constructed in Chapter 3.

5.3.1 Network-constrained regression method

In this chapter, we develop a new network-constrained regression method, named as eGBL which stands for edge-based Generalized Boosted Lasso, to select feature genes whose expression in blastocysts are associated with a time-lapse parameter.

Let $y = (y_1, ..., y_n)^T$ be a quantitative time-lapse parameter which contains a vector of time-lapse parameters for n embryos. Let $x_i = (x_{i1}, ..., x_{ip})$ be the vector of expression levels at blastocyst stage for p genes in embryo i, where $i = 1, ..., n$. Following the linear regression model introduced in Chapter 2.3.1.1, the simple linear model between y_i and x_i is defined as:

$$y_i = \beta_0 + x_i\beta + \epsilon_i, \quad i = 1, ..., n \tag{5.2}$$

where $\beta = (\beta_1, ..., \beta_p)^T$ are the regression coefficients to the p genes and β_0 is the intercept of the linear model. The coefficients can be given by the ordinary least square (OLS) estimator $\hat{\beta}$ which minimizes the loss function $L(\beta)$ defined by the sum of squared residual as follows:

$$\hat{\beta} = \arg\min_{\beta} L(\beta) = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - x_i\beta)^2 \tag{5.3}$$

In the transcriptomic studies where the high-dimensional data usually with $p \gg n$, the OLS estimation does not perform well due to the overfitting. Consequently, regularized regression has been proposed, in which a penalty $P(\beta)$ is added to the loss function for regularization of the coefficients. The estimator $\hat{\beta}$ is thus given by:

$$\hat{\beta} = \arg\min_{\beta} \left(\sum_{i=1}^{n} (y_i - x_i\beta)^2 + P(\beta) \right) \tag{5.4}$$

As introduced in Chapter 2.3.1, several penalized regression methods have been developed to offer the smooth and sparse solutions for coefficient regularization for the sake of feature selection, such as Lasso, ElasticNet, Group Lasso, GEL, and the network-constrained regression methods Grace and GBL that incorporate the network-structured prior knowledge of genes into the coefficient regularization. However, there are some limitations for these existing methods, e.g., the low power for feature selection and the bias for prediction (see details in Chapter 2.3.1). Consequently, more efficient regression methods are needed to be developed, and we propose a new network-constrained regression method with a more efficient penalty function modified based on GBL penalty.

Among the up-mentioned penalized regression methods, GBL has been proved capable of providing better performance than the others in terms of feature selection [52]. The

penalty function of GBL is defined as follows:

$$P(\beta) = \lambda \sum_{i \sim j} \left[\left(\frac{|\beta_i|}{d_i} \right)^{\gamma} + \left(\frac{|\beta_j|}{d_j} \right)^{\gamma} \right]^{1/\gamma} \tag{5.5}$$

where $\gamma > 1$ and $\lambda > 0$ are two parameters to be specified. This penalty is capable of variable selection by shrinking the coefficients on each edge over the network based on the two assumptions of network-constrained regression: (i) hub genes are supposed to have larger coefficients due to more crucial roles in the network, and (ii) two genes that are linked in the network tend to have similar degree-scaled coefficients because they are functional correlated to each other. Specifically, for two linked gene i and j, the penalty captures grouping effects in shrinking the magnitudes of two scaled estimated coefficients towards each other, i.e. $|\hat{\beta}_i|/d_i = |\hat{\beta}_j|/d_j$, and particularly, enforces $\hat{\beta}_i = 0$ and $\hat{\beta}_j = 0$ when $\beta_i = \beta_j = 0$. The penalty performs regularization on the scaled coefficients which allows a predictor with bigger d_i to have larger $\hat{\beta}_i$, that is, a hub gene tends to have a larger coefficient. This penalty performs well in terms of variable selection, especially if $\gamma \to \infty$, and the penalty becomes:

$$P(\beta) = \lambda \sum_{i \sim j} \max \left(\frac{|\beta_i|}{d_i}, \frac{|\beta_j|}{d_j} \right) \tag{5.6}$$

In spite of the success in variable selection, GBL suffers a strong bias in prediction errors, which might be due to the limitations of the penalty. It is worthy to be noted that the penalty treats each edge equally without utilizing the strength of the correlation between two linked genes in the network. In practice, particularly in the dynamic biological systems, the correlations among genes are not simply defined as 0 or 1, but with a continuous measure denoting the different strength. Therefore, the edges in the network are assigned with a weight indicating the strength of the correlation between the two linked genes, which leads to a weighted network. To take the full advantage of such weighted network structured knowledge, we proposed an extension for the GBL penalty by incorporating the weights of edges, named as edge-based GBL (eGBL):

$$P(\beta) = \lambda \sum_{i \sim j} w(i,j) \left[\left(\frac{|\beta_i|}{d_i} \right)^{\gamma} + \left(\frac{|\beta_j|}{d_j} \right)^{\gamma} \right]^{1/\gamma} \tag{5.7}$$

where parameter $\gamma > 1$ and $\lambda > 0$ are the same as the GBL penalty. $w(i,j)$ denotes the weight of the edge between gene i and j. When $\gamma \to \infty$, it becomes:

$$P(\beta) = \lambda \sum_{i \sim j} w(i,j) \left[\max \left(\frac{|\beta_i|}{d_i}, \frac{|\beta_j|}{d_j} \right) \right] \tag{5.8}$$

5.3.2 Evaluation of regression method

We employ the Generalized Boosted Lasso (GBL) algorithm [52, 53] for implementation of the proposed method eGBL, which uses coordinate descent to learn coefficients for a linear model by minimizing the loss function. In order to test the performance of eGBL, we evaluate it on simulated data using the hold-out validation (see details in Chapter 2.3.5.1). Based on a simulated training set and a simulated test set, the linear model is fitted on the training set, which is then used to make the predictions on the test set. The performance can be evaluated by the following measures:

- $RMSE$
  Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors on training set:
  $$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2} \tag{5.9}$$
  where the predicted response $\hat{y}_i = x_i\hat{\beta} + \hat{\beta}_0$, $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)$ is the vector of estimated coefficients for each predictor, and $\hat{\beta}_0$ is the predicted intercept of the linear model.

- $PMSE$
  Prediction Mean Squared Error (PMSE) is the square root of the mean of the squared errors on independent test set:
  $$PMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2} \tag{5.10}$$
  where the predicted response $\hat{y}_i = x_i\hat{\beta} + \hat{\beta}_0$, $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)$ is the vector of estimated coefficients for each predictor, and $\hat{\beta}_0$ is the predicted intercept of the linear model fitted on the training set.

- $corR$
  The concordance between an estimator $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)$ and the actual coefficients $\beta = (\beta_1, ..., \beta_p)$, $corR$, is calculated by the Pearson Correlation Coefficient:
  $$corR = \frac{\sum_{i=1}^{p}[\beta_i - \overline{\beta}][\hat{\beta}_i - \overline{\hat{\beta}}]}{\sqrt{\sum_{i=1}^{p}[\beta_i - \overline{\beta}]^2}\sqrt{\sum_{i=1}^{p}[\hat{\beta}_i - \overline{\hat{\beta}}]^2}} \tag{5.11}$$

- $precision$, $recall$, F-score
  Let the coefficient estimation be considered as a classification problem. Given the actual coefficients $\beta = (\beta_1, ..., \beta_p)$, the actual positive feature set and the actual negative feature set are defined as the predictors with $\beta \neq 0$ and $\beta = 0$ respectively.

Similarly, the predicted positive feature set and the predicted negative feature set are defined as the predictors with $\hat{\beta} \neq 0$ and $\hat{\beta} = 0$ based on the estimator $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)$. The common measures including *precision*, *recall* and F-score are used for evaluating the classification (see details in Chapter 2.3.5.2).

5.4 Results

5.4.1 Simulation study

To evaluate the performance of the proposed network-constrained regression method eGBL, we set up four simulations similar as that of Li et al. [51]. We first simulate a weighted gene co-expression network compromised of 50 subnetworks, each of which includes one hub genes interacting with 10 neighbouring genes with the corresponding decreasing co-expression weights as $w = (0.95, 0.85, ..., 0.05)$. The resulting simulated weighted network includes 550 edges among 550 genes. Then, based on the simulated weighted network, two simulated gene expression datasets are set up with assumptions of two different proportions of feature genes.

Simulation set-up 1 The expression levels for the 50 hub genes in the simulated network follow standard normal, $x_h \sim N(0, 1), h = 1, ..., 50$. Given a hub gene x_h, the expression levels of its neighbour genes follow normal distributions with correlations w to standard normal, that is, the expression levels of the neighbour genes $x_{hj} \sim N(w_j * x_h, 1 - w_j^2), j = 1, ..., 10$.

We assume that 10% of the genes are features that are related to the response variable y, and select five hub genes and their neighbour genes assigned with non-zero coefficients. The coefficients of all genes are defined by:

$$\beta = (9, \underbrace{\frac{9}{\sqrt{10}}, \cdots, \frac{9}{\sqrt{10}}}_{10}, -7, \underbrace{\frac{-7}{\sqrt{10}}, \cdots, \frac{-7}{\sqrt{10}}}_{10}, 5, \underbrace{\frac{5}{\sqrt{10}}, \cdots, \frac{5}{\sqrt{10}}}_{10},$$
$$-3, \underbrace{\frac{-3}{\sqrt{10}}, \cdots, \frac{-3}{\sqrt{10}}}_{10}, 1, \underbrace{\frac{1}{\sqrt{10}}, \cdots, \frac{1}{\sqrt{10}}}_{10}, 0, \cdots, 0)$$

The response variable y is derived from a linear model $y = x\beta + \epsilon$, where the noise $\epsilon \sim N(0, \sigma_e^2), \sigma_e^2 = \sum_{i=1}^{n} \beta_i^2/2$.

The training set is simulated with 50 samples, $n = 50$, corresponding to a "large p, small n" situation which is common in transcriptomic study. To evaluate the performance of eGBL on an independent dataset, we simulate another test set in the same way with $n = 100$.

Simulation set-up 2 The second set-up is simulated in the same way as for the first set-up but with the assumption that 20% of the genes are features. The coefficients are defined by:

$$\beta = (9, \underbrace{\frac{9}{\sqrt{10}}, \cdots, \frac{9}{\sqrt{10}}}_{10}, -9, \underbrace{\frac{-9}{\sqrt{10}}, \cdots, \frac{-9}{\sqrt{10}}}_{10}, 7, \underbrace{\frac{7}{\sqrt{10}}, \cdots, \frac{7}{\sqrt{10}}}_{10}, -7, \underbrace{\frac{-7}{\sqrt{10}}, \cdots, \frac{-7}{\sqrt{10}}}_{10},$$
$$5, \underbrace{\frac{5}{\sqrt{10}}, \cdots, \frac{5}{\sqrt{10}}}_{10}, -5, \underbrace{\frac{-5}{\sqrt{10}}, \cdots, \frac{-5}{\sqrt{10}}}_{10}, 3, \underbrace{\frac{3}{\sqrt{10}}, \cdots, \frac{3}{\sqrt{10}}}_{10}, -3, \underbrace{\frac{-3}{\sqrt{10}}, \cdots, \frac{-3}{\sqrt{10}}}_{10},$$
$$1, \underbrace{\frac{1}{\sqrt{10}}, \cdots, \frac{1}{\sqrt{10}}}_{10}, -1, \underbrace{\frac{-1}{\sqrt{10}}, \cdots, \frac{-1}{\sqrt{10}}}_{10}, 0, \cdots, 0)$$

Simulation set-up 3 The third set-up is simulated in the same way as for the first set-up but with a different scale factor for the coefficients of the genes. The coefficients are defined by:

$$\beta = (9, \underbrace{\frac{9}{10}, \cdots, \frac{9}{10}}_{10}, -7, \underbrace{\frac{-7}{10}, \cdots, \frac{-7}{10}}_{10}, 5, \underbrace{\frac{5}{10}, \cdots, \frac{5}{10}}_{10},$$
$$-3, \underbrace{\frac{-3}{10}, \cdots, \frac{-3}{10}}_{10}, 1, \underbrace{\frac{1}{10}, \cdots, \frac{1}{10}}_{10}, 0, \cdots, 0)$$

Simulation set-up 4 The fourth set-up is simulated in the same way as for the third set-up but taking into account the opposite signs of the coefficients between two linked genes. The coefficients are defined by:

$$\beta = (9, \underbrace{\frac{9}{10}, \cdots, \frac{9}{10}}_{7}, \frac{-9}{10}, \frac{-9}{10}, \frac{-9}{10}, -7, \frac{7}{10}, \frac{7}{10}, \frac{7}{10}, \underbrace{\frac{-7}{10}, \cdots, \frac{-7}{10}}_{7},$$
$$5, \underbrace{\frac{5}{10}, \cdots, \frac{5}{10}}_{7}, \frac{-5}{10}, \frac{-5}{10}, \frac{-5}{10}, -3, \frac{3}{10}, \frac{3}{10}, \frac{3}{10}, \underbrace{\frac{-3}{10}, \cdots, \frac{-3}{10}}_{7},$$
$$1, \underbrace{\frac{1}{10}, \cdots, \frac{1}{10}}_{7}, \frac{-1}{10}, \frac{-1}{10}, \frac{-1}{10}, 0, \cdots, 0)$$

For each of the four simulation set-ups, we generate 100 simulated datasets. Each simulated dataset consists of a training set and a test set. For each simulated dataset, the linear model is fitted on the training set, which is then used to make the predictions on the test set for the hold-out validation. The performance measures *precision*, *recall*, F-score, $corR$ and $RMSE$ are calculated based on the training set, while $PMSE$ is calculated on the test set. We compare the performance of eGBL with four regularized regression

methods including Lasso, ElasticNet, Grace, and GBL. For both GBL and eGBL, we perform them with two choices of parameter γ, such as GBL2 and eGBL2 with $\gamma = 2$, and GBLinf and eGBLinf with $\gamma = \infty$. Table 5.2 summarizes the simulation results for the four simulation set-ups and Figure 5.2 provides the intuitive comparisons based on the results.

For all four set-ups, eGBL achieves the comparable superior performance as GBL in terms of *precision*, which outperforms Lasso, ElasticNet and Grace. Compared with GBL, eGBL introduces the edge weights of the network into the penalty function, which results in obvious increases in *recall*. It suggests that eGBL gains stronger power for identifying more true feature genes. Moreover, the slight increases in $corR$ and the decreases in the prediction errors also suggests that incorporating edge weights into the regularization can lead to more accurate fitted linear models.

GBL has been reported suffering a strong bias in prediction errors compared with Lasso, ElasticNet and Grace [52]. In our simulation study, we observe the same bias in $RMSE$ but not in $PMSE$, that is, the methods Lasso, ElasticNet and Grace provide relatively low prediction errors on the training set but higher ones on the test set. The big differences between $RMSE$ and $PMSE$ imply the strong overfitting in the linear models fitted by Lasso, ElasticNet and Grace. Thus these methods are inefficient for the variable selection problem in the high-dimensional transcriptome data.

The simulation set-up 2 is simulated with 20% true feature genes, which is higher than the proportions of true feature genes in the other three set-ups. Both GBL and eGBL provide superior performances in *precision* in set-up 2, but they suffer dramatical decreases in *recall* compared with the other three set-ups. It suggests that the power of eGBL drops as the proportion of the feature increases. However, in practice, on the basis of the assumption that there are a very small portion of genes relevant with the response, eGBL suits the tasks of feature gene selection in transcriptome data.

In the simulation set-up 4, we take into account the situation that the coefficients of two linked genes in the network have opposite signs, which is corresponding to the negative correlations in gene co-expression network. The *precision* of eGBL is slightly lower than GBL, but it is not greatly as compared with the gains in *recall*. Thus, eGBL is capable of capturing the negative correlations in the network.

Through the simulation study, we observe that eGBL with the parameter $\gamma = \infty$ offers the overall best performance. As a result, we apply eGBLinf to the real data to select the time-lapse parameter associated genes.

Table 5.2: Results of the simulation study. The performance measures are calculated by the mean of 100 simulations, where standard errors are given in parentheses.

