ACTIVE MODULES IDENTIFICATION IN MULTILAYER INTRACELLULAR NETWORKS

by

DONG LI

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
December 2017

Abstract

The network analysis has become a basic tool to gain insights on evolution and organization of living organisms in computational system biology. Since a group of genes may get involved into a biological process other than act alone, identifying modules or subnetworks from biological networks has been a central challenge to this field in the past decade. Several representative methods have been proposed to search such important modules using different intuitions while no unified framework exists yet, especially for multilayer networks, which can model gene expression dynamics and species conservation. This thesis provides a comprehensive study on active modules identification in multilayer intracellular networks, with the following main contributions:

- An improvement on a heuristic method for identifying active modules from protein-protein interaction (PPI) networks.
- A new objective of active modules to incorporate the topological structure and active property on the single layer and multilayer dynamic PPI network, and a convex optimization algorithm to solve it.
- A new definition for active modules in single layer and multilayer gene co-expression networks and a novel algorithm which achieves the state-of-the-art performance.
- A framework to conduct networks comparison via modules differentiation analysis, which can find condition-specific modules as well as conserved modules.

ACKNOWLEDGEMENTS

First of all, I want to express my sincere gratitude to my supervisor Dr. Shan He for his supervision, inspiration, and patience. His philosophy has influenced me a lot. Without his consistent guidance and support, I cannot present the thesis. I shall also thank his wife Dr. Liyun Zhang for helping my wife and me in life.

I am grateful to my thesis group members, Prof. Xin Yao and Dr. Ata Kabán, for their constructive comments and continuous support during my Ph.D. study.

I am happy to work with many collaborators: Prof. Mark Viant, Prof. John Colbourne, Prof. Ben Brown, Dr. Luisa Orsini, Prof. Graham Anderson, Prof. Zexuan Zhu, Prof. Zhisong Pan, Prof. Guyu Hu. Thank Prof. Michal Kočvara, Prof. Katya Scheinberg and Dr. Emanuele Rodolà for helpful discussion on optimization. I also benefit from open source campaign, especially the Bioconductor community.

I wish to thank Prof. Yaochu Jin from the University of Surrey and Dr. Ata Kabán for reading the thesis and being the examiners of my Ph.D. viva.

Thank School of Computer Science at the University of Birmingham for providing the excellent research environment. Thank the staff members for their work.

In the short and peaceful days in Birmingham, I have met several friends: Wen-Chi Yang, Guanbo Jia, Ahmed Al-Ajeli, Hector Basevi, Ning Shi, Koko Muroya, Fani Tsapeli, Simon Fong, Rafee Ibrahem, Lenka Mudrová, Bram Geron, Tomáš Jakl and all Shan He Lab members.

Last but not the least, I would like to thank my family members, especially my wife Lei Wang. This thesis is dedicated to them.

CONTENTS

1 Introduction **1**

1.1 Background 4

1.1.1 Gene expression data 4

1.1.2 Biological networks and graphs 5

1.1.3 Protein-protein interactions networks 7

1.1.4 Gene co-expression networks 8

1.2 Active modules identification 9

1.3 Multilayer network analysis 10

1.4 Research questions 12

1.4.1 Improvements on metaheuristics 12

1.4.2 Topological properties of active modules 13

1.4.3 Active modules for gene co-expression networks 13

1.4.4 Active modules for multilayer networks 14

1.4.5 Network comparison via modules 14

1.5 Thesis contributions 15

1.6 Thesis organization 16

2 Literature review **18**

2.1 Active module identification on connected graphs 18

2.1.1 Meta-heuristic methods 18

2.1.2 Exact approaches 20

2.1.3 Recap 24

2.1.4 Evaluation criteria 24
2.2 Modules detection on gene co-expression network 28
2.2.1 Communities detection on gene co-expression network 28
2.2.2 Weighted gene co-expression network analysis 30
2.3 Multilayer networks in computational biology 31
2.3.1 General multilayer networks analysis 31
2.3.2 Cross-species multilayer networks analysis 33
2.3.3 Multilayer co-expression network 35
2.4 Chapter summary 36

3 Active modules for protein-protein interactions network 38

3.1 Introduction 38
3.2 Active modules identification by an improved EA 40
3.2.1 Algorithms description 42
3.2.2 Empirical evaluation 47
3.3 Active modules identification via convex optimization 55
3.3.1 Optimization on single-layer connected graph 57
3.3.2 Aggregation on multilayer PPI network 62
3.3.3 Empirical evaluation 64
3.4 Chapter summary 71

4 Active modules for gene co-expression network 72

4.1 Introduction 72
4.2 Continuous optimization on single network 74
4.2.1 Continuous optimization formulation 75
4.2.2 Empirical evaluation 78
4.3 Continuous optimization on multilayer network 84
4.3.1 Algorithms 84
4.3.2 Empirical evaluation 86

4.4 Discussion . . . 92
4.4.1 Related works . . . 92
4.4.2 The optimization problem . . . 93
4.4.3 Difference with WGCNA . . . 93
4.5 Chapter summary . . . 94

5 Networks comparison **95**
5.1 Introduction . . . 95
5.2 Modules differential analysis . . . 97
5.2.1 Networks construction . . . 97
5.2.2 Modules detection . . . 98
5.2.3 Network comparison . . . 101
5.3 Empirical evaluation . . . 104
5.3.1 Simulated study . . . 104
5.3.2 Application to Ecology . . . 105
5.3.3 Application to Nanotoxicology . . . 106
5.4 Chapter summary . . . 108

6 Conclusions and Further Work **110**
6.1 Conclusions . . . 110
6.1.1 Definition of active modules . . . 111
6.1.2 Efficient algorithms . . . 112
6.1.3 Computational interpretation . . . 113
6.2 Limitations . . . 113
6.2.1 Limitations of active modules on PPI network . . . 114
6.2.2 Limitations of AMOUNTAIN . . . 114
6.2.3 Limitations of MDOA . . . 115
6.3 Future Work . . . 115
6.3.1 Overlapping and hierarchical modules detection . . . 115

6.3.2 Heterogenous multiple layers 116
6.3.3 High-order graph features 117
6.3.4 General networks comparison 117

Appendix A: A brief introduction to several algorithms **118**

A.1 Simulated annealing 118
A.2 Genetic algorithm 119
A.3 Linear programming 120

Appendix B: Supplementary to Chapter 3 **123**

B.1 Convergence analysis of Algorithm 7 123
B.2 Supplementary modules and GO terms 126

Appendix C: Supplementary to Chapter 4 **127**

C.1 NP-hardness of problem (5) 127
C.2 Piecewise root finding for Euclidean projection on elastic net 128
C.3 Comparison with non-convex optimization 130
C.3.1 Modified algorithm on single layer WGCN 130
C.4 Supplementary Tables 131

Appendix: List of References **132**

Acronyms

PPI Protein-Protein Interactions

DPPI Dynamic Protein-Protein Interactions

DPIN Dynamic Protein-Protein Interactions Networks

DNA Deoxyribonucleic Acid

RNA Ribonucleic Acid

mRNA Messenger RNA

miRNA microRNA

RNA-Seq RNA sequencing

GWAS Genome-wide association study

GEO Gene Expression Omnibus

DEGs Differentially Expressed Genes

BioGRID Biological General Repository for Interaction Datasets

STRING Search Tool for the Retrieval of Interacting Genes/Proteins

GCN Gene Co-expression Network

WGCN Weighted gene co-expression network

WGCNA Weighted gene co-expression network analysis

TOM Topological Overlap Matrix

STN Signal Transduction Networks

MWCSP Maximum-Weight Connected Subgraph Problem

PCST Prize-Collecting Steiner Tree Problem

ILP Integer Linear Programming

QP Quadratic Programming

EA Evolutionary Algorithm

GA Genetic Algorithm

SA Simulated Annealing

MA Memtic Algorithm

GO Gene Ontology

FDR False Discovery Rate

KEGG Kyoto Encyclopedia of Genes and Genomes

RHS recurrent heavy subgraph

MSCR Multi-Stage Convex Relaxation

CCF Connected Components Finding

BFS Breadth-first Search

DREAM Dialogue for Reverse Engineering Assessments and Methods

AMOUNTAIN Active modules for multilayer networks: a continuous optimization approach