Set-up	Method	*precision*	*recall*	F-score	*corR*	*RMSE*	*PMSE*
Set-up 1	Lasso	0.447(0.159)	0.221(0.064)	0.280(0.061)	0.565(0.119)	8.40(5.79)	22.73(3.29)
	ElasticNet	0.435(0.123)	0.311(0.06)	0.351(0.054)	0.598(0.089)	7.24(5.45)	22.77(2.78)
	Grace	0.681(0.326)	0.293(0.196)	0.293(0.071)	0.576(0.084)	17.13(9.85)	26.87(3.19)
	GBL2	0.988(0.041)	0.366(0.196)	0.503(0.215)	0.641(0.180)	26.72(5.85)	29.90(5.46)
	eGBL2	0.992(0.018)	0.440(0.204)	0.580(0.202)	0.700(0.117)	25.02(5.74)	28.38(5.41)
	GBLinf	0.971(0.061)	0.466(0.198)	0.601(0.207)	0.706(0.173)	25.19(5.61)	28.00(5.39)
	eGBLinf	0.973(0.052)	0.536(0.172)	0.672(0.161)	0.752(0.122)	23.40(4.98)	26.35(5.20)
Set-up 2	Lasso	0.512(0.128)	0.157(0.052)	0.232(0.065)	0.385(0.102)	12.11(11.16)	39.62(5.39)
	ElasticNet	0.518(0.096)	0.211(0.051)	0.293(0.054)	0.424(0.087)	10.69(9.23)	38.95(4.87)
	Grace	0.522(0.186)	0.320(0.213)	0.316(0.106)	0.445(0.092)	17.15(13.00)	40.50(4.47)
	GBL2	0.975(0.068)	0.166(0.124)	0.266(0.163)	0.412(0.166)	44.96(6.26)	48.47(4.74)
	eGBL2	0.971(0.042)	0.201(0.120)	0.316(0.147)	0.470(0.124)	43.10(6.25)	47.86(5.01)
	GBLinf	0.971(0.069)	0.182(0.110)	0.292(0.154)	0.417(0.168)	46.13(5.11)	49.29(4.66)
	eGBLinf	0.975(0.044)	0.221(0.105)	0.348(0.137)	0.490(0.124)	44.46(5.28)	48.07(4.96)
Set-up 3	Lasso	0.438(0.166)	0.173(0.052)	0.235(0.058)	0.632(0.205)	6.72(3.60)	14.34(1.84)
	ElasticNet	0.439(0.145)	0.255(0.062)	0.309(0.061)	0.599(0.139)	5.99(3.67)	14.59(1.87)
	Grace	0.410(0.365)	0.386(0.287)	0.211(0.123)	0.418(0.196)	7.76(7.91)	17.41(2.98)
	GBL2	0.967(0.059)	0.466(0.203)	0.598(0.201)	0.813(0.172)	14.25(3.21)	16.31(3.12)
	eGBL2	0.950(0.078)	0.516(0.187)	0.643(0.165)	0.850(0.096)	13.10(2.89)	15.28(2.78)
	GBLinf	0.961(0.069)	0.492(0.200)	0.622(0.194)	0.788(0.202)	14.49(2.94)	16.50(2.94)
	eGBLinf	0.963(0.044)	0.551(0.181)	0.681(0.160)	0.845(0.127)	13.52(2.55)	15.52(2.88)
Set-up 4	Lasso	0.416(0.187)	0.151(0.053)	0.205(0.057)	0.553(0.201)	6.63(3.65)	14.01(1.70)
	ElasticNet	0.404(0.145)	0.213(0.058)	0.264(0.058)	0.531(0.146)	6.32(3.66)	14.22(1.70)
	Grace	0.505(0.379)	0.284(0.251)	0.191(0.117)	0.445(0.177)	8.62(7.22)	16.26(2.20)
	GBL2	0.960(0.072)	0.379(0.191)	0.512(0.190)	0.753(0.140)	13.75(2.48)	15.41(2.21)
	eGBL2	0.946(0.086)	0.426(0.195)	0.555(0.183)	0.772(0.105)	12.94(2.42)	14.72(2.13)
	GBLinf	0.928(0.106)	0.448(0.186)	0.577(0.182)	0.729(0.165)	13.31(2.40)	15.23(2.16)
	eGBLinf	0.929(0.085)	0.485(0.171)	0.614(0.156)	0.777(0.111)	12.83(2.09)	14.66(2.09)

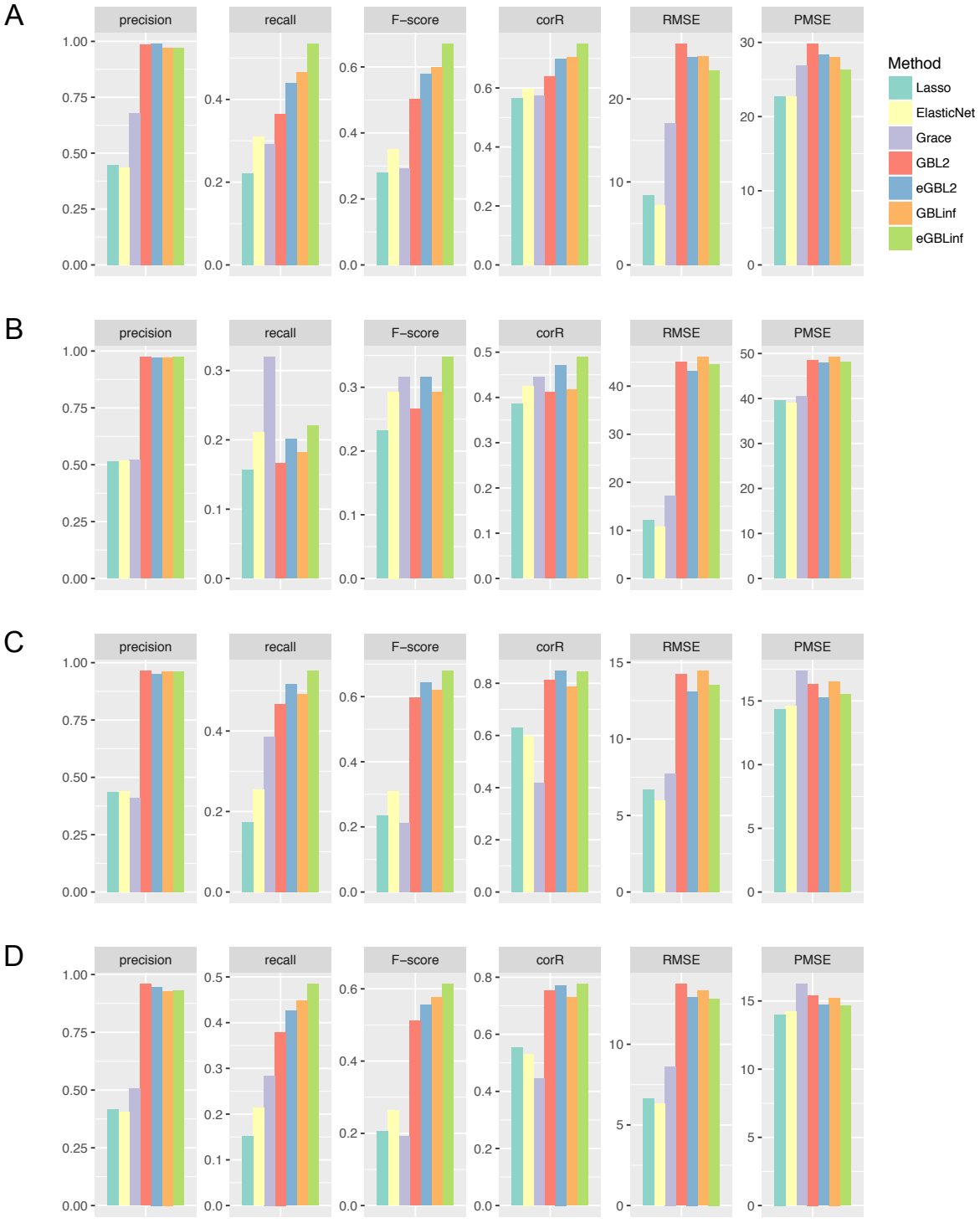


Figure 5.2: Barplots of the results of the simulation study. (A) Simulation set-up 1; (B) Simulation set-up 2; (C) Simulation set-up 3; (D) Simulation set-up 4.

5.4.2 Real data study

5.4.2.1 Selection of time-lapse parameter associated genes

We apply eGBL on the EmbryoScope blastocyst transcriptomes to select feature genes associated with each time-lapse parameter listed in Table 5.1. Here, the gene functional network incorporated in eGBL refers to the gene co-expression network of human pre-implantation embryonic development constructed in Chapter 3. The co-expression network is constructed by assigning the Pearson Correlation Coefficient (PCC) of gene co-expression across multiple development stages as the weight to each edge in the gene-gene interaction network formed by the interactome data.

Because the number of genes is exponentially larger than the sample size in the transcriptome data, we firstly screen the genes to reduce the dimension of the feature space. For each time-lapse parameter, the correlation between the parameter and each gene is evaluated by the Spearman's Rank Correlation Coefficient (SCC). The genes significantly correlated with the parameter (SCC p-value ≤ 0.05) are selected as the candidate gene set. eGBL is then applied to the gene expression data of the candidate gene set to select the feature genes that are associated with the corresponding time-lapse parameter. Table 5.3 shows the numbers of candidate genes and feature genes selected for each time-lapse parameter.

Table 5.3: Numbers of candidate genes and feature genes for each time-lapse parameter.

Parameter	# candidate genes	# feature genes
CC2	813	15
CC3	237	22
S2	408	19
T5	267	19

In order to explore the functional involvement of the time-lapse parameters, we perform the Gene Ontology (GO) enrichment with the feature genes for each time-lapse parameter. The significantly enriched GO biological process terms indicate the potential associations between the time-lapse parameter and these function terms, which helps to understand the crucial roles of the time-lapse parameter in human pre-implantation embryo development. The feature genes and the enriched GO biological processes for time-lapse parameters CC2, CC3, S2 and T5 are shown in Figure 5.3, 5.4, 5.5 and 5.6, respectively.

The enriched biological processes obtained through the function enrichment analysis are involved in several crucial functions related with embryonic development such as "regulation of transcription", "cell cycle", "metabolic process", "viral life cycle", "signal transduction" and "histone modification". The parameter associated genes selected by eGBL reveal some interesting biological insights.

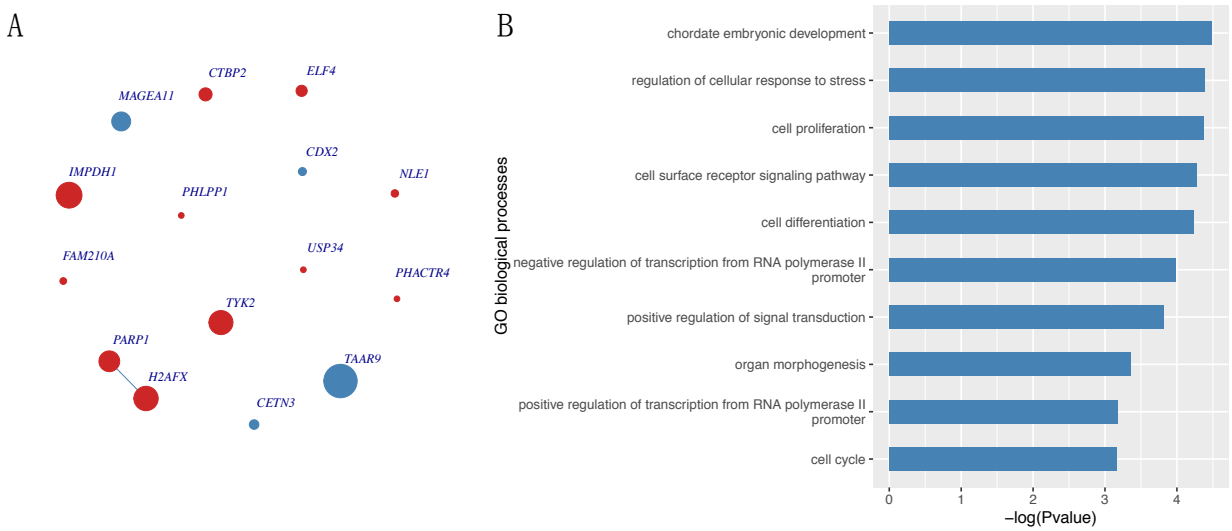


Figure 5.3: Feature genes associated with time-lapse parameter CC2. (A) The feature genes. Each node represents a gene and the node size is proportional to the absolute value of its regression coefficient. Nodes in red represent the positively associated genes (with $\beta > 0$) and nodes in blue represent the negatively associated genes (with $\beta < 0$). Each edge represents the co-expression interaction between two genes and the edge width is proportional to the co-expression correlation. Edges in red represent the positive co-expression and edges in blue represent the negative co-expression. (B) Gene Ontology biological processes enriched by the feature genes.

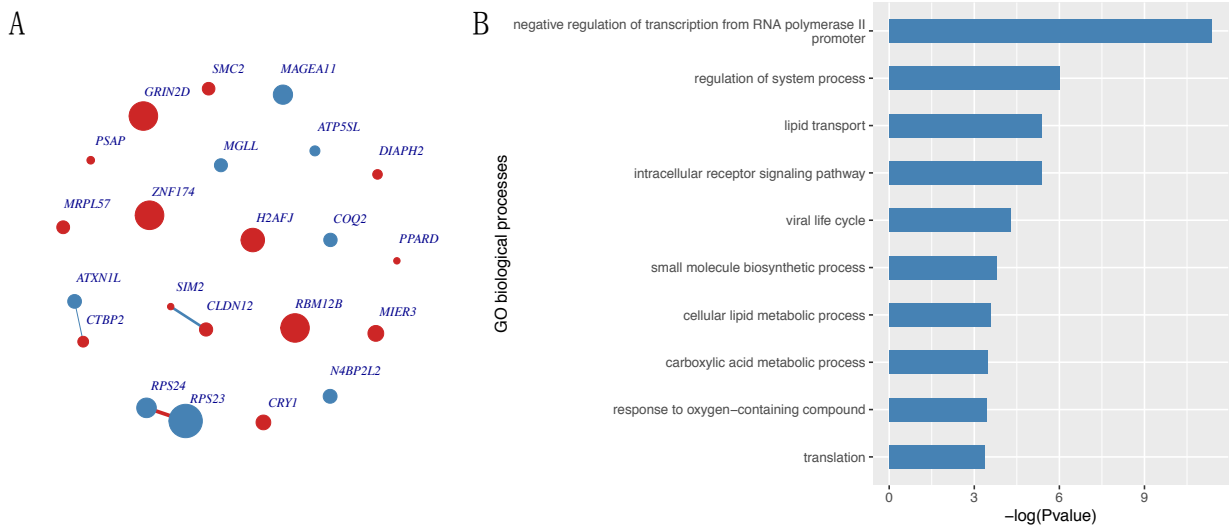


Figure 5.4: Feature genes associated with time-lapse parameter CC3. Figure legends are the same as Figure 5.3.

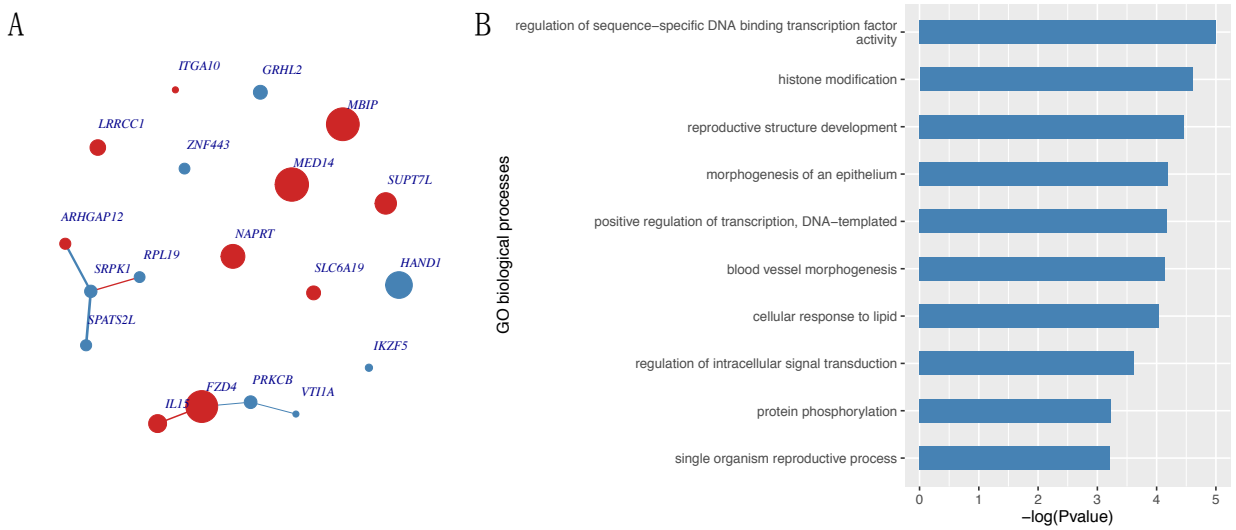


Figure 5.5: Feature genes associated with time-lapse parameter S2. Figure legends are the same as Figure 5.3.

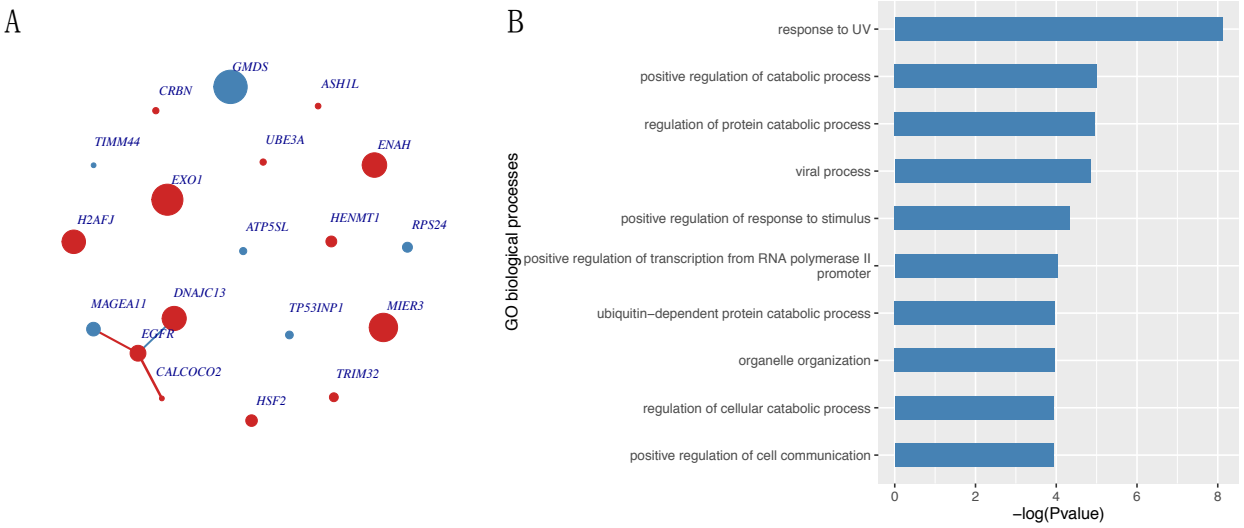


Figure 5.6: Feature genes associated with time-lapse parameter T5. Figure legends are the same as Figure 5.3.

Several genes have been reported to play important roles in early stage of pre-implantation embryonic development which is the period from oocyte to 8-cell stage, such as HSF2 [168], H2AFX [169], H2AFJ [170], DIAPH2 [171], ELF4 [172] and RPL19 [173]. The evidence indicates the crucial roles of the key time-lapse parameters during the early pre-implantation embryonic development.

By contrast, some other genes have been proved associated with the late stage of embryonic development, such as PARP1 [174], CDX2 [175], ATP5SL [176, 177], PPARD [178], GRHL2 [179], HAND1 [180] and EGFR [181]. Dysfunction of these genes will result in low qualities in blastocyst, trophoblast and placenta, which might reduce the success rate of pregnancy during the IVF treatment. The association between the key time-lapse parameters and the late stage embryo development related genes accounts for why the parameters can be used to predict the blastocyst qualities and provides clues for understanding the underlying molecular mechanisms.

5.4.2.2 Characterization of time-lapse parameter associated genes in the gene co-expression network of human embryonic development

We incorporate the gene co-expression network of human pre-implantation embryonic development constructed in Chapter 3 into the selection of feature genes associated with time-lapse parameters as prior functional relationships of genes. In order to understand the roles of the selected feature genes during human embryonic development, we annotate them to the gene co-expression network taking into account the development stage-specific functional modules identified in Chapter 3.

In Chapter 3, 42 functional gene modules are identified as embryonic development stage-specific modules from the co-expression network. 14, 17, 16 and 14 modules out of the 42 modules are specific for oocyte, 4-cell, 8-cell and blastocyst stages, respectively. In the co-expression network, genes located between the 42 stage-specific modules are referred to as inter-module genes. If an inter-module gene has at least two links to genes belonging to a module, the inter-module gene is defined as crosstalking with the module. Such crosstalks between genes and modules suggest potential functional associations between them. The inter-module genes crosstalking with more than one module are selected as pivot genes. Pivot genes that crosstalk with modules associated with different development stages tend to play regulatory roles during the development of embryos across the involved stages.

On the basis of the stage-specific modules and the pivot genes, we annotate the time-lapse parameter associated genes to the co-expression network, and get the following interesting topological characteristics:

- The S2 associated gene, PRKCB, is located within a 4-cell stage-specific module.

- The T5 associated gene, EGFR, is located within a 4-cell stage-specific module.
- The CC3 associated gene, RPS24, is a pivot gene crosstalking with oocyte, 4-cell, 8-cell, blastocyst stage-specific modules.
- The S2 associated genes, RPL19 and SRPK1, are pivot genes crosstalking with 4-cell, 8-cell, blastocyst stage-specific modules.
- The T5 associated gene, EXO1, is a pivot gene crosstalking with oocyte, 4-cell, 8-cell, blastocyst stage-specific modules.
- The T5 associated gene, HSF2, is a pivot gene crosstalking with 4-cell, 8-cell, blastocyst stage-specific modules.
- The T5 associated gene, RPS24, is a pivot gene crosstalking with oocyte, 4-cell, 8-cell, blastocyst stage-specific modules.
- The T5 associated gene, UBE3A, is a pivot gene crosstalking with oocyte, 4-cell, blastocyst stage-specific modules.

For the up-mentioned time-lapse parameter associated genes, their key topological characteristics in the co-expression network suggest potential critical roles involved in human pre-implantation embryo development. It provides clues for exploring the underlying molecular mechanisms that account for the impacts of early stage time-lapse parameters on late stage embryos (i.e., blastocysts). We take the T5 associated gene UBE3A for a case study. UBE3A is a pivot gene crosstalking with three embryonic development stage-specific modules in the co-expression network, as show in Figure 5.7. It has been reported that UBE3A is expressed throughout human pre-implantation development [182, 183] and is associated with neurodevelopmental and metabolic disorders in human [184]. As shown in Figure 5.7, we find UBE3A crosstalks with three modules which are specific for oocyte stage, 4-cell stage and blastocyst stage respectively. It is concordant with the aforementioned scientific evidence. The crosstalks between UBE3A and the oocyte/4-cell specific modules suggest the underlying mechanisms explaining the key roles of the parameter T5 during the early stage of pre-implantation embryonic development, while the crosstalks with the blastocyst specific module help to understand the impacts of T5 on blastocysts at the molecular level.

5.5 Implementation

The data sources of the transcriptome and interactome data studied in this chapter are described in Chapter 5.2.1. The proposed network-constrained regression method,

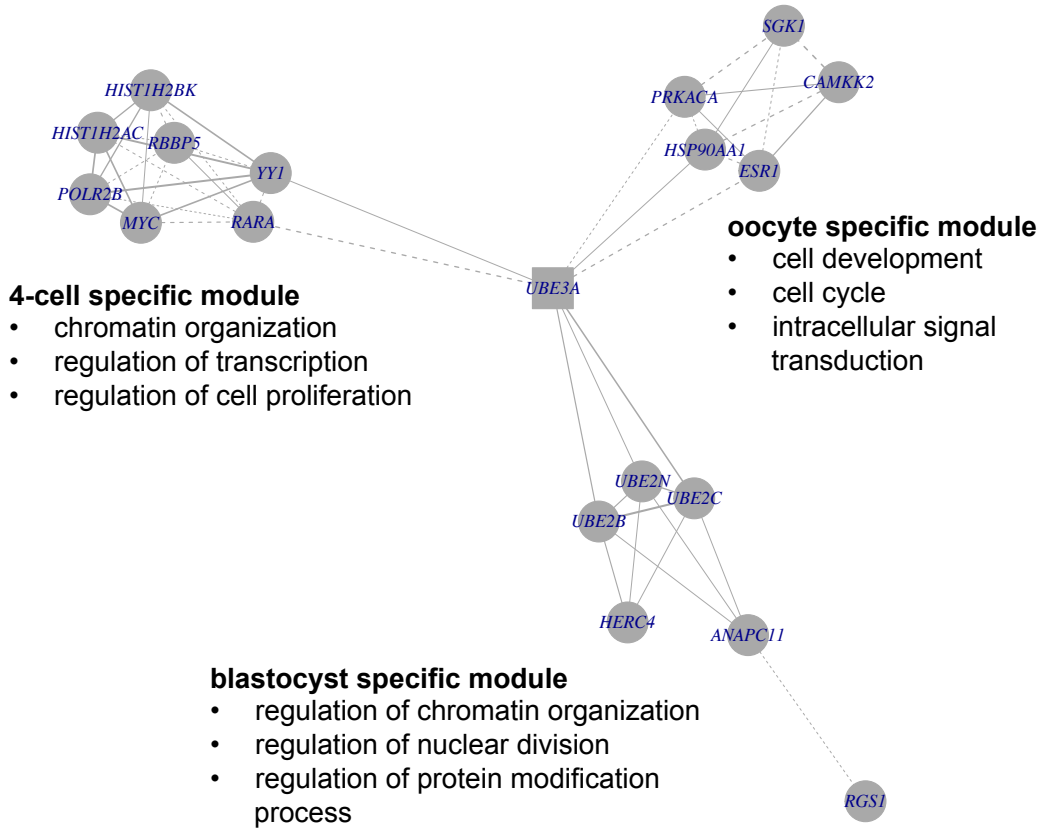


Figure 5.7: Crosstalks between pivot gene UBE3A and embryonic development stage-specific modules. Nodes in square represent pivot genes and nodes in circle represent genes within modules. Edges in solid lines represent positive co-expression and edges in dashed lines represent negative co-expression.

eGBL, is implemented in R. We perform the simulation study and real data study using eGBL. The source codes for eGBL, simulation study and real data study can be accessed from `https://github.com/bioinfoxh/eGBL`. Because the manuscript of this study is in preparation for submission, this gene expression dataset is not public at the moment, but we will upload the processed datasets to GitHub after we submit the manuscript.

5.6 Summary

In this chapter, we propose an approach for feature selection by integrating transcriptome and interactome data using network-constrained regression. We develop a more efficient network-constrained linear regression method, named eGBL, by incorporating the edge weights into the GBL network-constrained penalty, which takes the advantage of weighted network. We evaluate the performance of eGBL on four simulated datasets built with different proportions of features, different magnitudes of coefficients and different signs of coefficients. We show that eGBL outperforms several common regularized regression

methods and provides superior performance on feature selection.

We apply eGBL to explore whether the key time-lapse parameters capable of predicting EmbryoScope blastocyst qualities are associated with transcriptional patterns. For each time-lapse parameter, we apply eGBL to the transcriptome data of blastocysts to select the feature genes by fitting the linear model incorporating the human pre-implantation embryonic development network for regularization. We find scientific evidence that several selected feature genes play important roles across the stages of embryonic development. The early stage associated feature genes indicate the crucial roles of the key time-lapse parameters during the early pre-implantation embryonic development. The late stage associated feature genes account for the prediction capability of key time-lapse parameters on blastocyst qualities from the molecular level.

This chapter provides an efficient approach using network-constrained regression for transcriptome and interactome data integration in the field of multi-omics. The proposed network-constrained regression method, eGBL, shows superior capability for selecting feature genes related with responses from transcriptome data taking into account the network structured relationship between genes. It keeps comparable *precision* in feature selection as the best state-of-art methods, but improves the power. We believe eGBL is a valuable tool for the field of multi-omics, which helps researchers to obtain clues for the underlying mechanisms through data mining from transcriptome data combined with interactome data.

There are still room for further improvement of eGBL. eGBL offers superior performances for feature selection in term of *precision*, but suffers a strong bias in *recall* especially in the dataset with a higher proportion of features, which means that eGBL can provide a correct feature set but missing a certain part of the actual positive features. More efforts will be put in improving the penalty function to gain more power of network-constrained regression methods which is capable to recall more actual positive features without sacrifice in the *precision*.

Chapter 6

Superlayer neural network for epigenome and transcriptome integration

6.1 Introduction

Epigenetic modifications, such as DNA methylation and histone modifications, are reversible and heritable modifications on the DNA in a cell that can affect gene expression without changing the DNA sequence. DNA methylation, a process that occurs by the addition of a methyl (CH3) group to DNA molecular, has been proved playing important roles in regulating gene expressions [22]. Furthermore, increasing scientific evidence clearly indicates that aberrant DNA methylation can result in human diseases such as cancer and ageing related disorders [185]. However, the complex mechanisms of how DNA methylation regulates gene expression is still poorly understood and it is challenging for the systematic integration between transcriptome data and epigenome data, particularly, the DNA methylome data.

A big challenge for multi-omic data integration lies in the complexity and heterogeneity of different types of omic data. So far, the complex regulation mechanisms between different molecular levels have not been sufficiently characterized yet. As a result, the relationships between the omic data corresponding to the molecular levels are still not well understood, which might lead to deficiency in developing statistical or machine learning models for integrating the heterogeneous datasets because of lack of proper assumptions for appropriate models.

Recent advances in deep learning have impacted various scientific and industrial fields including computational biology [186, 187]. Artificial neural networks have shown superior performance in supervised learning with omic data, e.g., using genes as features to predict the clinical outcome for patients [188, 189]. Taking advantage of the non-linear modelling

capacity, neural networks have also been well-suited for multi-omic data integration, which has been successfully applied in cancer diagnosis by integrating multiple omic data such as gene expression, DNA methylation, gene copy number alteration and miRNA expression [190, 191, 192]. It is of great importance to figure out how to apply neural network models to integrate multi-omic data in effective and efficient ways. In collaboration with a PhD student, Petar Velickovic, and a Part II student, Ioana Bica, in the Computer Laboratory, we aim to explore the applicability of neural networks to multi-omic data integration, in particular, the integration of transcriptome and DNA methylome data in this chapter.

6.2 Multi-omics

6.2.1 Multi-omic data

6.2.1.1 Epigenome

The epigenome data studied in this chapter is the DNA methylation data of human breast cancer [193], including 760 cancer and 84 normal samples. The DNA methylome data are measured by the Illumina Human Methylation 450k BeadChip (450K array) and the preprocessed probe methylation levels are downloaded from The Cancer Genome Atlas (TCGA) [194]. We only keep the probes mapped to the body region of each gene which is defined as the region from the 201th nucleotide until the last nucleotide in the sequence of the gene. The average value of the gene body probes is calculated as the methylation level for each gene.

6.2.1.2 Transcriptome

The breast cancer transcriptome data for the matched samples of the DNA methylome data are also downloaded from TCGA. The transcriptome data are measured by RNA-seq experiments and the FPKM values calculated from the RNA-seq data are downloaded and used as the expression abundances for each gene.

6.2.2 Problem definition

A proper and efficient neural network model is critical for DNA methylome and transcriptome data integration for supervised learning problems such as classification. Because of the differences in biological mechanisms and the measuring techniques, DNA methylation and gene expression data exhibit different structures. The raw abundances of gene expression and DNA methylation follow different distributions [195] and such heterogeneity between the two types of omic data should be take into account when developing statistical and machine learning models to integrate them [196]. Besides, DNA methylation and

gene expression are not independent because it is reported that DNA methylation play regulatory roles on gene expression. It thus suggests that there are potential correlated relationships between the two types of data, particularly, between the gene body DNA methylation and gene expression which has been reported is not a simple linear correlation and still poorly understood [197, 198, 199].

A few neural network models have been developed to integrate different types of omic data for human cancer classification. The research problem of this chapter is to compare the performances of existing neural network models as well as other common machine learning methods for cancer classification by integrating DNA methylation and gene expression data. Based on the comparative study, we aim to improve the existing neural network models and propose more efficient models for integration of DNA methylation and gene expression data.

6.3 Methodology

6.3.1 Neural network models

On the basis of existing neural network models, there are two ways to integrate the input multiple omic datasets: series integration and parallel integration.

For the series integration, the multiple omic datasets are simply stacked together by sample and then the stacked dataset is used as a whole input set for a neural network model. For example, Chaudhary et al. [191] developed a feedforward neural network with stacked gene expression, DNA methyaltion and miRNA expression to predict survivals of liver cancer.

For the parallel integration, the neural network architecture is built based on a hierarchy structure which consists of several superlayers of neural networks. The hierarchy structure is similar as the multilayer network framework in the field of network science. Each type of omic data is considered as the input set for a specific superlayer to be learned in the corresponding neural network independently. The final predicted output is summarized from the outputs of all the superlayers. For example, Sun et al. [190] proposed a multimodal deep neural network for breast cancer prognosis prediction, which consists of three independent superlayers of neural networks corresponding to gene expression, copy number alteration and clinical data, respectively, as well as a final output layer fusing the outputs from the three superlayers. Liang et al. [192] developed another multimodal deep neural network for ovarian cancer detection based on three omic datasets (gene expression, DNA methylation and miRNA expression), where the three superlayer neural networks were merged into a hidden layer followed by an output layer for the entire model. Besides, Velickovic et al. [200] proposed a cross-modal convolutional neural network (X-CNN)

inspired by the multilayer network framework, which consists of superlayer convolutional neural networks capable of multiple type data integration. X-CNN allows the information flowing between the independent superlayers through the cross-connections between them. It also introduces a merge-layer that consists of two layers of fully connected feedforward neurons before the final output layer of the whole model. The cross-modal neural network architecture is a generalized structure for parallel integration as the cross-connections can be set into different hidden layers.

Following the up-mentioned series integration strategy, we apply the feedforward neural network model, also know as multilayer perceptron (MLP), illustrated in Figure 6.1, to integrate DNA methylation and gene expression for cancer classification. For the parallel integration strategy, the X-CNN model is not suitable for multi-omic data integration. X-CNN is developed based on the convolutional neural network which is used for the image data with two-dimensional features, and it is thus not suitable for the omic data with one-dimensional features. Consequently, we propose a new cross-modal neural network model based on feedforward neural networks for DNA methylation and gene expression integration, which is referred to as superlayer neural network (SNN), illustrated in Figure 6.2. SNN is the first cross-modal neural network that is developed for multi-omic data integration.