MODA Modules differential analysis

LIST OF FIGURES

1.1 Schematic of a general multilayer network, modified from [Mucha et al., 2010]. There are four layers, each layer is a network. The nodes in each layer may be different, and interlayer interactions are shown as red dash lines. 11

1.2 The main contents of this thesis 17

3.1 Modules identified by GA on simulated data. The red nodes are not connected though they are supposed to be. 47

3.2 Modules identified by modified GA with proposed decoding scheme on the same simulated data as in Figure 3.1. The red nodes are connected. 48

3.3 Convergence rate comparison of three algorithms: SA, GA and MA from one trail. Technically, we record the objective value every iteration of all three algorithms, but from Table 3.1, the total iterations are different. We sample the objective every 1000 fitness evaluations for three algorithms. MA is the first to reach the stable status and also the highest module score. 49

3.4 The construction process of PPI network from two updated databases STRING and BioGRID. 51

3.5 The largest connected component of PPI network from BioGRID 51

3.6 The largest connected component of PPI network from STRING 52

3.7 The first identified module plotted by STRING, where edges represent both known interactions including curated databases and experimentally determined and predicted interactions such as gene neighborhood and gene co-occurrence. Colored nodes standard for query proteins and first shell of interactors, and white nodes for second shell of interactors. 54

3.8 The relatively small module plotted by GeneMANIA, where most of the edges are co-expression links according to previous studies. 55

3.9 Comparison of the ground truth with partition by spectral algorithm (3.16) and proposed method (3.9), on the widely used Zachary karate club network. Red nodes mean the members in the target module. The node size in the middle figure means node score. 65

3.10 The identified modules by `BioNet` and Algorithm 7. The red nodes are shared by both methods, yellow for `BioNet` and green for Algorithm 7. The shape of nodes indicates score, squares indicate negative scores and circles for positive. Algorithm 7 is able to connect more low-scored regions which may have close relationship with the biological mechanisms. 67

3.11 The F-measure of identified complexes from each time point layer, consensus graph and the baseline algorithm. 'CYC-P' means the precision evaluated by CYC2008 [Pu et al., 2008] and 'MIPS-R' means the recall evaluated by MIPS [Mewes et al., 2004]. 'F' is the F-measure defined as in (3.19). 'SPICi-C' is SPICi [Jiang and Singh, 2010] on the consensus graph. 70

4.1 Parameters selection for module identification on large network with 10000 nodes and the module contains 100 nodes. With grid search we find the optimal $\alpha \in [0.3, 0.4]$ and $\log(\lambda) \in [-5, -1]$ lead to $F = 1$. 79

4.2 The proportion of differentially expressed genes in the first identified module, using different λ value in the objective (4.2). The module size is strictly constrained to be 100. 81

4.3 The first identified active module of single layer co-expression network, and the DEGs. In the module, node sizes are proportional to the intensities of gene activities and edge widths to the correlation coefficients (cut-off at 0.8). The green nodes are shared by identified module and DEGs, and the red nodes are only in DEGs. 84

4.4 The first identified module in the human layer at 1 hour time point, plotted by STRING, where edges represent the known interactions. Colored nodes standard for query proteins and first shell of interactors, and white nodes for second shell of interactors. 89

4.5 The first identified module in the mouse layer at 1 hour time point, plotted by STRING, interactions are denser compared with human layer. Key transcriptional factors Stat2/Irf1 are densely surrounded by interactions. . 90

4.6 The first identified conserved module for a three layer network where each layer represents nodes from 12h, 24h and 48h respectively. The red nodes are two probes of gene RORC, a signature gene w.r.t. Th17 lineage commitment. Plotted by `igraph` [Csardi and Nepusz, 2006]. 92

5.1 Scatter plot of variable X_1 and X_2. Removing or adding the last two data points would make an impact on their correlation coefficient. 99

5.2 An score based function to pock the optimal height of hierarchical clustering tree in `MODA`. The right hand figure shows the cutting maximizes average partition density of resulted modules. 101

5.3 Overview of `MODA`. (a) Condition-specific networks construction from a set of samples. (b) Modules identification using various methods. (c) Module differential analysis based on the similarity of each two modules from background network and condition-specific network. 103

5.4 Levelplot of correlation matrix calculated on simulated gene expression profile X_1. There are five modules, each module has roughly identical size 100. But the last two modules contain similar samples. 105

5.5 Levelplot of correlation matrix calculated on simulated gene expression profile X_2. There are four modules, and each of three has the size 100 but the last module has larger size 200. 105

5.6 A combination bar plot of statistics of frequency about which module can be condition specific and conserved. Modules shown at the bottom are from background network, and both modules from background network and each condition-specific networks are identified by Louvain algorithm with module size constrained to be 50-200. 107

5.7 Gene co-expression network of module 104, in which the red genes are DE genes and edge width means the correlation coefficients. Edges with correlation value less than 0.7 are removed for visualization. Plotted by R igraph package. 109

B.1 The Euclidean projection of a vector $\mathbf{g} \in \mathbb{R}^n$ onto the feasible region $\Omega = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{x}^\top \mathbf{1} = 0, \ \|\mathbf{x}\|_2^2 \leq 1\}$, which is a intersection of a hyperplane and a ℓ_2-ball, and the boundary of ℓ_2-ball in the $(n-1)$-dimensional space. 125

LIST OF TABLES

1.1 Typical biological networks 6

2.1 The confusion matrix of single module identification. 25
2.2 Illustration of hypergeometric in GO enrichment analysis. 26

3.1 A comparison between three algorithms based on the number of fitness evaluations. 49
3.2 Enrichment analysis result of the first module identified by Algorithm 4 at 2h time point. 53
3.3 The top 5 enriched KEGG pathways of the modules identified by Algorithm 1, and heuristics algorithm which targets at a group of high-scored and connected nodes, on single human PPI network. 68
3.4 Overview of the resulted complexes of Algorithm 7 and SPICi [Jiang and Singh, 2010]. *#Items* means the total GO terms of all complexes and *Ration* is proportion of complexes which are significantly enriched by at least one GO term at a given FDR ($\leq$0.05). 71

4.1 Best performances of two methods in different networks, where n is the size of network and k is the size of true module. F score values (mean $\pm$ std) are calculated based on 50 runs. 79
4.2 Overview of top 10 modules identified from single layer WGCN of human, at 2 hour time point. PPI P-value indicates if there are significant known protein interactions in the module. 83

4.3 Modules identified from two species, using objective (6) in main body when $\lambda_1 = 1$ and $\lambda_2 = 1$. The second and third columns are module size and last column are the number of conserved genes. 88

4.4 The first identified module at 1 hour for human and mouse, when $\lambda_3 = 1000$ in the objective function and 52 conserved genes are found. 91

5.1 Parameters in the pipeline of `MODA`, on Daphnia magna microarray data GSE35150. 106

6.1 Differences between active modules identification on intracellular networks discussed in this thesis. 111

A.1 How simulated annealing is inspired by annealing in metallurgy (* or achieves maximal iteration) 119

A.2 How genetic algorithm is inspired by natural selection 120

CHAPTER 1

INTRODUCTION

With the increased use of high-throughput experimental data such as genomics, transcriptomics and proteomics data, our understanding of living organisms has advanced at the molecular level. Due to the similar organization and evolutionary behavior of intracellular compounds, the network-based approaches have been intensively used in computational system biology in the past decade [Barabasi and Oltvai, 2004; Barabási et al., 2011]. There have been lots of works focused on exploring global and local topological properties of biological networks, such as community structure [Girvan and Newman, 2002] and network motifs [Milo et al., 2002], which were shown to have a close connection with diverse biological functions. As basic methodologies, existing tools in conventional graph theory as well new approaches in complex networks have been applied in kinds of biological networks analysis, which continuously contribute to this field.

The topology of a biological network does not always precisely reflect the function or disease-determined regions of intracellular networks [Barabási et al., 2011], which are the real concerns in biology. As stated above, community (or module, subnetwork) detection is one of the most important subjects in network analysis. The topological modules that are purely derived from topology, only overlap with the functional module to some extent. Moreover, the topological module identified by community detection ignore the activity of the biological components (i.e., nodes), which cannot depict the dynamics of the intracellular. A straightforward example is that, as the basic infrastructure, a protein-

protein interactions (PPI) network keeps unchanged for a certain species under different cellular responses, but the gene expression may change dramatically, which reflects the mechanism behind these intracellular activities. As a result, modules identified through conventional graph partitioning or clustering could provide limited information about the module-mechanism association.

To better address these concerns, the **active modules** identification, which aims to find connected regions of the network showing striking changes in molecular activity or phenotypic signatures that are associated with a given cellular response in biological networks [Mitra et al., 2013], has become an important issue in network biology. The active modules not only consider the nodes attribution which may reveal the regulatory and signalling mechanisms of a given cellular response, but also cover the topological features such as connectedness. [Ideker et al., 2002] proposed the first general method to discover active connected regions by given network with scores on each node, which reflects the changes in particular conditions. The seminal work [Ideker et al., 2002] defined an optimization problem on a weighted molecular interaction network, which incorporates basic network structure with high-throughput omics data, and proposed a simulated annealing algorithm to solve it. Although there are significant advances in the past decades (See a comprehensive review by [Mitra et al., 2013]), there are three main challenges need to be addressed to make active module a more useful tool to reveal mechanisms of biological phenomena.

Identifying a connected subgraph with maximal activity has been proven to be NP-hard. Heuristic methods [Ideker et al., 2002] are generally suited for such kind of problems but brings several drawbacks in representation and efficiency. With relaxation or approximation, the original combinatorial problems can be transformed into integer linear programming [Dittrich et al., 2008], but such mathematical programming suffers from scalability issue. In other words, identifying modules from a general graph is difficult, especially when the graph is large.

The second challenge is how to extend active modules identification to other types of

biological networks, e.g., gene co-expression networks as complete graphs [Li et al., 2011]. Compared with increasing vast amount of high-throughput omics data, the speed of constructing reliable and complete PPI networks [Szklarczyk et al., 2014; Chatr-Aryamontri et al., 2015] and metabolic networks [Thiele and Palsson, 2010], which heavily rely on experiment and human validation, is quite slow. For some non-model species or some new model species such as Daphnia [Orsini et al., 2016], the PPI networks do not even exist. The lack of reliable networks poses a challenge to the detection of informative and active modules to reveal the true molecular mechanisms in the biological systems. In contrast, the gene co-expression network [Stuart et al., 2003] is a pure data-driven gene network, which only relies on gene expression profile. The idea of combining network structure and omics data is also suitable for gene co-expression networks.

The third challenge is how to identify active modules from multilayer networks. Multilayer networks [Kivelä et al., 2014] provide a general framework to model temporal and spatial change of interactions, at least contributing three aspects for current network biology research: 1) The multilayer network could model the dynamic properties for biological process, 2) The multilayer network could reflect responses of the overall system under multiple conditions and 3) The multilayer network provides a framework for gene orthologous mapping and helps the identification of core genes across species. Identifying active modules from above multilayer networks is supposed to find unchanged part of vertices across layers as well as layer-specific part of vertices.

This thesis addresses the above three interrelated challenges. We adopt the new developments in multilayer networks and optimization techniques, aiming to provide a unified active modules identification framework which can help biologists to generate more reliable testable hypothesis from biological networks. We study several popular intracellular networks and their multilayer counterparts, and validate the proposed methods in biological scenarios. As direct outcomes of the study, we implement these algorithms as several practical tools for more generalized usage.

The rest of this chapter introduce several specific aspects involved in this thesis. Sec-

tion 1.1 gives out some background knowledge about the data and networks we use in the thesis. Section 1.2 formally defines the underlying problem of identifying active modules from an interaction network, followed by a brief introduction of multilayer networks in Section 1.3. Subsequently, we explain the research questions and the motivations in Section 1.4. Finally, section 1.5 states the main contributions of this thesis and section 1.6 illustrates the thesis structure.

1.1 Background

1.1.1 Gene expression data

Gene expression happens when a gene is used in the synthesis of a functional gene product, including proteins and functional Ribonucleic Acid (RNA). The principle of expression profiling is to quantify the changing expression levels of each transcript during development and under different conditions [Wang et al., 2009]. Advances in experimental molecular biology brought microarray and sequence-based profiling techniques in succession, which can efficiently quantify the transcriptome and produce a large amount of gene expression data [Brazma and Vilo, 2000; Mortazavi et al., 2008]. Here we do not examine the difference between microarray and sequence-based transcriptome profiling in details, but only use the term **gene expression data** at the abstract level.

We represent the gene expression data as an expression matrix X, in which each entry x_{ij} represent the expression value of gene i in sample j, as (1.1) shows. More commonly, we may use log ratio $x_{ij} = log_2(T_{ij}/R_{ij})$ in practice, where T_{ij} is the mRNA expression level of gene i in experimental group and R_{ij} is the level of control group. We can see the larger value x_{ij} is, the more significant gene i in sample j expressed. The samples contain

a whole genome at all conditions, which depend on the experimental design.

$$
\begin{array}{c}
genes \\
\\
\\
\\
\\
\end{array}
\begin{bmatrix}
x_{11} & x_{12} & \dots & x_{1m} \\
x_{21} & x_{22} & \dots & x_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \dots & x_{nm}
\end{bmatrix}
\\
samples
\tag{1.1}
$$

Microarray and sequence-based expression data are publicly available at Gene Expression Omnibus (GEO, `https://www.ncbi.nlm.nih.gov/geo`) [Edgar et al., 2002], with a set of tools for programmatic access and simple analysis. The repository covers a wide range of species and experiments. All gene expression data in this thesis come from GEO, specified by the GEO accession numbers, such as GSE35103. In general, the samples replicate number for each treatment in a specific experiment is limited due to costs.

Gene expression data has been used in numerous computational biology problems. Specifically, in network biology, the expression value can be used to score individual genes at certain condition. The expression matrix made up by experimental group and control group can be used to filter out some differentially expressed genes (DEGs), which may play important roles in a gene network. Furthermore, a *wide* expression matrix can be used to build gene co-expression network, as introduced below.

1.1.2 Biological networks and graphs

As a well-known fact, a group of genes may get involved into a biological process other than acting alone [Barabási et al., 2011], a systematic modeling on their interactions is thus useful. As a general tool to model the components and their interactions, the graph (network) representation is a natural choice in the biological system. Network analysis thus becomes a fundamental tool in computational system biology in the past years.

A network is normally referred as a graph $G = (V, E)$, which consists of a set of vertices V and edges E. Weights can be assigned on vertex $v_i \in V$ or edge $e_{ij} \in E$, and the

interactions may exist between some pair of vertices. In computational biology, vertices and edges in different kinds of networks represent different compounds or interactions. For example, nodes in protein-protein interactions (PPI) networks are proteins, and edges can be physically direct interactions between proteins, experimentally determined interactions or functional cooccurrence [Szklarczyk et al., 2014]. In contrast, the nodes in gene co-expression networks are simply genes, and the edges measure the similarities between genes by pairwise correlation. As a complementary, there are some other networks also under the umbrella of network biology, such as regulatory network, a directed graph modeling the regulation relationship between genes and metabolic network, a bipartite graph consisting of metabolites and physical processes. The meaning of vertices and edges in popular biological networks are summarized as in Table 1.1. This thesis focuses on PPI networks and gene co-expression networks, as two kinds of different graphs.

Network	Graph	Nodes	Edges	Direction
PPI	Connected	Proteins	Physical interaction	Undirected
Co-expression	Complete	Genes	Similarities	Undirected
Regulatory	Connected	Genes	Regulation	Directed
Metabolic	Bipartite	Metabolites and reactions	Metabolic pathways	Undirected

Table 1.1: Typical biological networks

Various computational techniques have been applied to these biological networks. A general methodology for biological networks analysis is to explore the properties of corresponding graphs, and further relate them to organizing principles and mechanisms. These analysis cover from basic topological parameters describing the topological structure of network [Assenov et al., 2008], such as network size, network diameter, radius, density, node degree, etc. From a systematic viewpoint, emergent behavior [Bhalla and Iyengar, 1999] and modular structure [Girvan and Newman, 2002; Spirin and Mirny, 2003] have been studied, such as dynamics [Prill et al., 2005] and hierarchical structure [Ravasz et al., 2002], which are further related to network organization and biological functions. From these examples, we can see that the basic principle of biological network analysis is a combination of graph theory and background knowledge.