6.3.1.1 MLP

A multilayer perceptron (MLP) is built for breast cancer classification by series integration of gene expression and DNA methylation features. It consists of an input layer, four fully connected hidden layers and an output layer, illustrated in Figure 6.1.

Input layer In the input layer, the DNA methylation features and gene expression features are stacked together by sample, and passed on to the hidden layers by the 52 nodes within the layer.

Hidden layers The four hidden layers consist of 256, 128, 64 and 32 neurons, respectively, which are fully connected between the adjacent layers. The ReLU activation function (see Equation 2.30) is chosen for the neurons in the hidden layers.

Output layer The output layer consists of two neurons and uses the softmax transformation (see Equation 2.31) as the activation function to provide a probability distribution over the possible classes.

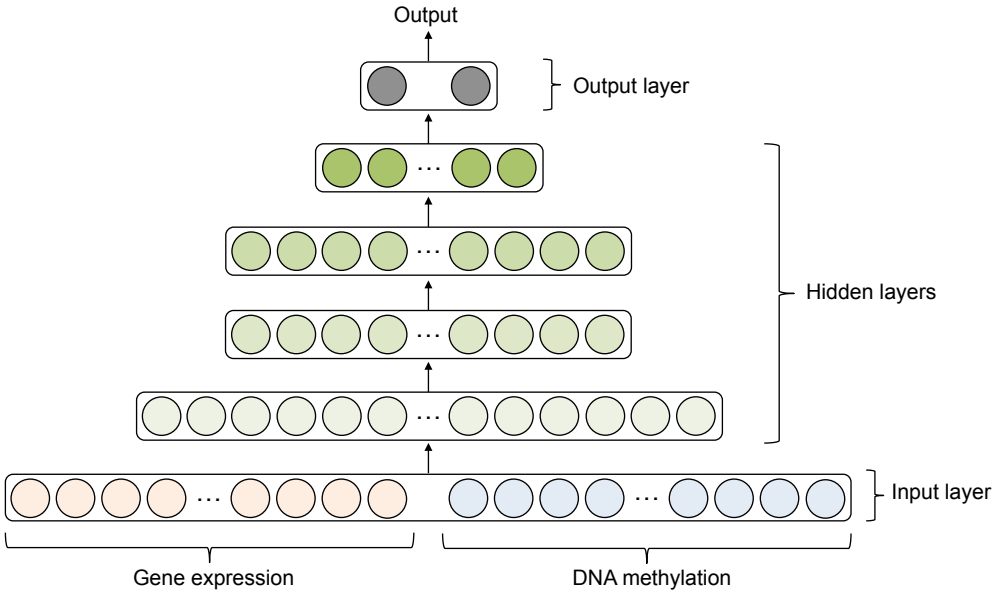


Figure 6.1: Illustration of the MLP structure. The MLP consists of an input layer, four hidden layers and an output layer. Each node represents a neuron. The nodes in orange in the input layer represent the gene expression features and the nodes in blue represent the DNA methylation features. The nodes in green represent the hidden layers. The nodes in grey represent the output layer. The arrows indicate the data flow in the MLP.

6.3.1.2 SNN

The generalized superlayer neural network (SNN) model is illustrated in Figure 6.2, in which the cross-connections can be assigned at different hidden layers between two superlayers according to the task requirements. The structure shown in Figure 6.2 is a SNN model with cross-connections at the third hidden layers between two superlayers, which we use for the breast cancer classification task. It is comprised of four parts: the input layers, the superlayers, the merge layers and the output layer.

Input layers The SNN includes two independent input layers, which receive the gene expression features and DNA methylation features, respectively, and pass on them to the corresponding superlayers.

Superlayers The SNN includes two superlayers, each of which consists of a fully connected feedforward neural network including four hidden layers with 128, 64, 32, 16 neurons respectively. The gene expression features and DNA methylation features are learned by the two superlayers separately. In order to exchange the information between the two superlayers, cross-model connections are added at the hidden layers between the superlayers, which allow the information to flow between them after several layers independent learning. Between the two four-hidden-layered superlayers, there are three optional locations to add the cross-model connections: at the second hidden layers, at

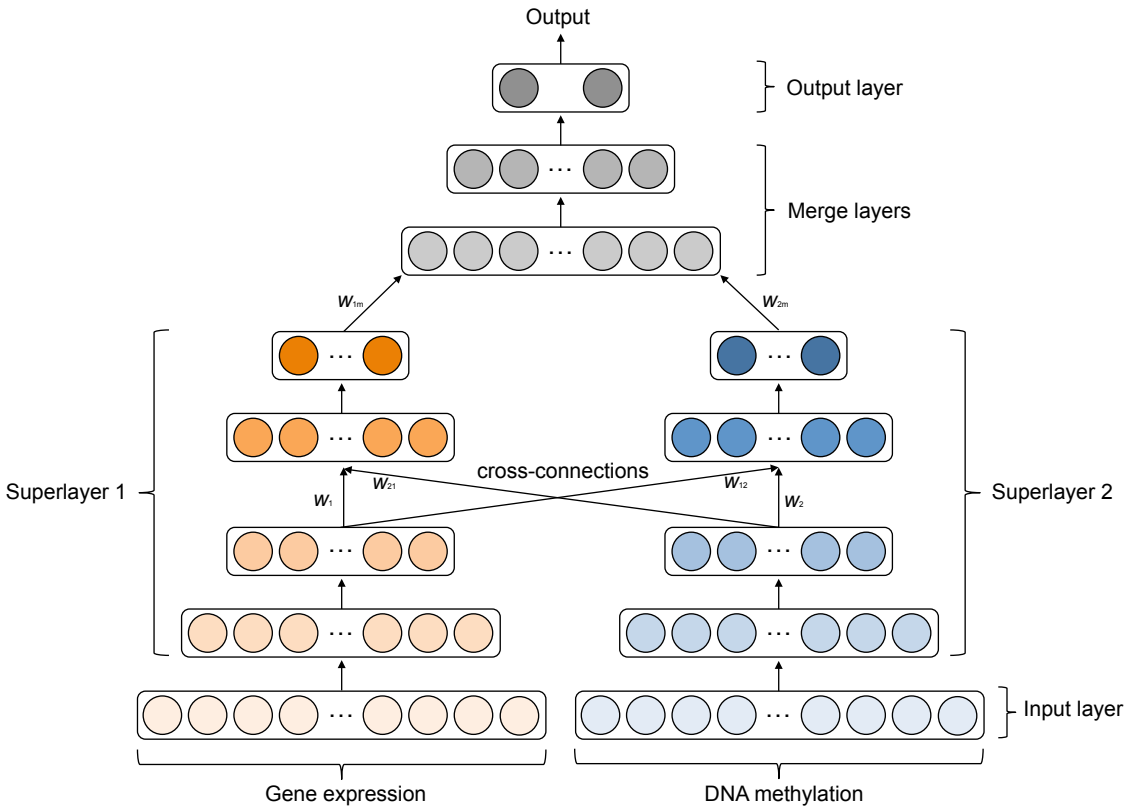


Figure 6.2: Illustration of the SNN structure with cross-model connections at the third hidden layers between the superlayers. The SNN consists of two input layers, two superlayers, two merge layers and an output layer. Each node represents a neuron. The nodes in orange represent the superlayer for gene expression and the nodes in blue represent the superlayer for DNA methylation. The nodes in grey represent the merge layers and the output layer. The arrows indicate the data flow in the SNN. The arrows between the two superlayers represents the cross-connections.

the third hidden layers, and at the fourth hidden layers. As illustrated in Figure 6.2, the cross-model connections are added at the third hidden layers. The operations performed by the neurons in the third hidden layer based on the cross-connections are defined as follows:

$$\begin{aligned} y_1 &= w_1 x_1 + w_{21} x_2 + b_1 \\ y_2 &= w_2 x_2 + w_{12} x_1 + b_2 \end{aligned} \tag{6.1}$$

where x_1 and x_2 are the outputs of the second hidden layers from superlayer 1 and superlayer 2 respectively, and y_1 and y_2 are the outputs of the third hidden layers correspondingly. w_1 and w_2 are the within superlayer weights, and b_1 and b_2 are the within superlayer biases for the third hidden layer. w_{12} and w_{21} are the cross-connection weights.

Merge layers Following the superlayers, the merge layers consist of two fully connected hidden layers with 64 and 16 neurons respectively. The information from the two superlayers are combined together in the first hidden layer, where the operations of the neurons are

defined as:

$$y_m = w_{1m}x_{1m} + w_{2m}x_{2m} + b_m \tag{6.2}$$

where x_{1m} and x_{2m} are the outputs of the two superlayers. w_{1m} and w_{2m} are the combined weights for superlayer 1 and superlayer 2 respectively. b_m is the bias for the first hidden layer, and y_m is the output of the first hidden layer. In both the merge layers and the superlayers, the ReLU activation function is chosen for the neurons.

Output layer The output layer is the same as the MLP model, which consists of two neurons and uses the softmax transformation as the activation function to provide a probability distribution over the possible classes.

6.3.2 Neural network model training and evaluation

To train the MLP and SNN models for the classification task, the feature set is selected as the 26 genes of the tumour necrosis factor receptor superfamily (TNFRS) which has been proven to play important roles during tumorigenesis and could be the potential targets for cancer therapy [201]. These genes are responsible for production of receptors that are able to bind to tumour necrosis factors (TNFs), proteins capable of inducing cell death (apoptosis) of tumorous cells, and their activity levels are thus expected to be correlated with the incidence of specific cancers. The gene expression and DNA methylation of the TNFRS genes are used as the inputs for the neural network models.

The MLP and SNN models are learned by backpropagation with the gradient descent optimisation (see details in Chapter 2.3.2.2). The loss function is defined by the cross entropy loss between the predicted outcome $\hat{y}_i$ and the actual outcome y_i with a weight decay regularization as follows:

$$L(w) = -\sum_{i=1}^{k} y_i \log(\hat{y}_i) + \frac{\lambda}{2}||w||_2 \tag{6.3}$$

where $||w||_2$ denotes the L_2-norm of the weights and λ is the tuning parameter for the weight decay. The L_2-norm regularization provides the smooth solution for weight estimation which avoid the exploding of weights by penalizing the large ones.

The optimal parameters for each neural network model are trained by stratified nested 5-fold cross-validation (see details in Chapter 2.3.5.1) by evaluating their performance measures *accuracy*, *precision*, *recall* and F-score (see details in Chapter 2.3.5.2).

6.3.3 Comparison with other common classifiers

We compare the performances of neural network models with some other common classifiers such as Support Vector Machine (SVM) (see details in Chapter 2.3.3) and Random Forest (see details in Chapter 2.3.4). The SVM (with RBF kernel) and Random Forest models are also trained and evaluated on the series integrated DNA methylation and gene expression data using the same methods as the neural network models, that is, stratified nested 5-fold cross-validation.

6.4 Results

6.4.1 Overall performances of classification models

To study the performances of the proposed MLP and SNN models as well as the comparisons with other common classification methods such as SVM and Random Forrest, we train and evaluate the optimal classifier for each of the following models by stratified nested 5-fold cross-validation, and their performances on breast cancer classification are shown in Table 6.1.

- MLP-Expr: MLP model for gene expression data exclusively.
- MLP-Meth: MLP model for DNA methylation data exclusively.
- MLP: MLP model for series integration of gene expression and DNA methylation data.
- SNN-nCC: SNN model without cross-model connections for parallel integration of gene expression and DNA methylation data.
- SNN-CC2: SNN model with cross-model connections at the second hidden layers for parallel integration of gene expression and DNA methylation data.
- SNN-CC3: SNN model with cross-model connections at the third hidden layers for parallel integration of gene expression and DNA methylation data.
- SNN-CC4: SNN model with cross-model connections at the fourth hidden layers for parallel integration of gene expression and DNA methylation data.
- SVM: SVM model for series integration of gene expression and DNA methylation data.
- RandomForest : Random Forest model for series integration of gene expression and DNA methylation data.

Table 6.1: Performances of MLP models, SNN models, SVM and Random Forest on breast cancer classification. The performance measures are calculated by the mean of five test sets in the outer loop of the stratified nested 5-fold cross-validation, where standard errors are given in parentheses.

Models	*accuracy*	*precision*	*recall*	F-score
MLP-Expr	0.983 (0.010)	0.997 (0.003)	0.984 (0.009)	0.991 (0.006)
MLP-Meth	0.982 (0.008)	0.995 (0.003)	0.986 (0.008)	0.990 (0.005)
MLP	0.987 (0.004)	0.997 (0.003)	0.988 (0.005)	0.993 (0.002)
SNN-nCC	0.985 (0.007)	0.995 (0.005)	0.988 (0.009)	0.991 (0.004)
SNN-CC2	0.987 (0.009)	0.997 (0.003)	0.988 (0.009)	0.993 (0.005)
SNN-CC3	0.991 (0.008)	0.999 (0.003)	0.991 (0.010)	0.995 (0.005)
SNN-CC4	0.988 (0.005)	0.997 (0.003)	0.989 (0.007)	0.993 (0.003)
SVM	0.980 (0.007)	0.987 (0.005)	0.991 (0.004)	0.989 (0.004)
RandomForest	0.981 (0.006)	0.984 (0.007)	0.995 (0.003)	0.990 (0.004)

As shown in Table 6.1, all the MLP models, all the SNN models, SVM and Random Forest provide superior performances on breast cancer classification, and achieve classification accuracies higher than 98%. There are two potential reasons for such surprising high accuracies. The first reason is the feature set. In this study, we use the 26 genes of the tumour necrosis factor receptor superfamily (TNFRS) as the feature set for breast cancer classification. The TNFRS has been proven related with cancer progression, which suggests that the expression and methylation values of the TNFRS genes are strongly correlated with the outcome classes of the patients, i.e., cancer vs. normal. Such potential strong correlation between the feature set and the outcome classes is likely to result in the superior performance for classification. The other reason is the biased sample size of the training set. The breast cancer data include 760 cancer and 84 normal samples. Thus, the training set used to train the model is biased in the sample size, in which the cancer class is much larger than the normal class. Such biased training set tends to result in potential preference for cancer class when training the classifier. To evaluate the classification tasks on biased dataset, we need to consider not only the *accuracy* but also the *precision* and *recall*.

Overall, the neural network models (the MLP models and SNN models) perform better than the SVM and Random Forest on breast cancer classification in terms of both *accuracy* and *precision*, but provide a slight decrease in *recall*. The SNN models outperform the MLP models by providing higher *accuaries* and *recalls* as well as comparable *precisions*. Among all the models, the SNN-CC3 provides the best performance on breast cancer classification, with the *accuracy*, *precision* and *recall* all above 99%. It suggests that the parallel integration using SNN models is the more efficient strategy for gene expression and DNA methylation integration than the series integration.

6.4.2 Performances of MLP models

Three MLP models are trained for breast cancer classification. The MLP-Expr performs cancer classification using the gene expression data of the 26 TNFRS genes exclusively, while the MLP-Meth using the DNA methylation data exclusively. The MLP performs cancer classification using the series integration of gene expression and DNA methylation data. As shown in Table 6.1, the MLP-Expr and MLP-Meth provide comparable *accuracies* for cancer classification, but the MLP outperforms them with increased *accuracy* and *recall*. It suggests that the heterogeneous gene expression and DNA methylation data contain intrinsic characteristics which are complementary to each other. Integrating such complementary characteristics of the two types of data using MLP is capable of describing the non-linear relationships between them, which leads to improved performance for cancer classification compared with using the gene expression data or DNA methylation data exclusively.

6.4.3 Performances of SNN models

Four SNN models are trained for breast cancer classification using parallel integration of gene expression and DNA methylation data with cross-connections at different positions: SNN-nCC without cross-connections, SNN-CC2 with cross-connections at the second hidden layers, SNN-CC3 with cross-connections at the third hidden layers, and SNN-CC4 with cross-connections at the fourth hidden layers. As shown in Table 6.1, the SNN models with cross-connections (SNN-CC2, SNN-CC3, SNN-CC4) outperform the SNN model without cross-connection (SNN-nCC). It suggests that the cross-connections exchange the complementary information between the two heterogeneous data during the independent training in each superlayer respectively, which results in better performances than the SNN-nCC without the information exchange. Among the SNN models with cross-connections, the SNN-CC3, exchanging information at the middle of hidden layers, provides the best performance on breast cancer classification, with the *accuracy*, *precision* and *recall* all above 99%. It outperforms the SNN-CC2 and SNN-CC4 which exchange the information at earlier and later hidden layer respectively.