1.1.3 Protein-protein interactions networks

In protein-protein interactions networks, vertices are proteins, and edges represent protein-protein interactions, which are physical contacts between two or more proteins that occur in a cell or in a living organism in vivo [De Las Rivas and Fontanillo, 2010]. Generalized protein-protein interactions include direct (physical) as well as indirect associations, such as functional associations [Szklarczyk et al., 2014]. These interactions may refer a functional relationship that a pair of proteins contributes to a common biological process.

The network structure of protein-protein interactions in yeast has been reported in [Schwikowski et al., 2000], and the topological features along with functional modules have been discovered in such network [Jeong et al., 2001]. As certain graphical properties and functional modules reveal the association behind complex biological processes, modules identification [Mitra et al., 2013] and other analysis on PPI thus become important to understand the mechanisms of living organisms.

There are several databases and repositories providing protein-protein interactions for a wide range of species [De Las Rivas and Fontanillo, 2010], from which we can construct PPI networks. Most of them include numerical scores or confidence for the interactions, these PPI networks can be weighted in terms of edges. Take *homo sapiens* for example, BioGRID [Chatr-Aryamontri et al., 2015] and STRING v10.0 [Szklarczyk et al., 2014] are two widely used and updated databases. Specifically, the BioGRID has 362,775 interactions while STRING stores 8,548,002 protein pairs, with a combined score ranging from 150 to 999 for each link. The number of reported protein interactions has also been growing from different model organism species [Chatr-aryamontri et al., 2016].

The PPI networks from different databases have been serving as basic network infrastructures in numerous applications, due to their high accessibility for many species. Specifically, in active modules identification, a PPI network is considered as the primary choice. Moreover, modules from a PPI network are relatively easy to interpret since it has been intensively studied to relate a group of proteins with prior knowledge.

1.1.4 Gene co-expression networks

A gene co-expression network (GCN) [Stuart et al., 2003] is an undirected graph, where the edge can be weighted as co-expression degree or unweighted as to be normalized. The nodes stand for genes and edge measure the co-expression between gene pairs. By the definition, the gene co-expression can be constructed with only gene expression data: given a gene expression matrix as (1.1), a symmetric edge weight matrix $W \in \mathbb{R}^{n \times n}$ can be used as adjacency matrix for a graph, where $w_{i,j} = corr(x_i, x_j)$.

Weighted gene co-expression network analysis (WGCNA) [Langfelder and Horvath, 2008] has been widely used in the past decade. Moreover, network properties exploration and modules detection have been unified in the framework of `WGCNA` [Zhang and Horvath, 2005]. Modules detection on such networks is normally based on hierarchical clustering which takes the similarity matrix as input, followed by modules function association if additional sample trait information is available. Being different from PPI network analysis, in which the networks are normally assumed as ready by databases, the gene co-expression networks need to be constructed from expression profiles. Several popular similarity metrics have been compared in [Kumari et al., 2012].

In summary, the typical process of weighted gene co-expression network analysis includes three steps:

- Network construction: to construct the network from gene expression profiles as microarray or RNA-Seq. Pearson correlation is a commonly choice [Zhang and Horvath, 2005], and gene similarity is measured by a power adjacency function $s_{ij} = |cor(x_i, x_j)|^{\beta}$, where x_i and x_j are expression vectors for gene i and gene j, β is the power parameter needs to be tuned to make connectivities of all nodes fit scale-free topology criterion [Zhang and Horvath, 2005].

- Network analysis: to extract basic topological features and active modules. A module in a weighted gene co-expression network is considered as a group of similar nodes, in which the nodes densely interact with each other. Clustering or graph

partition can be applied. The popular software `WGCNA` uses a straightforward hierarchical clustering on a specialized similarity matrix, covering all nodes.

- Result interpretation: to associate these features or patterns with biological process, pathways or diseases. `WGCNA` suggests relating modules with phenotypic traits, which could possibly point out which module or significant gene is responsible to certain phenotypic feature. In addition, integrative gene set enrichment analysis also provide a guideline to explain the functions and processes associated with the module, denoted by a gene list.

1.2 Active modules identification

Active modules identification emerged from graph based representation of genes interaction and mRNA expression level based gene annotation [Ideker et al., 2001]. A natural hypothesis about the underlying mechanism behind the changes in gene expression is that, if we can find the connected regions (modules) of the network that show strike changes actually govern the expression of genes, under certain response [Ideker et al., 2002].

To be more general and abstract, we present the PPI network or other interaction networks as an undirected and connected graph $G = (V, E)$, where nodes in V represent proteins or genes, and edges in E represent the interaction relationships between two nodes. Node activities can be measured by assigning each node i a single score to denote the activity of the corresponding component in a certain condition, such as the fold-change or the p-value, which measure the gene expression level contrasted with the control group. It is required to find a connected subnetwork with largest activity. Based on the definition of connected graph, this problem is formally defined as Problem 1, which has been proven to be NP-hard [Karp, 1972; Ideker et al., 2002].

Definition. Connected graph. A graph is connected when there is a path between every pair of vertices.

Problem 1. Given a graph $G = (V, E)$ with vertices weights $\mathbf{z} \in \mathbb{R}^{|V|}$ for each $v \in V$, find a connected subnetwork $S = (V_S, E_S)$ of G with maximal weight $f(S) = \sum_{v \in V_S} z_v$.

As effective tools to solve the combinatorial problem, metaheuristic algorithms have been widely applied to search solutions. The original paper [Ideker et al., 2002] proposed to use simulated annealing algorithm to address this problem. Other methods include extended simulated annealing [Guo et al., 2007], greedy algorithm [Ulitsky and Shamir, 2009], graph-based heuristic algorithm [Rajagopalan and Agarwal, 2005], genetic algorithm [Ma et al., 2011] and some exact approaches based on integer linear programming [Dittrich et al., 2008; Backes et al., 2012]. A more detailed review on the theses existing methods is in Chapter 2.

1.3 Multilayer network analysis

Multilayer network analysis [Kivelä et al., 2014] has received much attention due to emergent case studies arose in several disciplines, which all take the multiple layers into consideration to improve our understanding of the complex systems. Some important concepts such as modularity [Mucha et al., 2010], dynamics [Boccaletti et al., 2014] have been extended to multilayer networks. Figure 1.1 shows an example of a general multilayer network.

In the field of computational biology, several attempts have been made [Li et al., 2011; El-Kebir et al., 2015; Zinman et al., 2015] to mine conserved modules across multiple networks or multiple layers. More concrete scenarios when multilayer networks are useful are illustrated as below.

Dynamics. Although the behaviors of living organisms were considered to be dynamic, traditional network-based methods primarily focused on static network, which is a snapshot of the real case. A multilayer network provides a powerful tool for modeling this time series networks [Mucha et al., 2010], with each layer standing for a time point. Protein complexes and functional modules identified from dynamic PPI network

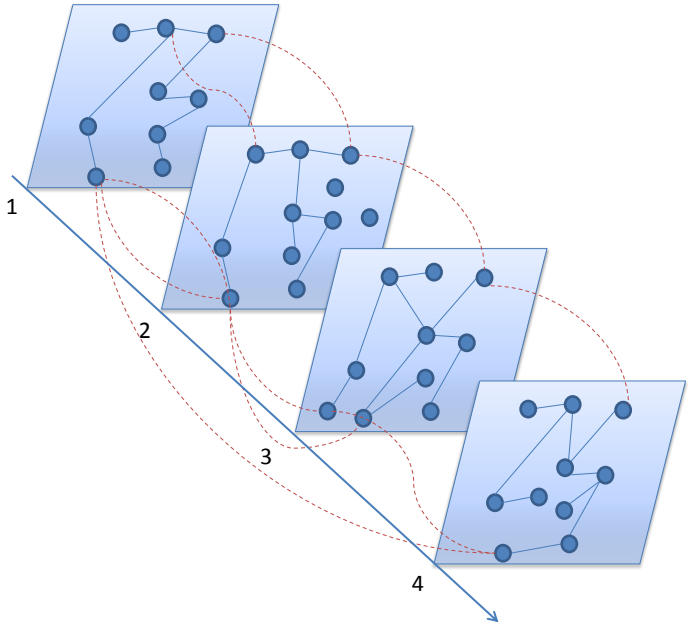


Figure 1.1: Schematic of a general multilayer network, modified from [Mucha et al., 2010]. There are four layers, each layer is a network. The nodes in each layer may be different, and interlayer interactions are shown as red dash lines.

[Wang et al., 2014] have led to more biological information discovery. The time-invariant part of each layer, also referred as the conserved module in dynamic multilayer network may reveal important functions. While time-point specific module may be related to the life-time of a certain procedure.

Orthologs. The multilayer network can also capture the core gene set across species, such as conserved active modules [Stuart et al., 2003; Deshpande et al., 2010]. By constructing multilayer network with each layer representing one species, we may find similar patterns in a species that has relatively more prior information, thus to gain biological knowledge of interested species. Finding conserved modules may also improve our understanding of the evolutionary biological procedure by highlighting the similarities and differences in key patterns between species [El-Kebir et al., 2015; Zinman et al., 2015].

Conditions. Besides dynamic and cross-species multilayer networks, comparing multiple networks or multislice network [Carchiolo et al., 2011] under multiple conditions is also included in multilayer network analysis. The networks comparison framework can be

regarded as multilayer network system, where each layer is a condition-specific network. Since there are no explicit inter-layer links to be utilized, we can also consider the network comparison as a following-up operation after modules identification.

1.4 Research questions

Based on the introduction of biological networks, active modules identification problem and multilayer networks, this thesis provides on a general framework of identifying active modules from multilayer intracellular networks. Specifically, this thesis tries to answer five research questions as explained in the following sections.

1.4.1 Improvements on metaheuristics

As stated in section 1.2, metaheuristics such as evolutionary algorithms (EA) have been used in active module identification [Ideker et al., 2002; Ma et al., 2011]. These algorithms used binary encoding, in which the '1' positions indicate the module membership while '0' means background. How to ensure the connectivity of resulted module without losing efficiency is important to such solutions, since the binary encoding on the connected graph itself does not guarantee the connectivity of '1' positions. Existing works introduced more constraint to tackle this problem [Ideker et al., 2002], which caused indispensable overhead. Some other works even failed to pay attention of this issue. For example, [Ma et al., 2011] adopted a genetic algorithm which maximizes a summed nodes and edges scores but ignored the connectivity constraints. As a result, a group of isolated nodes may be found.

Another issue for existing metaheuristics lies at computational efficiency. With basic evolutionary algorithms, one can only obtain feasible results on small graphs. How to develop more efficient algorithms by incorporating some more advanced techniques is also to be explored. Based on these considerations, the first research question is

How can we make improvements on metaheuristics for active modules identification?

We try to answer this question with simple strategies, as the first attempt in studying active modules identification on PPI networks. The details are described in Chapter 3, section 3.2.

1.4.2 Topological properties of active modules

The original definition of active modules [Mitra et al., 2013] poses no topological requirements other than connectivity. But evidence shows that pure topological features, such as communities [Girvan and Newman, 2002] or motifs [Milo et al., 2002], also lead to meaningful result [Palla et al., 2005]. Especially for the communities, which may correspond to functional units [Newman, 2013]. Combining the more topological features as community structure might help to identify active modules more precisely. Accordingly, the second research question is

Is it beneficial to consider community structure of active modules?

The contribution about incorporating topological structure and module activity will be discussed in Chapter 3, section 3.3.

1.4.3 Active modules for gene co-expression networks

Inspired by intensive works in active modules identification in connected graphs such as PPI networks and metabolic networks, we started to study the gene co-expression networks, another popular type of intracellular networks which can be represented as complete graphs. Compared with a formal definition of active modules on PPI network, active modules for gene co-expression networks are different due to the essential differences between a complete graph and a connected graph. The third research question:

Is it useful to define and identify active modules in weighted gene co-expression networks such as in PPI networks?

Whether a group of genes extracted from a co-expression network has significant biological meanings which will be answered in Chapter 4, section 4.2.

1.4.4 Active modules for multilayer networks

As introduced above, there are various reasons to model complex systems as multilayer networks. Take the dynamic property for example, although the main databases provide protein-protein interactions as static repositories, the real in the living organism is considered to be dynamic [Taylor et al., 2009], both in terms of activities of certain nodes and modular structure of the network. And the gene expression is also dynamic if we consider the gene co-expression networks. Active modules identification for multilayer networks has potential values to reveal functions across different layers. However, there is no existing algorithm for active module identification from multilayer networks. Therefore, our fourth research question is:

> How can we identify active modules from a multilayer network?

Considering the fundamental differences between WGCNs and PPI networks, we try to answer the above question in two chapters: Active modules identification multilayer dynamic PPI network is discussed in Chapter 3, section 3.4. And active modules for dynamic multilayer gene co-expression network is discussed in Chapter 4, section 4.3.

1.4.5 Network comparison via modules

Although there are several methods for biological networks comparison [Sharan and Ideker, 2006; Pržulj, 2007], existing methods take the whole network as a unit, or extract high-level topological features. Such comparison ignores structure information, an important characteristics of biological networks. Therefore, the fifth research question goes like:

> After getting modules for a set of networks, is it useful to compare them at a modular level?

The comparison aims to reveal the differences between each condition-specific network and the background network. And the central idea is to reveal the differences via modules instead of individual vertices or edges, which may help to gain high-level insights that focus on functional units. And this aspect is discussed in Chapter 5.

1.5 Thesis contributions

This thesis provides a comprehensive study of active modules identification on multilayer intracellular networks, and contributes the field from the following four aspects:

- On single layer PPI network, we propose a memetic algorithm with a new binary decoding scheme which ensures the connectivity of identified modules, which can be considered as an improvement to heuristics. (Chapter 3, section 3.2.)
- We incorporate the topological structure and active property to derive a new objective of the active module on the single layer and multilayer dynamic PPI network and solve it by a continuous optimization algorithm. (Chapter 3, section 3.3)
- On single layer and multilayer gene co-expression network, we define the active modules and develop a continuous optimization approach to identify them, which achieves the state-of-the-art performances. (Chapter 4)
- Given gene expression profile sampling from a set of different conditions, we develop a package to conduct modules differentiation analysis, which is able to find condition-specific modules as well as conserved modules. (Chapter 5)
- Based on the proposed algorithms and methods, we contributed two R Bioconductor packages, `MODA` at `https://bioconductor.org/packages/MODA` and `AMOUNTAIN` at `https://bioconductor.org/packages/AMOUNTAIN`, and several open-source repositories on GitHub `https://github.com/fairmiracle`.

This thesis summarizes the contribution from the following publications:

– [Li et al., 2017a] *Active module identification in intracellular networks using a memetic algorithm with a new binary decoding scheme*, D. Li, Z. Pan, G. Hu, Z. Zhu and S. He. BMC Genomics (2017) 18(2): 209.

– [Orsini et al., 2017] *Early transcriptional response pathways in Daphnia magna are coordinated in networks of crustacean specific genes*, L. Orsini, J. Brown, O. Solari, D. Li, S. He, et al. Molecular Ecology (2017).

– [Li et al., 2016b] *Active modules for multilayer weighted gene co-expression networks: a continuous optimization approach*, D.Li, Z. Pan, G. Hu, G. Anderson and S.He. bioRxiv (2016), 056952. *Submitted.*

– [Li et al., 2016a] *MODA: MOdule Differential Analysis for weighted gene co-expression network*, D.Li, J.Brown, O.Solari Z. Pan, G. Hu, and S.He. bioRxiv (2017), p.053496.

– [Li et al., 2017b] *Extracting active modules from multilayer PPI network: a convex optimization approach*, D.Li, Z. Pan, G. Hu, Z.Zhu and S.He. *Submitted.*

1.6 Thesis organization

This thesis consists of six chapters. The first chapter introduces the background and the importance of this topic. We state the research questions to be studied and summarize the contributions.

In chapter 2, we review part of important literature in related fields. Section 2.1 introduces widely used modules identification methods on connected graphs. Section 2.2 reviews related works in weighted gene co-expression network. Section 2.3 reviews advances in multilayer network research.

In chapter 3, we study active modules identification on the protein-protein interactions network, which is a weighted and connected graph. Section 3.1 describes the motivation and defines the problem. Section 3.2 proposes a novel memetic algorithm to ensure the

connectivity of identified modules on PPI network. Section 3.3 derives a new objective of active modules and extends it to a multilayer dynamic network.