The superior performances of the SNN models compared with the MLP models suggests that parallel integration is more efficient for gene expression and DNA methylation integration than the series integration. For the series integration in the MLP, since the gene expression and the DNA methylation are passed on to the neural network as an entire stacked feature set, the two types of features are nested together in the beginning and throughout the whole learning process, which ignores their intrinsic properties. Due to the heterogeneity of gene expression and DNA methylation data, their intrinsic characteristics provide valuable underlying biological meanings. Ignoring such characteristics might lead

to loss of information in exploring the relationships between them. By contrast, the parallel integration in the SNN takes the advantage of the intrinsic properties of the two types of data. The gene expression features and DNA methylation features are learned separately in the corresponding superlayer, which is capable of remaining the intrinsic characteristics within each superlayer. The cross-connections between two superlayers allow the information exchanging during the independent learning, which takes into account the potential correlations between the two type of features. Therefore, the SNN is capable of capturing more underlying information for gene expression and DNA methylation integration than the MLP.

The cross-connections added at the hidden layers allow the information to flow between two types of data after several layers independent learning. The positioning of the cross-connections in the SNN is crucial for the data integration because it determines to which degrees the two types of data affect each other. In practice, the positioning of the cross-connections are needed to be carefully considered according to the depth of the superlayer as well as the heterogeneous properties of the multiple data. If the cross-connections are assigned to the end of the superlayers as in the SNN-CC4, it will not take the full advantage of the potential correlations between the two types of data. By contrast, if the cross-connections happen too early as in the SNN-CC2, it might result in excessive influence between the data as the two types of features are fused together in the beginning of the superlayer learning, which is similar as the series integration. The SNN-CC3 assigns the cross-connections in the middle of the superlayer in order to gain a balance for the influences between each other. It exchanges information between gene expression and DNA methylation data at the middle of hidden layers, which allows sufficient independent learning before and after exchanging the information.

As a result, the SNN-CC3 with parallel integration strategy is recommended for the neural network based integration of DNA methylome and transcriptome for breast cancer classification tasks. Moreover, the SNN models can be also extended with extra superlayers in order to integrate more types of omic data. We believe that the SNN represents a valuable tool for multi-omic data integration.

6.4.4 Cross-connections in SNN-CC3

In order to understand the contributions of the cross-connections for the classification, the activations of neurons in specific layers of the SNN-CC3 on the samples in the test set are visualized using t-SNE [202], which is a dimensionality reduction technique well-suited for embedding high-dimensional data into a space of two or three dimensions, shown in Figure 6.3. The cross-connections between the two superlayers in the SNN-CC3 happen when the information is passed on from the second hidden layer to the third hidden layer. After the activations of neurons in the third hidden layer, the samples in the test set are separated

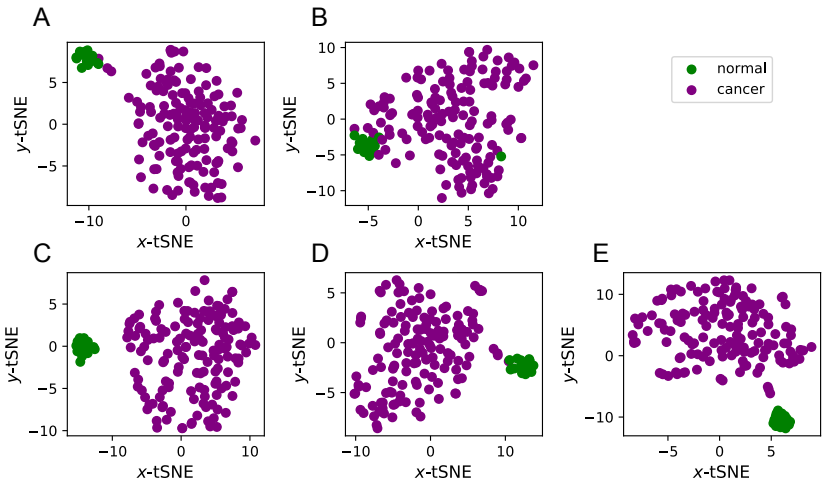


Figure 6.3: t-SNE visualization for the activations of SNN-CC3 layers on the test set. Each node represents a sample in the test set. Nodes in green represent the cancer samples, while nodes in purple represent the normal samples. The scatter plots illustrate the t-SNE visualization for the activations of specific layers in the SNN-CC3: (A) The first hidden layers in the superlayers before cross-connections; (B) The second hidden layers in the superlayers before cross-connection; (C) The third hidden layers in the superlayers after cross-connections; (D) The fourth hidden layers in the superlayers after cross-connections; (E) After the merge layers.

into two clusters corresponding to the cancer samples and the normal samples respectively (Figure 6.3C). Compared with the embedded distributed cancer and normal samples based on the activations of the second layer (Figure 6.3B), the cross-connections lead to a markedly improvement on discriminating cancer samples from the normal samples.

6.5 Implementation

The data sources of the DNA methylome and transcriptome data studied in this chapter are described in Chapter 6.2.1. The neural network models, including MLP-Expr, MLP-Meth, MLP, SCC-nCC, SNN-CC2, SNN-CC3 and SNN-CC4, are implemented using TensorFlow [203] in Python. The SVM and Random Forest are implemented in R using the packages e1071 and randomForest respectively. The processed datasets used in this study and the source codes for these models can be accessed from `https://github.com/bioinfoxh/SNN`.

6.6 Summary

In this chapter, we propose an approach for classification by integrating epigenome and transcriptome using neural networks. We introduce two neural network models for DNA methylation and gene expression integration based on two strategies: (i) the multilayer

perceptron (MLP) for series integration strategy, in which the DNA methylation and gene expression features are stacked together by samples; (ii) the superlayer neural network (SNN) for parallel integration strategy, in which the DNA methylation features and gene expression features are learned separately in superlayers but with cross-connections allowing the crosstalks between them.

We train the optimal MLP and SNN on a breast cancer dataset using stratified nested 5-fold cross-validation and compare their performances on the cancer patients classification. The SNN provides superior performances and outperforms the MLP due to its capability of learning the intrinsic characteristics of the heterogeneous datasets. We compare the neuron activations between the layers before the cross-connections and after the cross-connections in the SNN, and find that the cross-connections lead to a markedly improvement on discriminating the two classes of samples in the latter layer. We recommend the parallel integration strategy (i.e. the SNN) for the neural network based integration of DNA methylome and transcriptome data.

This chapter provides an efficient approach using superlayer neural networks (SNN) for epigenome and transcriptome data integration in the field of multi-omics. It is the first cross-modal neural network model for cancer classification by integrating the DNA methylome and transcriptome data, which outperforms common machine learning classification methods and provides superior performances on human breast cancer classification. We believe the proposed SNN models are useful tools for the field of multi-omics, which helps researchers to understand the underlying mechanisms through data mining from epigenome and transcriptome data.

The proposed SNN models could be extended or improved in the following directions. Firstly, the proposed SNN model can be extended with more superlayers in order to integrate more types of omic data such as the genome copy number variation and the somatic mutations data. Secondly, the SNN model can be extended with more advanced neural network architectures such as RNN and LSTM (see details in Chapter 2.3.2.2), which is able to deal with data with more complex structures such as the sequential data. Finally, neural networks are superior prediction tools but suffer the low interpretability of the non-linear classification decisions. Recently, advanced neural network models have been proposed by incorporating real graph structure into neural network architectures, such as Graph Convolutional Networks (GCN) [204] and GraphSAGE [205]. Inspired by these graph-based neural network models, it is possible to extend the SNN model by incorporating graph structures, which is therefore capable of integrating the molecular networks into SNN to increase the interpretability of the model.

Chapter 7

Conclusion

7.1 Contributions

In the field of multi-omics, the main goal is to integrate heterogeneous high-throughput multiple omic data and transform them into biological knowledge about the underlying mechanisms of biological systems. The biggest challenge for multi-omic data integration is lack of appropriate and efficient machine learning and statistical models to integrate the multiple high-dimensional heterogeneous omic data. It is because of the two inherent characteristics of multi-omic data: high-dimension and heterogeneity. The high-dimension of omic data easily leads to overfitting in machine learning models because the features (i.e., genes) are far more than the samples and the features are usually not independent but correlated to each other. The heterogeneity of different types of omic data brings difficulties for applicabilities of machine learning models because of lack of prior knowledge about the relationships between the intrinsic characteristics of different types of omic data. There is a famous quote "All models are wrong but some are useful" which is attributed to the famous statistician George Box [206]. The quote is considered to be applicable to not only statistical models but to scientific models generally. Following this quote, the main task of multi-omic studies is to find and apply the "useful models" to integrate different types of omic data taking into account their intrinsic characteristics. This thesis builds a bridge between computer science and biology by developing efficient approaches using appropriate machine learning models for data mining through multi-omic data integration, which are capable of transforming multi-omic data into underlying biological insights. Following the hypothesis stated in Chapter 1.2, this thesis has made contributions on methodology development for multi-omic data integration with focus on four specialized topics.

We address the unsupervised learning problem of identifying condition responsive gene functional modules based on transcriptome and interactome data. Current methods for this problem are not well established for multi-condition transcriptome data as they are

developed to identify gene modules based on differential expression of genes between case and control samples. Besides, most of the methods only select condition associated modules but ignore the roles of individual genes within the modules. To overcome the limitations, we propose an approach for gene module detection by integrating multi-condition transcriptome data and interactome data using network overlapping module detection method. It is capable of identifying condition-specific modules and important genes within the modules following an efficient and comprehensive computational framework, which consists of four steps: (1) construction of gene co-expression network by evaluating co-expression correlation coefficient between each interacted gene pair based on their gene expression; (2) detection of overlapping gene modules from the co-expression network using network overlapping module detection method; (3) identification of condition-associated modules by assessing the significance of enrichment with condition-associated genes within the modules using ANOVA-GSEA; (4) selection of condition-specific feature modules and feature genes using GEL logistic regression with K-fold cross-validation. We apply the proposed approach on the transcriptome data of human pre-implantation embryos across multiple development stages and identify human embryonic development stage-specific modules and genes. Interesting biological insights are revealed from the dynamic expression patterns of the stage-specific modules and the multiple function genes located in the overlapping modules, which provides clues for understanding the potential molecular mechanisms during human pre-implantation embryonic development.

We address the unsupervised learning problem of identifying translational regulated gene functional modules based on transcriptome, translatome and interactome data. Current methods for this problem mainly focus on gene levels which aim to identify differentially translated genes from transcriptome and translatome data, but none of them are capable of identifying underlying gene functional patterns. We propose a novel approach to identify gene functional modules related with translational regulation by integrating transcriptome, translatome, and interactome data using multilayer network, which consists of five steps: (1) construction of multilayer differential expression network by integrating transcriptome and translatome with interactome data respectively; (2) selection of seed genes for module detection by evaluating their degrees of differential translation; (3) detection of modules from the multilayer network using greedy search for each seed gene by minimizing the entropy-based local modularity function; (4) identification of translation efficiency (TE) regulated modules by the refinements including significance assessment, redundancy deletion and dynamic evaluation; (5) visualization of TE-regulated modules as graphs with incorporated multilayer information from the networks. We also develop an R package, TERM, for implementation of the proposed approach, which is the first tool for identifying translational regulated modules from ribosome profiling data. We apply the proposed approach on a published ribosome profiling data of mTOR perturbed prostate

cancer cells and mine several TE-regulated modules associated with mTOR perturbation. The identified translational regulated genes and modules downstream mTOR provide valuable clues for understanding mTOR associated translational regulation mechanisms in prostate cancer genesis and metastasis.

We address the supervised learning problem of selecting feature genes for scalar responses based on transcriptome and interactome data. Regarding the problem of feature selection from transcriptome data, the high-dimension and strong correlations of the feature set (genes) easily result in overfitting of linear regression models. The common linear regression methods and state-of-art network-constrained regression methods usually suffer low power and high bias for feature selection. To overcome these limitations, we develop a more efficient network-constrained linear regression method, named eGBL, by incorporating the edge weights into the GBL network-constrained penalty, which takes the advantage of weighted network. Simulation studies show that eGBL outperforms several common regularized regression methods and provides superior performance on feature selection. We apply eGBL to explore whether the key time-lapse parameters capable of predicting EmbryoScope blastocyst qualities are associated with transcriptional patterns. For each time-lapse parameter, we use eGBL on the transcriptome data of blastocysts to select the feature genes by fitting the linear model incorporating the human pre-implantation embryonic development network for regularization. We find scientific evidence that several selected feature genes play important roles across the stages of embryonic development. The early stage associated feature genes indicate the crucial roles of the key time-lapse parameters during the early pre-implantation embryonic development. The late stage associated feature genes account for the prediction capability of key time-lapse parameters on blastocyst qualities from the molecular level.

We address the supervised learning problem of disease classification based on epigenome and transcriptome data. Regarding this problem, neural networks have been used to perform patients classification through multi-omic data integration. But there are some limitations for the integration strategies of the state-of-art methods. The series integration strategy, in which the DNA methylation and gene expression features are stacked together by samples, ignores the intrinsic characteristics of epigenome and transcriptome data. The parallel integration strategy, in which the DNA methylation features and gene expression features are learned in separate neural networks followed by an integrated output, ignores the crosstalks between the two types of data. To overcome the limitations, we introduce two neural network models for DNA methylation and gene expression integration based on the two strategies: (i) the multilayer perceptron (MLP) for series integration strategy; (ii) the superlayer neural network (SNN) for parallel integration strategy, in which the DNA methylation features and gene expression features are learned separately in superlayers but with cross-connections allowing the crosstalks between them. We train the optimal

MLP models and SNN models with cross-connections at different hidden layers on a breast cancer dataset for cancer patients classification using stratified nested 5-fold cross-validation. The SNN model with cross-connections at the middle of hidden layers provides superior performances and outperforms the MLP as well as other common classification methods such as SVM and Random Forest due to its capability of learning the intrinsic characteristics of the heterogeneous datasets.

Our proposed approaches primarily contribute to the fields of bioinformatics and computer science. The biological insights obtained from the works presented in this thesis by applying the proposed approaches in multi-omic data of human early embryos and human cancers also provide interesting clues for the field of biology. This thesis offers effective and efficient approaches for integrative analysis of multi-omic data using appropriate machine learning models. Since the approaches are developed to handle heterogeneous high-dimensional datasets, they can be applied to datasets of other fields which present the similar structures. From the computational biology point of view, the approaches and tools developed in this thesis are generalized analysis frameworks, which are applicable across a various range of biological systems with the available multi-omic data. Using the proposed approaches, we assist biologists in transforming their in-house generated omic data into meaningful biological insights through the integrative analysis with molecular interaction networks, which provides valuable clues for understanding the underlying molecular mechanisms in complex biological systems.

7.2 Future work

Future work of multi-omic data integration could be improved or extended in several directions.

Firstly, the qualities of multi-omic data could be improved by more advanced high-throughput experiment technologies. Currently, the multi-omic data suffer high noises because different types of omic data come from separate experiments or different samples which usually bring in high technical and biological variances. Multi-omic data with high qualities will improve the efficiency of machine learning models for the data integration. This limitation could be improved with the development of advanced high-throughput experiment technologies such as the single-cell sequencing technologies [207] which measure a single cell at multiple molecular levels simultaneously and thus reduce the technical and biological variances. This direction is out of the scope of this thesis because the development of high-throughput experiment technologies mainly depends on contributions from the field of biotechnology.

Secondly, network-based approaches for multi-omic data integration could be extended by adding more levels of data and improved by employing more advanced machine learning

techniques. For instances, each of the approaches proposed in this thesis could be extended or improved for better performances. The approach proposed in Chapter 3 could be improved by employing more efficient network module detection methods which are capable of identifying modules from weighted network. The approach proposed in Chapter 4 could be extended to include other types of omic data such as proteome data by adding more layers in the multilayer network, and be improved by employing more efficient seed gene selection methods which are applicable to unbalanced data. The network-constrained regression method, eGBL, proposed in Chapter 5 could be improved by developing more efficient penalty functions to gain higher power for feature selection. The SNN models proposed in Chapter 6 can be extended with more superlayers to integrate more types of omic data, and be improved by employing more advanced neural network architectures such as RNN and LSTM (see details in Chapter 2.3.2) which are applicable to data with more complex structures.

Finally, besides the network-based approaches, multi-omic data could be integrated by using other computational models, e.g., mathematical stochastic processes, depends on the purpose of specific research problems. Recently, we perform a successful multi-omic study to characterise signalling factors in human diseases by integrating DNA methylation data and gene expression data using phylogenetic based stochastic processes [208], which suggests the applicability of a wide range of computational models for multi-omic data integration.

Bibliography

[1] Christoph Bock, Matthias Farlik, and Nathan C Sheffield. Multi-omics of single cells: Strategies and applications. *Trends in biotechnology*, 34:605–608, August 2016. ISSN 1879-3096. doi: 10.1016/j.tibtech.2016.04.004.

[2] Iain C Macaulay, Chris P Ponting, and Thierry Voet. Single-cell multiomics: Multiple measurements from single cells. *Trends in genetics : TIG*, 33:155–168, February 2017. ISSN 0168-9525. doi: 10.1016/j.tig.2016.12.003.