In chapter 4, we study active modules identification on the gene co-expression network, which is a weighted and complete graph. Section 4.1 defines the formal problem on this kind of graph and reviews related works. We propose a continuous optimization algorithm to mine an active module with maximal edge weights as well as vertex weights in section 4.2, and extend it to the multilayer case in section 4.3.

In chapter 5, we propose a general framework for modules comparison among different networks. Section 5.1 reviews related works and point out the gap between existing method and the need for exploring multiple conditions. Section 5.2 describes the general framework, and section 5.3 validates the proposed method on the simulated dataset and real-world dataset.

The last chapter concludes the thesis and proposes several promising directions.

The structure of this thesis is illustrated in Figure 1.2.

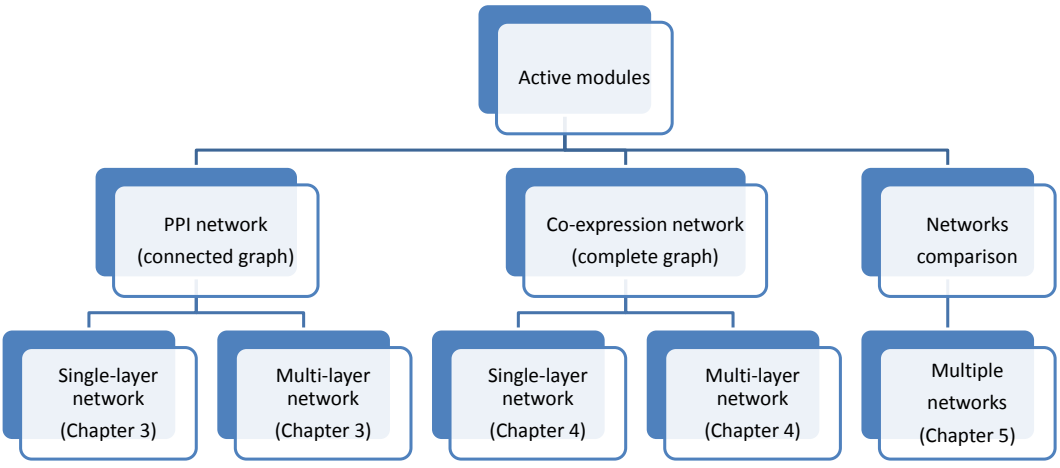


Figure 1.2: The main contents of this thesis

CHAPTER 2

LITERATURE REVIEW

This chapter reviews related works mentioned in Introduction chapter and gives out existing research in modules identification on single and multilayer networks. Section 2.1 reviews existing methods on active modules identification on connected graphs such as protein-protein interaction networks, which is a well-defined combinatorial optimization problem. Section 2.2 reviews representative works in weighted gene co-expression network analysis. Section 2.3 reviews advances in multilayer networks, including general analysis and modules identification. Section 2.4 summarizes the chapter.

2.1 Active module identification on connected graphs

2.1.1 Meta-heuristic methods

Problem 1 in section 1.2 formally defines optimization problem of active module identification on connected and (vertex) weighted graphs. To solve this problem, one of the most successful and probably the first approach was proposed in 2002 [Ideker et al., 2002], later integrated as a plugin "jActiveModule" in the network analysis and visualization platform Cytoscape [Shannon et al., 2003]. This method aims to find a highly scored connected subnetwork given a network with each node having a score to reflect its activity. Node i is assigned a p-value p_i to indicate the significance of expression changes over

certain conditions, from the mRNA expression profile. P-value is converted to $z-$score by $z_i = \Phi^{-1}(1-p_i)$ for each node, where Φ^{-1} is the inverse of normal cumulative distribution function (CDF). If we assume random variable p_i follow the uniform distribution in $[0, 1]$, then z_i follows the standard normal distribution, which is preferred to measure the node activity.

To find a subnetwork which has high nodes scores, the aggregation of $z-$scores for subnetwork A–the z_A is defined as:

$$z_A = \frac{1}{\sqrt{k}} \sum_{i \in A} z_i \tag{2.1}$$

where k is the number of nodes in subnetwork A. Here the sum score is divided by $\sqrt{k}$ instead of k because of a need to keep the variance of z_A consistent with z_i, but independent of k.[1] Finally, in order to get a subnetwork which has higher aggregation $z-$score compared with a random set of nodes, a corrected subnet score s_A is suggested to use:

$$s_A = \frac{z_A - \mu_k}{\sigma_k} \tag{2.2}$$

where the mean μ_k and standard deviation σ_k are computed based on a Monte Carlo approach, taking several rounds of randomly sampling k nodes from the network.

This combinatorial optimization problem 1 is also called Maximum-Weight Connected Subgraph Problem (MWCSP), which is equivalent to finding a maximum weight clique in a weighted graph, a known NP-complete problem [Karp, 1972; Ideker et al., 2002]. In [Ideker et al., 2002] the searching procedure was expressed as a combinatorial optimization problem and solved by simulated annealing, as Algorithm 1 shows.

The same optimization algorithm was extended to PPI network with edge scores [Guo et al., 2007] or both node and edge scores [Wang and Chen, 2010]. Simulated annealing was also used to search active modules in metabolic network [Bryant et al., 2013], where the metabolites and reactions are connected in a bipartite graph.

[1] $Var(z_A) = Var(\frac{1}{\sqrt{k}} \sum_{i \in A} z_i) = \frac{1}{k} \sum_{i \in A} Var(z_i) = Var(z_i)$

Algorithm 1: Active modules identification using simulated annealing, from [Ideker et al., 2002].

Input: The graph $G = (V, E)$, maximal iterations number N and Temperature T_i which decreases geometrically from T_{start} to T_{end}.

Output: The subgraph G_W module.

Initialization: set each $v \in V$ as active/inactive with probability 0.5;
for *i = 1,2,..N,* **do**
    Randomly pick $v \in V$ and toggle its state;
    Compute the updated score s_i of G_W;
    if $s_i > s_{i-1}$ **then**
        keep v toggled;
    else
        keep v toggled with probability $p = e^{(s_i - s_{i-1})/T_i}$;
end
Output G_W and its highest-scoring component;

Later on, a dozen of heuristic optimization methods were proposed to solve the same or similar problem, including extended simulated annealing [Guo et al., 2007], greedy algorithm [Ulitsky and Shamir, 2007, 2009], graph-based heuristic algorithm [Rajagopalan and Agarwal, 2005] and genetic algorithm (GA) [Klammer et al., 2010; Ma et al., 2011]. These works improve the basic SA from multiple aspects such as introducing edge weights or combining node and edge weights, proposing more efficient operators in EAs or considering graph topology, while the basic question remains unchanged.

2.1.2 Exact approaches

Apart from meta-heuristic algorithms, there is another class of methods to solve the optimization problem. The so-called exact approaches mainly refer to integer linear programming (ILP) has been used to active module identification [Qiu et al., 2008, 2010; Dittrich et al., 2008; Zhao et al., 2008; Backes et al., 2012]. Finding maximum scoring subnetwork with a fixed size is modeled as a constrained maximum-weighted connected graph problem, which is solved by an integrating mixed integer linear programming with breadth-first search strategy. Based on the absolute signal to noise ratio (SNR) $t_i = |\mu_{i1} - \mu_{i2}|/(\sigma_{i1} - \sigma_{i2})$ (where μ_{i1} and σ_{i1} are the mean and standard deviation of gene expression level) to mea-

sure the activity of node i, [Qiu et al., 2008] reformulated finding maximum scoring subnetwork with a fixed size R and a specified root node v_1 as a constrained MWCSP, and solved by integrating mixed integer linear programming (ILP) with breadth-first search strategy.

$$\max_{x_i} S = \frac{1}{R}\sum_{i=1}^{n} t_i x_i$$

Subject to

$$\begin{aligned}
&\sum_{j=1}^{n} c_{1j} = R - 1 \\
&\sum_{j=1}^{n} c_{ji} - \sum_{j\neq 1}^{n} c_{ji} = x_i, \qquad i = 2, ..., n \\
&c_{ij} \leq (R-1)x_i, \qquad i, j = 1, ..., n \\
&x_i \in \{0, 1\}, \qquad i = 1, ..., n
\end{aligned} \tag{2.3}$$

where $x_i = 1$ means node v_i is in the module, c_{ij} are dummy variables representing the flow between selected nodes and $n = |V|$ is the number of nodes in network. The constraints ensure the connectedness of module, while the key parameter need to be determined is the module size R. In practice R is picked from the range $[1, ...K]$ to maximize module score W, where K is the user-defined maximal module size. Equation (2.3) was solved by an existing solver `lp_solve` without relaxation.