[3] Ali Ebrahim, Elizabeth Brunk, Justin Tan, Edward J O'Brien, Donghyuk Kim, Richard Szubin, Joshua A Lerman, Anna Lechner, Anand Sastry, Aarash Bordbar, Adam M Feist, and Bernhard O Palsson. Multi-omic data integration enables discovery of hidden biological regularities. *Nature communications*, 7:13091, October 2016. ISSN 2041-1723. doi: 10.1038/ncomms13091.

[4] Marylyn D Ritchie, Emily R Holzinger, Ruowang Li, Sarah A Pendergrass, and Dokyoon Kim. Methods of integrating data to uncover genotype-phenotype interactions. *Nature reviews. Genetics*, 16:85–97, February 2015. ISSN 1471-0064. doi: 10.1038/nrg3868.

[5] Pedro Larranaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Inaki Inza, José A Lozano, Rubén Armananzas, Guzmán Santafé, Aritz Pérez, et al. Machine learning in bioinformatics. *Briefings in bioinformatics*, pages 86–112, 2006.

[6] Adi L Tarca, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. Machine learning and its applications to biology. *PLoS computational biology*, 3(6): e116, 2007.

[7] Yuichiro Anzai. *Pattern recognition and machine learning*. Elsevier, 2012.

[8] Uri Alon, Naama Barkai, Daniel A Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[9] Aik Choon Tan and David Gilbert. Ensemble machine learning on gene expression data for cancer classification. *Applied bioinformatics*, 2:S75–S83, 2003. ISSN 1175-5636.

[10] Rasool Fakoor, Faisal Ladhak, Azade Nazi, and Manfred Huber. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the International Conference on Machine Learning*, volume 28, 2013.

[11] Ash A Alizadeh, Michael B Eisen, R Eric Davis, Chi Ma, Izidore S Lossos, Andreas Rosenwald, Jennifer C Boldrick, Hajeer Sabet, Truc Tran, Xin Yu, et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503, 2000.

[12] Jacques Lapointe, Chunde Li, Craig P Giacomini, Keyan Salari, Stephanie Huang, Pei Wang, Michelle Ferrari, Tina Hernandez-Boussard, James D Brooks, and Jonathan R Pollack. Genomic profiling reveals alternative genetic pathways of prostate tumorigenesis. *Cancer research*, 67:8504–8510, September 2007. ISSN 0008-5472. doi: 10.1158/0008-5472.CAN-07-0673.

[13] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41, 2001.

[14] Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101, 2004.

[15] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature reviews genetics*, 12(1):56, 2011.

[16] Wei Zhang, Jeremy Chien, Jeongsik Yong, and Rui Kuang. Network-based machine learning and graph theory algorithms for precision oncology. *npj Precision Oncology*, 1(1):25, 2017.

[17] Lei Chen, Tao Huang, Chuan Lu, Lin Lu, and Dandan Li. Machine learning and network methods for biology and medicine. *Computational and Mathematical Methods in Medicine*, 2015, 2015.

[18] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561, 1970.

[19] Tim Hubbard, Daniel Barker, Ewan Birney, Graham Cameron, Yuan Chen, L Clark, Tony Cox, J Cuff, Val Curwen, Thomas Down, et al. The ensembl genome database project. *Nucleic acids research*, 30(1):38–41, 2002.

[20] Bradley E Bernstein, Alexander Meissner, and Eric S Lander. The mammalian epigenome. *Cell*, 128(4):669–681, 2007.

[21] Andrew B Conley and I King Jordan. Endogenous retroviruses and the epigenome. In *Viruses: Essential Agents of Life*, pages 309–323. Springer, 2012.

[22] Rudolf Jaenisch and Adrian Bird. Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals. *Nature genetics*, 33:245, 2003.

[23] Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics*, 10:57–63, January 2009. ISSN 1471-0064. doi: 10.1038/nrg2484.

[24] Helen A King and Andr P Gerber. Translatome profiling: methods for genome-scale analysis of mrna translation. *Briefings in functional genomics*, 15:22–31, January 2016. ISSN 2041-2657. doi: 10.1093/bfgp/elu045.

[25] Gloria A Brar and Jonathan S Weissman. Ribosome profiling reveals the what, when, where and how of protein synthesis. *Nature reviews Molecular cell biology*, 16 (11):651, 2015.

[26] Abdelali Haoudi and Halima Bensmail. Bioinformatics and data mining in proteomics. *Expert review of proteomics*, 3:333–343, June 2006. ISSN 1744-8387. doi: 10.1586/14789450.3.3.333.

[27] Yan V Sun and Yi-Juan Hu. Integrative analysis of multi-omics data for discovery and functional studies of complex human diseases. *Advances in genetics*, 93:147–190, 2016. ISSN 0065-2660. doi: 10.1016/bs.adgen.2015.11.004.

[28] Michael Caldera, Pisanu Buphamalai, Felix Müller, and Jörg Menche. Interactome-based approaches to human disease. *Current Opinion in Systems Biology*, 3:88–94, 2017.

[29] Barabasi and Albert. Emergence of scaling in random networks. *Science (New York, N.Y.)*, 286:509–512, October 1999. ISSN 1095-9203.

[30] Jacques Demongeot and Lloyd A Demetrius. Complexity and stability in biological systems. *International Journal of Bifurcation and Chaos*, 25(07):1540013, 2015.

[31] Cohen, Erez, ben Avraham, and Havlin. Resilience of the internet to random breakdowns. *Physical review letters*, 85:4626–4628, November 2000. ISSN 1079-7114. doi: 10.1103/PhysRevLett.85.4626.

[32] L H Hartwell, J J Hopfield, S Leibler, and A W Murray. From molecular to modular cell biology. *Nature*, 402:C47–C52, December 1999. ISSN 0028-0836. doi: 10.1038/35011540.

[33] Sabine Tornow and H W Mewes. Functional modules by relating protein interaction networks and gene expression. *Nucleic acids research*, 31:6283–6289, November 2003. ISSN 1362-4962.

[34] Jing-Dong J Han, Nicolas Bertin, Tong Hao, Debra S Goldberg, Gabriel F Berriz, Lan V Zhang, Denis Dupuy, Albertha J M Walhout, Michael E Cusick, Frederick P Roth, and Marc Vidal. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430:88–93, July 2004. ISSN 1476-4687. doi: 10.1038/nature02555.

[35] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[36] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[37] Guilherme Ferraz de Arruda, Emanuele Cozzo, Tiago P Peixoto, Francisco A Rodrigues, and Yamir Moreno. Disease localization in multilayer networks. *Physical Review X*, 7(1):011014, 2017.

[38] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multilayer tissue networks. *Bioinformatics (Oxford, England)*, 33:i190–i198, July 2017. ISSN 1367-4811. doi: 10.1093/bioinformatics/btx252.

[39] Xiaoke Ma, Zaiyi Liu, Zhongyuan Zhang, Xiaotai Huang, and Wanxin Tang. Multiple network algorithm for epigenetic modules via the integration of genome-wide dna methylation and gene expression data. *BMC bioinformatics*, 18(1):72, 2017.

[40] Laura Cantini, Enzo Medico, Santo Fortunato, and Michele Caselle. Detection of gene communities in multi-networks reveals cancer drivers. *Scientific reports*, 5: 17386, December 2015. ISSN 2045-2322. doi: 10.1038/srep17386.

[41] Clara Dismuke and Richard Lindrooth. Ordinary least squares. *Methods and Designs for Outcomes Research*, 93:93–104, 2006.

[42] Fumio Hayashi. Econometrics. 2000. *Princeton University Press. Section*, 1:60–69, 2000.

[43] Scott Fortmann-Roe. Understanding the bias-variance tradeoff. 2012.

[44] Arthur E Hoerl and Robert W Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, 2000.

[45] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[46] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2): 301–320, 2005.

[47] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[48] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[49] Patrick Breheny. The group exponential lasso for bi-level variable selection. *Biometrics*, 71(3):731–740, 2015.

[50] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456): 1348–1360, 2001.

[51] Caiyan Li and Hongzhe Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics (Oxford, England)*, 24:1175–1182, May 2008. ISSN 1367-4811. doi: 10.1093/bioinformatics/btn081.

[52] Wei Pan, Benhuai Xie, and Xiaotong Shen. Incorporating predictor network in penalized regression with application to microarray data. *Biometrics*, 66:474–484, June 2010. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2009.01296.x.

[53] Peng Zhao and Bin Yu. Boosted lasso. Technical report, CALIFORNIA UNIV BERKELEY DEPT OF STATISTICS, 2004.

[54] BWAC Farley and W Clark. Simulation of self-organizing systems by digital computer. *Transactions of the IRE Professional Group on Information Theory*, 4(4):76–84, 1954.

[55] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[56] R Hecht-Nielsen. Neural network primer: part i. *AI Expert*, pages 4–51, 1989.

[57] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

[58] Dan C Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1237. Barcelona, Spain, 2011.

[59] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5): 855–868, 2009.

[60] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[61] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[62] Stephan R Sain. The nature of statistical learning theory, 1996.

[63] Tin Kam Ho. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE, 1995.

[64] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[65] L Rocach and O Maimon. Clustering methods data mining and knowledge discovery handbook. *Springer US*, page 321, 2005.

[66] Shailesh Tripathi, Salissou Moutari, Matthias Dehmer, and Frank Emmert-Streib. Comparison of module detection algorithms in protein networks and investigation of the biological meaning of predicted modules. *BMC bioinformatics*, 17:129, March 2016. ISSN 1471-2105. doi: 10.1186/s12859-016-0979-8.

[67] Clara Pizzuti and Simona E Rombo. Algorithms and tools for protein–protein interaction networks clustering, with a special focus on population-based stochastic methods. *Bioinformatics*, 30(10):1343–1352, 2014.

[68] Gary D Bader and Christopher WV Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC bioinformatics*, 4 (1):2, 2003.

[69] László Lovász et al. Random walks on graphs: A survey. *Combinatorics, Paul erdos is eighty*, 2(1):1–46, 1993.

[70] Anton J Enright, Stijn Van Dongen, and Christos A Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584, 2002.

[71] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *nature*, 466(7307):761, 2010.

[72] István A Kovács, Robin Palotai, Máté S Szalay, and Peter Csermely. Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PloS one*, 5(9):e12528, 2010.

[73] Emmanuelle Becker, Benot Robisson, Charles E Chapple, Alain Gunoche, and Christine Brun. Multifunctional proteins revealed by overlapping clustering in protein interaction network. *Bioinformatics (Oxford, England)*, 28:84–90, January 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/btr621.

[74] Gary A Churchill. Fundamentals of experimental design for cdna microarrays. *Nature genetics*, 32 Suppl:490–495, December 2002. ISSN 1061-4036. doi: 10.1038/ng1031.

[75] Dominic J Allocco, Isaac S Kohane, and Atul J Butte. Quantifying the relationship between co-expression, co-regulation and gene function. *BMC bioinformatics*, 5:18, February 2004. ISSN 1471-2105. doi: 10.1186/1471-2105-5-18.

[76] Eran Segal, Michael Shapira, Aviv Regev, Dana Pe'er, David Botstein, Daphne Koller, and Nir Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature genetics*, 34: 166–176, June 2003. ISSN 1061-4036. doi: 10.1038/ng1165.

[77] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4: Article17, 2005. ISSN 1544-6115. doi: 10.2202/1544-6115.1128.

[78] Qi Liao, Changning Liu, Xiongying Yuan, Shuli Kang, Ruoyu Miao, Hui Xiao, Guoguang Zhao, Haitao Luo, Dechao Bu, Haitao Zhao, et al. Large-scale prediction of long non-coding rna functions in a coding–non-coding gene co-expression network. *Nucleic acids research*, 39(9):3864–3878, 2011.

[79] Sebastian Vlaic, Theresia Conrad, Christian Tokarski-Schnelle, Mika Gustafsson, Uta Dahmen, Reinhard Guthke, and Stefan Schuster. Modulediscoverer: Identification

of regulatory modules in protein-protein interaction networks. *Scientific reports*, 8: 433, January 2018. ISSN 2045-2322. doi: 10.1038/s41598-017-18370-2.

[80] Sipko van Dam, Urmo Vsa, Adriaan van der Graaf, Lude Franke, and Joo Pedro de Magalhes. Gene co-expression analysis for functional classification and gene-disease predictions. *Briefings in bioinformatics*, January 2017. ISSN 1477-4054. doi: 10.1093/bib/bbw139.

[81] Ioannis A Maraziotis, Konstantina Dimitrakopoulou, and Anastasios Bezerianos. Growing functional modules from a seed protein via integration of protein interaction and gene expression data. *BMC bioinformatics*, 8:408, October 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-408.

[82] Igor Ulitsky and Ron Shamir. Identifying functional modules using expression profiles and confidence-scored protein interactions. *Bioinformatics (Oxford, England)*, 25: 1158–1164, May 2009. ISSN 1367-4811. doi: 10.1093/bioinformatics/btp118.

[83] Zheng Guo, Lei Wang, Yongjin Li, Xue Gong, Chen Yao, Wencai Ma, Dong Wang, Yanhui Li, Jing Zhu, Min Zhang, Da Yang, Shaoqi Rao, and Jing Wang. Edge-based scoring and searching method for identifying condition-responsive protein-protein interaction sub-network. *Bioinformatics (Oxford, England)*, 23:2121–2128, August 2007. ISSN 1367-4811. doi: 10.1093/bioinformatics/btm294.

[84] Le Ou-Yang, Dao-Qing Dai, Xiao-Li Li, Min Wu, Xiao-Fei Zhang, and Peng Yang. Detecting temporal protein complexes from dynamic protein-protein interaction networks. *BMC bioinformatics*, 15:335, October 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-335.

[85] Kathy K Niakan, Jinnuo Han, Roger A Pedersen, Carlos Simon, and Renee A Reijo Pera. Human pre-implantation embryo development. *Development (Cambridge, England)*, 139:829–841, March 2012. ISSN 1477-9129. doi: 10.1242/dev.060426.

[86] Rebecca J Chason, John Csokmay, James H Segars, Alan H DeCherney, and D Randall Armant. Environmental and epigenetic effects upon preimplantation embryo metabolism and development. *Trends in endocrinology and metabolism: TEM*, 22: 412–420, October 2011. ISSN 1879-3061. doi: 10.1016/j.tem.2011.05.005.

[87] Rafael A Irizarry, Bridget Hobbs, Francois Collin, Yasmin D Beazer-Barclay, Kristen J Antonellis, Uwe Scherf, and Terence P Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics (Oxford, England)*, 4:249–264, April 2003. ISSN 1465-4644. doi: 10.1093/biostatistics/4.2.249.

[88] Stuart D Pepper, Emma K Saunders, Laura E Edwards, Claire L Wilson, and Crispin J Miller. The utility of mas5 expression summary and detection call algorithms. *BMC bioinformatics*, 8:273, July 2007. ISSN 1471-2105. doi: 10.1186/1471-2105-8-273.

[89] Laurent Gautier, Leslie Cope, Benjamin M Bolstad, and Rafael A Irizarry. affyanalysis of affymetrix genechip data at the probe level. *Bioinformatics*, 20(3):307–315, 2004.

[90] Ethan G Cerami, Benjamin E Gross, Emek Demir, Igor Rodchenkov, Ozgn Babur, Nadia Anwar, Nikolaus Schultz, Gary D Bader, and Chris Sander. Pathway commons, a web resource for biological pathway data. *Nucleic acids research*, 39:D685–D690, January 2011. ISSN 1362-4962. doi: 10.1093/nar/gkq1039.

[91] Patrcia P Silveira, Andr K Portella, Marcelo Z Goldani, and Marco A Barbieri. Developmental origins of health and disease (dohad). *Jornal de pediatria*, 83:494–504, 2007. ISSN 0021-7557. doi: doi:10.2223/JPED.1728.

[92] Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559, 2008.

[93] Michael Watson. Coxpress: differential co-expression in gene expression data. *BMC bioinformatics*, 7(1):509, 2006.

[94] Bruno M Tesson, Rainer Breitling, and Ritsert C Jansen. Diffcoex: a simple and sensitive method to find differentially coexpressed gene modules. *BMC bioinformatics*, 11(1):497, 2010.

[95] Min Jin Ha, Veerabhadran Baladandayuthapani, and Kim-Anh Do. Dingo: differential network analysis in genomics. *Bioinformatics*, 31(21):3413–3420, 2015.

[96] Mahdi Jalili, Tom Gebhardt, Olaf Wolkenhauer, and Ali Salehzadeh-Yazdi. Unveiling network-based functional features through integration of gene expression into protein networks. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 2018.

[97] Haisu Ma, Eric E Schadt, Lee M Kaplan, and Hongyu Zhao. Cosine: Condition-specific sub-network identification using a global optimization method. *Bioinformatics*, 27(9):1290–1298, 2011.

[98] Li Chen, Jianhua Xuan, Rebecca B Riggins, Yue Wang, and Robert Clarke. Identifying protein interaction subnetworks by a bagging markov random field-based method. *Nucleic acids research*, 41(2):e42–e42, 2012.