A another representative method `heinz` [Dittrich et al., 2008] proposed a new scoring function for modules, which uses an aggregation statistics of P-values to measure the activities of nodes. Based on scoring function, the authors transformed Problem 1 to the following Prize-Collecting Steiner Tree Problem (PCST), and solve it using an existing ILP method [Ljubić et al., 2006]. The optimization solver is from commercial library CPLEX.

Problem 2. PCST. Given a graph $G = (V, E)$, with vertex weight $\mathbf{z}$ for each $v \in V$ and edge weight $\mathbf{c}$ for each $e \in E$, find a connected subgraph $S = (V_S, E_S)$ with maximum weight $f(S) = \sum_{v\in V_S} z_v - \sum_{e\in E_S} c_e$.

We have seen a similar problem on gene regulatory network in [Backes et al., 2012],

where they formulated it as ILP problem and solved it with branch-and-cut-algorithms (B&C-algorithms). The module size is constrained by an explicit equation. The difference is that [Backes et al., 2012] solved on the original directed graph instead of transforming it into a PCST, while both of them called the CPLEX for the optimization part. Besides, [Backes et al., 2012] considered the direction of the graph, as gene regulation relationships are directed. As an extensive research of `heinz`, [Beisser et al., 2012] used integrated analysis combining re-sampling techniques to make the result more robust. The only perturbation in generating consensus members lies at using different data for each single network while keeping the module identification method unchanged.

Both previous methods consider the module score as an aggregation of node score while module can also be extracted from edge information. [Zhao et al., 2008] proposed an integer linear programming (ILP) method for uncovering signal transduction networks (STNs) from PPI network. Given weighted PPI network $G = (V, E, W)$ where $W = [w_{ij}]$ is the edge weight representing either confidence score of the interaction or strong correlation coefficient based on gene expression data.

$$\max_{x_i, y_{ij}} S = -\sum_{i=1}^{|V|}\sum_{j=1}^{|V|} w_{ij}y_{ij} + \lambda \sum_{i=1}^{|V|}\sum_{j=1}^{|V|} y_{ij}$$

Subject to

$$\begin{aligned}
& y_{ij} \leq x_i, \\
& y_{ij} \leq x_j, \\
& \sum_{j=1}^{|V|} y_{ij} \geq 1, \text{if i is starting or ending protein}, \\
& \sum_{j=1}^{|V|} y_{ij} \geq 2x_i, \text{if i is a protein known in STN}, \\
& x_i \in \{0,1\}, i = 1, ..., |V|, \\
& y_{ij} \in \{0,1\}, i, j = 1, ..., |V|,
\end{aligned} \tag{2.4}$$

where $x_i = 1$ means node v_i is in the STN, $y_{ij} = 1$ means edge $E(i, j)$ is in the STN and n

is the number of nodes in network. λ is the user-defined parameter to control the trade-off between STN weights and STN size. Constraints in (2.4) ensure the connectedness of module by excluding the possibilities of isolated nodes or edges. In order to deal with large scale network, [Zhao et al., 2008] relax the constraints from binary variables $x_i \in \{0, 1\}$ and $y_{ij} \in \{0, 1\}$ to continuous variables $x_i \in [0, 1]$ and $y_{ij} \in [0, 1]$. Thus problem (2.4) is converted to linear programming (LP) problem that can be solved efficiently by various tools.

It is natural to use both node and edge information to extract subnetwork if appropriate. [Wang and Xia, 2008] proposed a continuous optimization model to take into account both nodes and edges for weighted for a general biological network with both edge weights $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ and node scores $\mathbf{z} = [z_i] \in \mathbb{R}^n$. The non-negative node score quantifies the association between the node and specific condition, thus maximizing the aggregation module score is supposed to find a subnetwork which both densely connected and condition specific. The optimization model is formulated as a quadratic programming problem:

$$\begin{aligned} \max_{x_i} \; & S = \sum_{i=1}^{n} \sum_{i=1}^{n} w_{ij} x_i x_j + \lambda \sum_{i=1}^{n} x_i z_i \\ \text{Subject to} \quad & \\ & x_1^\beta + x_2^\beta +, ..., + x_n^\beta = 1 \\ & x_i \geq 0, \qquad\qquad\qquad\qquad i, j = 1, ..., n \end{aligned} \tag{2.5}$$

where x_i means the presence of node v_i in the module and n is the number of nodes in network. Parameter λ is used to balance the two terms in objective, and β is the parameter to calculate the vector norm. For instance $\beta = 2$ corresponds to a trust region problem which seeks an optimal solution for quadratic objective in a ball constraints. $\beta = 1$ and modifying the base of x_i to be $|x_i|$ in (2.5) can leads to a sparse solution thus a small module is preferred. After solving problem (2.5) as a continuous non-convex optimization problem, the non-zero entries in solution $\mathbf{x}$ is chosen as final module nodes

indicators.

2.1.3 Recap

[Wu et al., 2009] reviewed the problem of identifying functional modules from the computational perspective, referred two distinct approaches including meta-heuristic algorithms and integer linear programming algorithms and pointed out several limitations. Overall the exact approaches would produce more accurate solutions but are limited from scalability issue. And the implementation of such approaches is not easy; sometimes they depend on solvers provided by commercial software. Meta-heuristic algorithms would give feasible solutions with easy-to-implement programs, but no guarantee with respect to accuracy or convergence is provided. Both approaches provide approximate solutions to the original problem to some extent, the difference can be summarized as: heuristic algorithms **solving** the problem while exact approaches approximately **represent** the problem first. A more brief introduction to several to these algorithms is in Appendix A.

A more comprehensive review of this field can be found in [Mitra et al., 2013], which gives a wider picture about finding modular structure in biological networks.

2.1.4 Evaluation criteria

There are two ways for empirical studies to validate the proposed algorithms: simulated study and real-world data application. In simulated active modules identification, we construct networks with known modular structure and see whether the identified module matches the ground-truth. While in real-world data, we have to evaluation identified modules by biological explanation.

We will describe various data simulation in the following chapters. With ground-truth in hand, we can define the following performance measurement (2.6) for a module identification algorithm. From the topological view, the accuracy of each single module identification is considered as a binary classification problem. The performance is defined

by the 2×2 confusion matrix as Table 2.1 shows: the number of genes correctly detected (True positive, tp), the number of genes in identified module but not in the real module (False positive, fp), the number of genes in the real module but not in the identified module (False negative, fn), and the number of genes neither in identified module or in the real module (True negative, tn). Then the performance can be expressed by (2.6).

Table 2.1: The confusion matrix of single module identification.

	in identified module	not identified module
in true module	True positive	False negative
not in true module (the rest)	False positive	True negative

$$
\begin{aligned}
Recall &= \frac{tp}{tp + fn} \\
Precision &= \frac{tp}{tp + fp} \\
F &= 2 \times \frac{Precision \times Recall}{Recall + Precision}
\end{aligned}
\tag{2.6}
$$

For real world dataset, biological explanation via functional enrichment analysis has become a *de facto* procedure, which is a computational method to determine whether a set of genes has statistically significant similarity with known biological states. Integrative analysis with existing data bridges the gap between algorithmic output gene set and prior knowledge, further gains the biological insight. According to the review by [Huang et al., 2009], enrichment tools can be categorized into the following three classes:

- singular enrichment analysis (SEA)
- gene set enrichment analysis (GSEA)
- modular enrichment analysis (MEA)

The core part is SEA, which is also the most traditional strategy for enrichment analysis. Enrichment analysis on the following several aspects may help to understand biological meaning behind a list of genes (a module).