[99] Xianjun Shen, Li Yi, Xingpeng Jiang, Tingting He, Xiaohua Hu, and Jincai Yang. Mining temporal protein complex based on the dynamic pin weighted with connected affinity and gene co-expression. *PloS one*, 11(4):e0153967, 2016.

[100] Nazar Zaki and Antonio Mora. A comparative analysis of computational approaches and algorithms for protein subcomplex identification. *Scientific reports*, 4:4262, 2014.

[101] Wouter Saelens, Robrecht Cannoodt, and Yvan Saeys. A comprehensive evaluation of module detection methods for gene expression data. *Nature communications*, 9 (1):1090, 2018.

[102] M E J Newman. Fast algorithm for detecting community structure in networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69:066133, June 2004. ISSN 1539-3755. doi: 10.1103/PhysRevE.69.066133.

[103] Jean-Baptiste Angelelli and Laurence Reboul. Network modularity optimization by a fusionfission process and application to protein-protein interactions networks. *Proceedings of JOBIM 2008*, pages 105–110, 2008.

[104] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102:15545–15550, October 2005. ISSN 0027-8424. doi: 10.1073/pnas.0506580102.

[105] Gene Ontology Consortium. The gene ontology project in 2008. *Nucleic acids research*, 36(suppl_1):D440–D444, 2007.

[106] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[107] Samson O Obado and Michael P Rout. Cilia and nuclear pore proteins: Pore no more? *Developmental cell*, 38:445–446, September 2016. ISSN 1878-1551. doi: 10.1016/j.devcel.2016.08.019.

[108] Javier Fernandez-Martinez, Seung Joong Kim, Yi Shi, Paula Upla, Riccardo Pellarin, Michael Gagnon, Ilan E Chemmama, Junjie Wang, Ilona Nudelman, Wenzhu Zhang, Rosemary Williams, William J Rice, David L Stokes, Daniel Zenklusen, Brian T Chait, Andrej Sali, and Michael P Rout. Structure and function of the nuclear pore complex cytoplasmic mrna export platform. *Cell*, 167:1215–1228.e25, November 2016. ISSN 1097-4172. doi: 10.1016/j.cell.2016.10.028.

[109] M Smitherman, K Lee, J Swanger, R Kapur, and B E Clurman. Characterization and targeted disruption of murine nup50, a p27(kip1)-interacting component of the nuclear pore complex. *Molecular and cellular biology*, 20:5631–5642, August 2000. ISSN 0270-7306.

[110] Maya Capelson, Yun Liang, Roberta Schulte, William Mair, Ulrich Wagner, and Martin W Hetzer. Chromatin-bound nuclear pore components regulate gene expression in higher eukaryotes. *Cell*, 140:372–383, February 2010. ISSN 1097-4172. doi: 10.1016/j.cell.2009.12.054.

[111] Bernike Kalverda, Helen Pickersgill, Victor V Shloma, and Maarten Fornerod. Nucleoporins directly stimulate expression of developmental and cell-cycle genes inside the nucleoplasm. *Cell*, 140:360–371, February 2010. ISSN 1097-4172. doi: 10.1016/j.cell.2010.01.011.

[112] Filipe V Jacinto, Chris Benner, and Martin W Hetzer. The nucleoporin nup153 regulates embryonic stem cell pluripotency through gene silencing. *Genes & development*, 29:1224–1238, June 2015. ISSN 1549-5477. doi: 10.1101/gad.260919.115.

[113] Connie C Wong, Kevin E Loewke, Nancy L Bossert, Barry Behr, Christopher J De Jonge, Thomas M Baer, and Renee A Reijo Pera. Non-invasive imaging of human embryos before embryonic genome activation predicts development to the blastocyst stage. *Nature biotechnology*, 28:1115–1121, October 2010. ISSN 1546-1696. doi: 10.1038/nbt.1686.

[114] Eduardo Rdenas, Elke P F Klerkx, Cristina Ayuso, Anjon Audhya, and Peter Askjaer. Early embryonic requirement for nucleoporin nup35/npp-19 in nuclear assembly. *Developmental biology*, 327:399–409, March 2009. ISSN 1095-564X. doi: 10.1016/j.ydbio.2008.12.024.

[115] Cerstin Franz, Peter Askjaer, Wolfram Antonin, Carmen Lpez Iglesias, Uta Haselmann, Malgorzata Schelder, Ario de Marco, Matthias Wilm, Claude Antony, and Iain W Mattaj. Nup155 regulates nuclear envelope and nuclear pore complex formation in nematodes and vertebrates. *The EMBO journal*, 24:3519–3531, October 2005. ISSN 0261-4189. doi: 10.1038/sj.emboj.7600825.

[116] Amparo Galan, David Montaner, M Eugenia Poo, Diana Valbuena, Verónica Ruiz, Cristóbal Aguilar, Joaquín Dopazo, and Carlos Simón. Functional genomics of 5-to 8-cell stage human embryos by blastomere single-cell cdna analysis. *PloS one*, 5(10): e13615, 2010.

[117] Amy Ralston, Brian J Cox, Noriyuki Nishioka, Hiroshi Sasaki, Evelyn Chea, Peter Rugg-Gunn, Guoji Guo, Paul Robson, Jonathan S Draper, and Janet Rossant. Gata3 regulates trophoblast development downstream of tead4 and in parallel to cdx2. *Development (Cambridge, England)*, 137:395–403, February 2010. ISSN 1477-9129. doi: 10.1242/dev.038828.

[118] Stijn Marinus Van Dongen. *Graph clustering by flow simulation*. PhD thesis, 2001.

[119] Mt Szalay-Beko, Robin Palotai, Balzs Szappanos, Istvn A Kovcs, Balzs Papp, and Pter Csermely. Moduland plug-in for cytoscape: determination of hierarchical layers of overlapping network modules and community centrality. *Bioinformatics (Oxford, England)*, 28:2202–2204, August 2012. ISSN 1367-4811. doi: 10.1093/bioinformatics/bts352.

[120] Liying Yan, Mingyu Yang, Hongshan Guo, Lu Yang, Jun Wu, Rong Li, Ping Liu, Ying Lian, Xiaoying Zheng, Jie Yan, Jin Huang, Ming Li, Xinglong Wu, Lu Wen, Kaiqin Lao, Ruiqiang Li, Jie Qiao, and Fuchou Tang. Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature structural & molecular biology*, 20:1131–1139, September 2013. ISSN 1545-9985. doi: 10.1038/nsmb.2660.

[121] Peng Lu, Christine Vogel, Rong Wang, Xin Yao, and Edward M Marcotte. Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. *Nature biotechnology*, 25:117–124, January 2007. ISSN 1087-0156. doi: 10.1038/nbt1270.

[122] John W B Hershey, Nahum Sonenberg, and Michael B Mathews. Principles of translational control: an overview. *Cold Spring Harbor perspectives in biology*, 4, December 2012. ISSN 1943-0264. doi: 10.1101/cshperspect.a011528.

[123] Mamatha Bhat, Nathaniel Robichaud, Laura Hulea, Nahum Sonenberg, Jerry Pelletier, and Ivan Topisirovic. Targeting the translation machinery in cancer. *Nature reviews. Drug discovery*, 14:261–278, April 2015. ISSN 1474-1784. doi: 10.1038/nrd4505.

[124] Michal Grzmil and Brian A Hemmings. Translation regulation as a therapeutic target in cancer. *Cancer research*, 72:3891–3900, August 2012. ISSN 1538-7445. doi: 10.1158/0008-5472.CAN-12-0026.

[125] Nicholas T Ingolia, Sina Ghaemmaghami, John R S Newman, and Jonathan S Weissman. Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *Science*, 324:218–223, April 2009. ISSN 1095-9203. doi: 10.1126/science.1168978.

[126] Nicholas T Ingolia. Ribosome footprint profiling of translation throughout the genome. *Cell*, 165:22–33, March 2016. ISSN 1097-4172. doi: 10.1016/j.cell.2016.02.066.

[127] Adam B Olshen, Andrew C Hsieh, Craig R Stumpf, Richard A Olshen, Davide Ruggero, and Barry S Taylor. Assessing gene-level translational control from ribosome profiling. *Bioinformatics*, 29(23):2995–3002, 2013.

[128] Zhengtao Xiao, Qin Zou, Yu Liu, and Xuerui Yang. Genome-wide assessment of differential translations with ribosome profiling data. *Nature communications*, 7: 11194, 2016.

[129] Wenzheng Li, Weili Wang, Philip J Uren, Luiz OF Penalva, and Andrew D Smith. Riborex: fast and flexible identification of differential translation from ribo-seq data. *Bioinformatics*, 33(11):1735–1737, 2017.

[130] Xiaoke Ma, Long Gao, and Kai Tan. Modeling disease progression using dynamics of pathway connectivity. *Bioinformatics*, 30(16):2343–2350, 2014.

[131] Andrew C Hsieh, Yi Liu, Merritt P Edlind, Nicholas T Ingolia, Matthew R Janes, Annie Sher, Evan Y Shi, Craig R Stumpf, Carly Christensen, Michael J Bonham, Shunyou Wang, Pingda Ren, Michael Martin, Katti Jessen, Morris E Feldman, Jonathan S Weissman, Kevan M Shokat, Christian Rommel, and Davide Ruggero. The translational landscape of mtor signalling steers cancer initiation and metastasis. *Nature*, 485:55–61, February 2012. ISSN 1476-4687. doi: 10.1038/nature10912.

[132] Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P Tsafou, Michael Kuhn, Peer Bork, Lars J Jensen, and Christian von Mering. String v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic acids research*, 43:D447–D452, January 2015. ISSN 1362-4962. doi: 10.1093/nar/gku1003.

[133] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15:550, 2014. ISSN 1474-760X. doi: 10.1186/s13059-014-0550-8.

[134] Rita Spilka, Nicole Golob-Schwarzl, and Stefanie Krassnig. *Translational Control and mTOR in Cancer*. 05 2017. doi: 10.1007/978-3-319-53659-0_5.

[135] Alfredo Csibi, Karen Cornille, Marie-Pierre Leibovitch, Anne Poupon, Lionel A Tintignac, Anthony MJ Sanchez, and Serge A Leibovitch. The translation regulatory subunit eif3f controls the kinase-dependent mtor signaling required for muscle differentiation and hypertrophy in mouse. *PLoS One*, 5(2):e8994, 2010.

[136] Roberta Marchione, Serge A Leibovitch, and Jean-Luc Lenormand. The translational factor eif3f: the ambivalent eif3 subunit. *Cellular and Molecular Life Sciences*, 70 (19):3603–3616, 2013.

[137] Columba de la Parra, Beth A Walters, Phillip Geter, and Robert J Schneider. Translation initiation factors and their relevance in cancer. *Current opinion in genetics & development*, 48:82–88, 2018.

[138] Deborah Silvera, Silvia C Formenti, and Robert J Schneider. Translational control in cancer. *Nature Reviews Cancer*, 10(4):254, 2010.

[139] Meilin Wang, Atsushi Takahashi, Fang Liu, Dingwei Ye, Qiang Ding, Chao Qin, Changjun Yin, Zhengdong Zhang, Koichi Matsuda, Michiaki Kubo, et al. Large-scale association analysis in asians identifies new susceptibility loci for prostate cancer. *Nature communications*, 6:8469, 2015.

[140] Prasanna Parasuraman, Peter Mulligan, James A Walker, Bihua Li, Myriam Boukhali, Wilhelm Haas, and Andre Bernards. Interaction of p190a rhogap with eif3a and other translation preinitiation factors suggests a role in protein biosynthesis. *Journal of Biological Chemistry*, 292(7):2679–2689, 2017.

[141] Giovanni Lavorgna, Fulvio Chiacchiera, Alberto Briganti, Francesco Montorsi, Diego Pasini, and Andrea Salonia. Expression-profiling of apoptosis induced by ablation of the long ncrna trpm2-as in prostate cancer cell. *Genomics data*, 3:4–5, 2015.

[142] David Shahbazian, Philippe P Roux, Virginie Mieulet, Michael S Cohen, Brian Raught, Jack Taunton, John WB Hershey, John Blenis, Mario Pende, and Nahum Sonenberg. The mtor/pi3k and mapk pathways converge on eif4b to control its phosphorylation and activity. *The EMBO journal*, 25(12):2781–2791, 2006.

[143] Ke Ren, Xin Gou, Mingzhao Xiao, Ming Wang, Chaodong Liu, Zhaobing Tang, and Weiyang He. The over-expression of pim-2 promote the tumorigenesis of prostatic carcinoma through phosphorylating eif4b. *The Prostate*, 73(13):1462–1469, 2013.

[144] Ji-Ye Yin, Jian-Ting Zhang, Wei Zhang, Hong-Hao Zhou, and Zhao-Qian Liu. eif3a: A new anticancer drug target in the eif family. *Cancer letters*, 412:81–87, 2018.

[145] Masaki Mishima, Shigeo Wakabayashi, and Chojiro Kojima. Solution structure of the cytoplasmic region of na+/h+ exchanger 1 complexed with essential cofactor calcineurin b homologous protein 1. *Journal of Biological Chemistry*, 282(4):2741–2751, 2007.

[146] David W Good, Thampi George, and Bruns A Watts. Nerve growth factor inhibits na+/h+ exchange and absorption through parallel phosphatidylinositol 3-kinase-mtor and erk pathways in thick ascending limb. *Journal of Biological Chemistry*, 283(39):26602–26611, 2008.

[147] Soumya Chatterjee, Sebastian Schmidt, Stella Pouli, Sabina Honisch, Saad Alkahtani, Christos Stournaras, and Florian Lang. Membrane androgen receptor sensitive na+/h+ exchanger activity in prostate cancer cells. *FEBS letters*, 588(9):1571–1579, 2014.

[148] Padmanaban S Suresh, Rie Tsutsumi, and Thejaswini Venkatesh. Ybx1 at the crossroads of non-coding transcriptome, exosomal, and cytoplasmic granular signaling. *European journal of cell biology*, 2018.

[149] Kenjiro Imada, Masaki Shiota, Kenichi Kohashi, Kentaro Kuroiwa, YooHyun Song, Masaaki Sugimoto, Seiji Naito, and Yoshinao Oda. Mutual regulation between raf/mek/erk signaling and y-box-binding protein-1 promotes prostate cancer progression. *Clinical Cancer Research*, pages clincanres–3705, 2013.

[150] David J Pisapia, Steven Salvatore, Chantal Pauli, Erika Hissong, Ken Eng, Davide Prandi, Verena-Wilbeth Sailer, Brian D Robinson, Kyung Park, Joanna Cyrta, et al. Next-generation rapid autopsies enable tumor evolution tracking and generation of preclinical models. *JCO precision oncology*, 1:1–13, 2017.

[151] Henrique B da Silva, Eduardo P Amaral, Eduardo L Nolasco, Nathalia C de Victo, Rodrigo Atique, Carina C Jank, Valesca Anschau, Luiz F Zerbini, and Ricardo G Correa. Dissecting major signaling pathways throughout the development of prostate cancer. *Prostate cancer*, 2013, 2013.

[152] Meiling Gao, Rachana Patel, Imran Ahmad, Janis Fleming, Joanne Edwards, Stuart McCracken, Kanagasabai Sahadevan, Morag Seywright, Jim Norman, Owen Sansom, et al. Spry2 loss enhances erbb trafficking and pi3k/akt signalling to drive human and mouse prostate carcinogenesis. *EMBO molecular medicine*, 4(8):776–790, 2012.

[153] Lizhou Jia, Tingting Yang, Xuan Gu, Wei Zhao, Qi Tang, Xudong Wang, Jin Zhu, and Zhenqing Feng. Translation elongation factor eef1bα is identified as a novel prognostic marker of gastric cancer. *International Journal of Biological Macromolecules*, 2018.

[154] Md Khurshidul Hassan, Dinesh Kumar, Monali Naik, and Manjusha Dixit. The expression profile and prognostic significance of eukaryotic translation elongation factors in different cancers. *PloS one*, 13(1):e0191377, 2018.

[155] James M Dolezal, Arie P Dash, and Edward V Prochownik. Diagnostic and prognostic implications of ribosomal protein transcript expression patterns in human cancers. *BMC cancer*, 18(1):275, 2018.

[156] Weicheng Wu, Xixi Zheng, Jing Wang, Tianxiao Yang, Wenjuan Dai, Shushu Song, Lan Fang, Yilin Wang, and Jianxin Gu. O-glcnacylation on rab3a attenuates its effects on mitochondrial oxidative phosphorylation and metastasis in hepatocellular carcinoma. *Cell death & disease*, 9(10):970, 2018.

[157] A C Gingras, S G Kennedy, M A O'Leary, N Sonenberg, and N Hay. 4e-bp1, a repressor of mrna translation, is phosphorylated and inactivated by the akt(pkb) signaling pathway. *Genes & development*, 12:502–513, February 1998. ISSN 0890-9369.

[158] Andrew C Hsieh, Maria Costa, Ornella Zollo, Cole Davis, Morris E Feldman, Joseph R Testa, Oded Meyuhas, Kevan M Shokat, and Davide Ruggero. Genetic dissection of the oncogenic mtor pathway reveals druggable addiction to translational control via 4ebp-eif4e. *Cancer cell*, 17:249–261, March 2010. ISSN 1878-3686. doi: 10.1016/j.ccr.2010.01.021.

[159] Y Toh, S Kuninaka, K Endo, T Oshiro, Y Ikeda, H Nakashima, H Baba, S Kohnoe, T Okamura, G L Nicolson, and K Sugimachi. Molecular analysis of a candidate metastasis-associated gene, mta1: possible interaction with histone deacetylase 1. *Journal of experimental & clinical cancer research : CR*, 19:105–111, March 2000. ISSN 0392-9078.

[160] Lakshmi Prabhu, Antja-Voy Hartley, Matthew Martin, Fadumo Warsame, Emily Sun, and Tao Lu. Role of post-translational modification of the y box binding protein 1 in human cancers. *Genes & Diseases*, 2(3):240–246, 2015.

[161] M Mura, T G Hopkins, T Michael, N Abd-Latip, J Weir, E Aboagye, F Mauri, C Jameson, J Sturge, H Gabra, M Bushell, A E Willis, E Curry, and S P Blagden. Larp1 post-transcriptionally regulates mtor and contributes to cancer progression. *Oncogene*, 34:5025–5036, September 2015. ISSN 1476-5594. doi: 10.1038/onc.2014.428.

[162] Cherie Blenkiron, Daniel G Hurley, Sandra Fitzgerald, Cristin G Print, and Annette Lasham. Links between the oncoprotein yb-1 and small non-coding rnas in breast cancer. *PloS one*, 8:e80171, 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0080171.

[163] T Freour, J Lammers, C Splingart, M Jean, and P Barriere. [time lapse (embryoscope) as a routine technique in the ivf laboratory: a useful tool for better embryo selection?]. *Gynecologie, obstetrique & fertilite*, 40:476–480, September 2012. ISSN 1769-6682. doi: 10.1016/j.gyobfe.2012.07.008.

[164] Marcos Meseguer, Javier Herrero, Alberto Tejera, Karen Marie Hilligse, Niels Birger Ramsing, and Jose Remoh. The use of morphokinetics as a predictor of embryo implantation. *Human reproduction (Oxford, England)*, 26:2658–2671, October 2011. ISSN 1460-2350. doi: 10.1093/humrep/der256.

[165] N Basile, P Vime, M Florensa, B Aparicio Ruiz, J A Garca Velasco, J Remoh, and M Meseguer. The use of morphokinetics as a predictor of implantation: a multicentric study to define and validate an algorithm for embryo selection. *Human reproduction (Oxford, England)*, 30:276–283, February 2015. ISSN 1460-2350. doi: 10.1093/humrep/deu331.

[166] Amarjot Singh. *Automatic Methods for Human Embryo Component Extraction*. PhD thesis, Applied Sciences: School of Engineering Science, 2014.

[167] R Sciorio, J K Thong, and S J Pickering. Comparison of the development of human embryos cultured in either an embryoscope or benchtop incubator. *Journal of assisted reproduction and genetics*, December 2017. ISSN 1573-7330. doi: 10.1007/s10815-017-1100-6.

[168] M Rallu, M Loones, Y Lallemand, R Morimoto, M Morange, and V Mezger. Function and regulation of heat shock factor 2 during mouse embryogenesis. *Proceedings of the National Academy of Sciences of the United States of America*, 94:2392–2397, March 1997. ISSN 0027-8424.

[169] Jeffrey M Cloutier, Shantha K Mahadevaiah, Elias ElInati, Andr Nussenzweig, Attila Tth, and James M A Turner. Histone h2afx links meiotic chromosome asynapsis to prophase i oocyte loss in mammals. *PLoS genetics*, 11:e1005462, October 2015. ISSN 1553-7404. doi: 10.1371/journal.pgen.1005462.

[170] L Zhu, Z Jiang, J Duan, H Dong, X Zheng, LA Blomberg, DM Donovan, N Talbot, J Chen, and X Tian. 127 abundance of mrna for histone variants, histone, and dna modification enzymes in bovine in vivo oocytes and pre-implantation embryos. *Reproduction, Fertility and Development*, 29(1):172–172, 2017.

[171] Silvia Bione, Cinzia Sala, Chiara Manzini, Giulia Arrigo, Orsetta Zuffardi, Sandro Banfi, Giuseppe Borsani, Philippe Jonveaux, Christophe Philippe, Maurizio Zuccotti, et al. A human homologue of the drosophila melanogaster diaphanous gene is

disrupted in a patient with premature ovarian failure: evidence for conserved function in oogenesis and implications for human sterility. *The American Journal of Human Genetics*, 62(3):533–541, 1998.

[172] Mary Ann Suico, Tsuyoshi Shuto, and Hirofumi Kai. Roles and regulations of the ets transcription factor elf4/mef. *Journal of molecular cell biology*, 9:168–177, June 2017. ISSN 1759-4685. doi: 10.1093/jmcb/mjw051.

[173] Georgia Kakourou, Souraya Jaroudi, Pinar Tulay, Carleen Heath, Paul Serhal, Joyce C Harper, and Sioban B Sengupta. Investigation of gene expression profiles before and after embryonic genome activation and assessment offunctional pathways at the humanmetaphase ii oocyte andblastocyst stage. *Fertility and sterility*, 99: 803–814.e23, March 2013. ISSN 1556-5653. doi: 10.1016/j.fertnstert.2012.10.036.

[174] Myriam Hemberger, Tadashige Nozaki, Elke Winterhager, Hideyuki Yamamoto, Hitoshi Nakagama, Nobuo Kamada, Hiroshi Suzuki, Tsutomu Ohta, Misao Ohki, Mitsuko Masutani, and James C Cross. Parp1-deficiency induces differentiation of es cells into trophoblast derivatives. *Developmental biology*, 257:371–381, May 2003. ISSN 0012-1606.

[175] Dan Strumpf, Chai-An Mao, Yojiro Yamanaka, Amy Ralston, Kallayanee Chawengsaksophak, Felix Beck, and Janet Rossant. Cdx2 is required for correct cell fate specification and differentiation of trophectoderm in the mouse blastocyst. *Development (Cambridge, England)*, 132:2093–2102, May 2005. ISSN 0950-1991. doi: 10.1242/dev.01801.

[176] S Koerber, A N Santos, F Tetens, A Kchenhoff, and B Fischer. Increased expression of nadh-ubiquinone oxidoreductase chain 2 (nd2) in preimplantation rabbit embryos cultured with 20 *Molecular reproduction and development*, 49:394–399, April 1998. ISSN 1040-452X. doi: 10.1002/(SICI)1098-2795(199804)49:4⟨394::AID-MRD6⟩3.0.CO;2-I.

[177] M P Alcolea, B Colom, I Llad, F J Garca-Palmer, and M Gianotti. Mitochondrial differentiation and oxidative phosphorylation system capacity in rat embryo during placentation period. *Reproduction (Cambridge, England)*, 134:147–154, July 2007. ISSN 1470-1626. doi: 10.1530/REP-07-0012.

[178] Yaacov Barak, Yoel Sadovsky, and Tali Shalom-Barak. Ppar signaling in placental development and function. *PPAR research*, 2008:142082, 2008. ISSN 1687-4757. doi: 10.1155/2008/142082.

[179] Katharina Walentin, Christian Hinze, Max Werth, Nadine Haase, Saaket Varma, Robert Morell, Annekatrin Aue, Elisabeth Ptschke, David Warburton, Andong Qiu, Jonathan Barasch, Bettina Purfrst, Christoph Dieterich, Elena Popova, Michael Bader, Ralf Dechend, Anne Cathrine Staff, Zeliha Yesim Yurtdas, Ergin Kilic, and Kai M Schmidt-Ott. A grhl2-dependent gene network controls trophoblast branching morphogenesis. *Development (Cambridge, England)*, 142:1125–1136, March 2015. ISSN 1477-9129. doi: 10.1242/dev.113829.

[180] I C Scott, L Anson-Cartwright, P Riley, D Reda, and J C Cross. The hand1 basic helix-loop-helix transcription factor regulates trophoblast differentiation via multiple mechanisms. *Molecular and cellular biology*, 20:530–541, January 2000. ISSN 0270-7306.

[181] C M Chia, R M Winston, and A H Handyside. Egf, tgf-alpha and egfr expression in human preimplantation embryos. *Development (Cambridge, England)*, 121:299–307, February 1995. ISSN 0950-1991.

[182] M Monk and A Salpekar. Expression of imprinted genes in human preimplantation development. *Molecular and cellular endocrinology*, 183 Suppl 1:S35–S40, October 2001. ISSN 0303-7207.

[183] Ashreena Salpekar, John Huntriss, Virginia Bolton, and Marilyn Monk. The use of amplified cdna to investigate the expression of seven imprinted genes in human oocytes and preimplantation embryos. *MHR: Basic science of reproductive medicine*, 7(9):839–844, 2001.

[184] Christopher P Morgan, Jennifer Chan, and Tracy L Bale. Driving the next generation: Paternal lifetime experiences transmitted via extracellular vesicles and their small rna cargo. *Biological Psychiatry*, 2018.

[185] Vincenzo Calvanese, Ester Lara, Arnold Kahn, and Mario F Fraga. The role of epigenetics in aging and age-related diseases. *Ageing research reviews*, 8:268–276, October 2009. ISSN 1872-9649. doi: 10.1016/j.arr.2009.03.004.

[186] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics (Oxford, England)*, 32: 1832–1839, June 2016. ISSN 1367-4811. doi: 10.1093/bioinformatics/btw074.

[187] Polina Mamoshina, Armando Vieira, Evgeny Putin, and Alex Zhavoronkov. Applications of deep learning in biomedicine. *Molecular pharmaceutics*, 13:1445–1454, May 2016. ISSN 1543-8392. doi: 10.1021/acs.molpharmaceut.5b00982.

[188] David Capper, David T W Jones, Martin Sill, Volker Hovestadt, Daniel Schrimpf, Dominik Sturm, Christian Koelsche, Felix Sahm, Lukas Chavez, David E Reuss, Annekathrin Kratz, Annika K Wefers, Kristin Huang, Kristian W Pajtler, Leonille Schweizer, Damian Stichel, Adriana Olar, Nils W Engel, Kerstin Lindenberg, Patrick N Harter, Anne K Braczynski, Karl H Plate, Hildegard Dohmen, Boyan K Garvalov, Roland Coras, Annett Hlsken, Ekkehard Hewer, Melanie Bewerunge-Hudler, Matthias Schick, Roger Fischer, Rudi Beschorner, Jens Schittenhelm, Ori Staszewski, Khalida Wani, Pascale Varlet, Melanie Pages, Petra Temming, Dietmar Lohmann, Florian Selt, Hendrik Witt, Till Milde, Olaf Witt, Eleonora Aronica, Felice Giangaspero, Elisabeth Rushing, Wolfram Scheurlen, Christoph Geisenberger, Fausto J Rodriguez, Albert Becker, Matthias Preusser, Christine Haberler, Rolf Bjerkvig, Jane Cryan, Michael Farrell, Martina Deckert, Jrgen Hench, Stephan Frank, Jonathan Serrano, Kasthuri Kannan, Aristotelis Tsirigos, Wolfgang Brck, Silvia Hofer, Stefanie Brehmer, Marcel Seiz-Rosenhagen, Daniel Hnggi, Volkmar Hans, Stephanie Rozsnoki, Jordan R Hansford, Patricia Kohlhof, Bjarne W Kristensen, Matt Lechner, Beatriz Lopes, Christian Mawrin, Ralf Ketter, Andreas Kulozik, Ziad Khatib, Frank Heppner, Arend Koch, Anne Jouvet, Catherine Keohane, Helmut Mhleisen, Wolf Mueller, Ute Pohl, Marco Prinz, Axel Benner, Marc Zapatka, Nicholas G Gottardo, Pablo Herniz Driever, Christof M Kramm, Hermann L Mller, Stefan Rutkowski, Katja von Hoff, Michael C Frhwald, Astrid Gnekow, Gudrun Fleischhack, Stephan Tippelt, Gabriele Calaminus, Camelia-Maria Monoranu, Arie Perry, Chris Jones, Thomas S Jacques, Bernhard Radlwimmer, Marco Gessi, Torsten Pietsch, Johannes Schramm, Gabriele Schackert, Manfred Westphal, Guido Reifenberger, Pieter Wesseling, Michael Weller, Vincent Peter Collins, Ingmar Blmcke, Martin Bendszus, Jrgen Debus, Annie Huang, Nada Jabado, Paul A Northcott, Werner Paulus, Amar Gajjar, Giles W Robinson, Michael D Taylor, Zane Jaunmuktane, Marina Ryzhova, Michael Platten, Andreas Unterberg, Wolfgang Wick, Matthias A Karajannis, Michel Mittelbronn, Till Acker, Christian Hartmann, Kenneth Aldape, Ulrich Schller, Rolf Buslei, Peter Lichter, Marcel Kool, Christel Herold-Mende, David W Ellison, Martin Hasselblatt, Matija Snuderl, Sebastian Brandner, Andrey Korshunov, Andreas von Deimling, and Stefan M Pfister. Dna methylation-based classification of central nervous system tumours. *Nature*, 555:469–474, March 2018. ISSN 1476-4687. doi: 10.1038/nature26000.

[189] Safoora Yousefi, Fatemeh Amrollahi, Mohamed Amgad, Chengliang Dong, Joshua E Lewis, Congzheng Song, David A Gutman, Sameer H Halani, Jose Enrique Velazquez Vega, Daniel J Brat, and Lee A D Cooper. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific reports*, 7:11707, September 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-11817-6.

[190] Dongdong Sun, Minghui Wang, and Ao Li. A multimodal deep neural network for human breast cancer prognosis prediction by integrating multi-dimensional data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018.

[191] Kumardeep Chaudhary, Olivier B Poirion, Liangqun Lu, and Lana X Garmire. Deep learning-based multi-omics integration robustly predicts survival in liver cancer. *Clinical cancer research : an official journal of the American Association for Cancer Research*, 24:1248–1259, March 2018. ISSN 1078-0432. doi: 10.1158/1078-0432.CCR-17-0853.

[192] Muxuan Liang, Zhizhong Li, Ting Chen, and Jianyang Zeng. Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM transactions on computational biology and bioinformatics*, 12(4):928–937, 2015.

[193] Cancer Genome Atlas Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490:61–70, October 2012. ISSN 1476-4687. doi: 10.1038/nature11412.

[194] Katarzyna Tomczak, Patrycja Czerwiska, and Maciej Wiznerowicz. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology (Poznan, Poland)*, 19:A68–A77, 2015. ISSN 1428-2526. doi: 10.5114/wo.2014.47136.

[195] Sandeep K Singhal, Nawaid Usmani, Stefan Michiels, Otto Metzger-Filho, Kamal S Saini, Olga Kovalchuk, and Matthew Parliament. Towards understanding the breast cancer epigenome: a comparison of genome-wide dna methylation and gene expression data. *Oncotarget*, 7:3002–3017, January 2016. ISSN 1949-2553. doi: 10.18632/oncotarget.6503.

[196] Andrew E Teschendorff and Caroline L Relton. Statistical and integrative system-level analysis of dna methylation data. *Nature reviews. Genetics*, 19:129–147, March 2018. ISSN 1471-0064. doi: 10.1038/nrg.2017.86.

[197] Daudi Jjingo, Andrew B Conley, Soojin V Yi, Victoria V Lunyak, and I King Jordan. On the presence and role of human gene-body dna methylation. *Oncotarget*, 3:462–474, April 2012. ISSN 1949-2553. doi: 10.18632/oncotarget.497.

[198] Peter A Jones. Functions of dna methylation: islands, start sites, gene bodies and beyond. *Nature reviews. Genetics*, 13:484–492, May 2012. ISSN 1471-0064. doi: 10.1038/nrg3230.

[199] Eytan Zlotorynski. Epigenetics: Dna methylation prevents intragenic transcription. *Nature reviews. Molecular cell biology*, 18:212–213, March 2017. ISSN 1471-0080. doi: 10.1038/nrm.2017.25.

[200] Petar Veličković, Duo Wang, Nicholas D Lane, and Pietro Liò. X-cnn: Cross-modal convolutional neural networks for sparse datasets. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–8. IEEE, 2016.

[201] David A Schaer, Daniel Hirschhorn-Cymerman, and Jedd D Wolchok. Targeting tumor-necrosis factor receptor pathways for tumor immunotherapy. *Journal for immunotherapy of cancer*, 2:7, 2014. ISSN 2051-1426. doi: 10.1186/2051-1426-2-7.

[202] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[203] Martn Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: large-scale machine learning on heterogeneous systems. software available from tensorflow. org. 2015. *URL https://www. tensorflow. org*, 2015.

[204] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[205] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1025–1035, 2017.

[206] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[207] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, 50(8):96, 2018.

[208] Hui Xiao, Krzysztof Bartoszek, et al. Multi-omic analysis of signalling factors in inflammatory comorbidities. *BMC bioinformatics*, 19(15):439, 2018.