Gene Ontology. The basic of functional enrichment of a module (gene list) L is to assign the biological process annotations in Gene Ontology [Ashburner et al., 2000] to the

genes (or proteins) in module L. The basic idea is to compute the enrichment P-value, i.e. number of genes in the list that hit a given biology class as compared to pure random chance. Several statistical methods can be applied, including hypergeometric, binomial, chi-square, and Fisher's exact test [Khatri and Drăghici, 2005]. The hypergeometric formulation is directly derived from the problem if given several genes present an enrichment. Suppose the gene list L with length n, and we want to know whether X genes of them are sampled from a specific GO category G with length M. The background population size (e.g. the whole genome) is N. We have the following 2×2 table:

Table 2.2: Illustration of hypergeometric in GO enrichment analysis.

	in category	not in category	Total
class 1 (the candidate)	X	$n - X$	n
class 2 (the rest)	$M - X$	$N - M - n + X$	$N - n$
Total	M	$N - M$	N

According to the hypergeometric distribution, the probability mass function (pmf) of random variable X is given by

$$P(X = k) = \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}, \tag{2.7}$$

where $\binom{N}{n} = \frac{N!}{k!\,(N-k)!}$ is the Binomial coefficient. The result $P(X = k)$ is the probability of $X = k$. In other words, if none of the genes in the candidate list comes from the given GO category, we have to exclude all the possibilities (from 1 to n), then the P-value of this Hypergeometric Test is given by

$$p = 1 - \sum_{k=1}^{n} \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}} \tag{2.8}$$

If the p-value p is very low (less than the alpha level 0.05), we can say the candidate gene list L is significantly enriched by given GO category G.

Bonferroni correction p-value is used when multiple hypotheses are tested. The goal is to maintain the probability of falsely finding any significant hypothesis at the alpha

level [Boyle et al., 2004]. If we have m hypotheses, Bonferroni correction p-value is simply alpha value divided by m. We may also need adjusted p-value by Benjamini-Hochberg (BH) method for correction [Benjamini and Hochberg, 1995].

Pathways. A biological pathway is a series of actions among molecules in a cell that leads to a certain product or a change in a cell [Wikipedia, a]. KEGG Pathway database [Kanehisa and Goto, 2000] is a popular pathway search database highly used by biologists. Now KEGG can also be used as a reference for high-level functions of the cell and the organism [Kanehisa et al., 2015]. The rationale of enrichment analysis keeps the same in pathway enrichment analysis, in which the GO category is replaced with a known pathway. Mainstream tools mentioned above for gene set enrichment analysis all include pathways enrichment options.

Protein complexes. A gene module corresponds to a protein complex which contains multiple proteins interacting with each other. Given modules for human protein complex annotations, we can use BioMart [Smedley et al., 2009] to perform complex queries and filter out useful gene sets.

MiRNA families. MicroRNAs (miRNAs) play key roles in many human pathologies. The FAME algorithm [Ulitsky et al., 2010] can be used on groups of co-expressed genes to identify miRNAs and genomic miRNA clusters with functional importance in specific stages of early human development.

Disease and miRNA associations. DAVID [Huang et al., 2008] and Enrichr [Chen et al., 2013] include disease databases for gene lists. Sometimes it is necessary to figure out if the miRNAs are known to be associated with some disease, as recorded in mir2disease [Jiang et al., 2009]. This is the follow-up steps of miRNA families detection. Being similar to gene ontology, the hypergeometric p-value for the overlap between the detected set and the set of miRNA families associated with the disease is calculated by a similar form of (2.8). The Bioconductor package `DOSE` [Yu et al., 2015] implements disease ontology and enrichment analysis.

Quite a few tools have been developed to conduct GSEA. The Database for Anno-

tation, Visualization and Integrated Discovery (DAVID) [Fresno and Fernández, 2013; Huang et al., 2008] contains a comprehensive set of updated resources and tools to conduct an integrative analysis of gene lists, including functional annotation, link gene-disease associations, highlight protein functional domains and motifs and so on. Similar integrative analysis tools also include Enrichr [Chen et al., 2013], GeneMANIA [Warde-Farley et al., 2010], Metascape [Tripathi et al., 2015], STRING [Szklarczyk et al., 2014], all are web-based and support various annotations. Several of them also provide public API to allow users programmatically access to the resources, which help to deal with massive gene lists.

2.2 Modules detection on gene co-expression network

PPI networks have been chosen as basic network structure for many works, but the protein-protein interactions are known as incomplete and noisy [Wu et al., 2009]. Compared with increasing vast amount of high-throughput data, the speed of constructing reliable and complete PPI is quite slow. Gene co-expression network can be constructed from pervasive expression profiles. A common method to construct co-expression network is to compute the pair-wise correlation values and optionally to filter out significant ones. Several popular similarity metrics used in co-expression network construction have been compared in [Kumari et al., 2012]. In order to construct a reliable network, a minimal number of samples is suggested to be 20 by [Langfelder and Horvath, 2008; Ballouz et al., 2015]. A typical procedure to construct a co-expression network [Li et al., 2011] is shown in Algorithm 2.

2.2.1 Communities detection on gene co-expression network

Gene co-expression network (GCN) has been used as a basic tool in a wide range of applications. After constructing the GCN using Algorithm 2, communities detection or extraction could provide some insights about functional organization. One of the most

Algorithm 2: A typical procedure to construct a co-expression network.

Input: Gene expression profile $X \in \mathbb{R}^{n \times p}$.
Output: Edge list of GCN.

1 **Initialization**: For each pair of gene i and j, commonly by Pearson correlation coefficient $r_{ij} = cor(X_i, X_j)$ where X_i is the i-th column of X ;
2 **Normalization**;
3 Fisher's transformation $z_{ij} = \frac{1}{2} ln(\frac{1+r_{ij}}{1-r_{ij}})$;
4 Standardization $z'_{ij} = \frac{z_{ij}-\mu}{\sigma}$ where μ and σ are the mean and standard deviation of Fisher's score z;
5 Fisher's inverse transformation $r'_{ij} = \frac{exp(2z'_{ij}-1)}{exp(2z'_{ij}+1)}$;
6 **Cutoff (optional)**: Choose threshold τ, only when $r'_{ij} \geq \tau$ keep the edge between i and j. Essential for visualization;

popular communities detection method was proposed by Newman [Newman, 2006], which tries to maximize the modularity criterion over the whole nodes set V:

$$Q = \frac{1}{2m} \sum_{ij \in V} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \sigma(c_i, c_j), \tag{2.9}$$

where a_{ij} is the current connectivity between node i and j, degree $k_i = \sum_j a_{ij}$ and $2m$ is the sum of all degree k_i and $\sigma(c_i, c_j) = 1$ only when node i and j are partitioned into the same group.

Maximizing modularity Q is NP-hard [Brandes et al., 2007] since it is essentially a combinatorial problem. Various algorithms were proposed, either based on modularity maximization or beyond it [Fortunato, 2010]. [Wilkinson and Huberman, 2004] used the community discovery procedure Girvan-Newman algorithm [Girvan and Newman, 2002] to partition a gene co-occurrence network, which is similar to a GCN. The work is one of the first tried to relate the gene components with their functions, using a community discovery way. The community structure has also been explored in plant biology [Aoki et al., 2007], which provides novel insights into the system-level understanding of plant cellular processes. [Liu et al., 2014] proposed a heuristic method to extract modules from a GCN, constructed by low (grade II) and high (GBM) grade glioma expression profiles. [Ruan et al., 2010] reviewed some works on GCN analysis, including basic topological

analysis as well as module discovery in several applications.

2.2.2 Weighted gene co-expression network analysis

Active modules on weighted gene co-expression network (WGCN) is a relatively new topic, but general functional modules detection on WGCN has been studied for a long time. Although modularity (2.9) had been extended to weighted networks [Newman, 2004a], where the degree k_i is defined as the sum of connectivity of node i, the modules detection in WGCN without using a communities detection paradigm has been well-studied.

Zhang and Horvath established a general framework for weighted gene co-expression network analysis [Zhang and Horvath, 2005], and later developed the popular software `WGCNA` [Langfelder and Horvath, 2008]. Being different from selecting part of the entries in full correlation matrix, `WGCNA` keeps all the information, which makes a WGCN as a complete graph. The optimal cut-off threshold in constructing unweighted co-expression network is difficult to determine, and more importantly, throwing away a relatively large proportion of correlation coefficients would lead to information loss. Furthermore, the gene co-expression similarity between gene i and j is expressed as $a_{ij} = |cor(x_i, x_j)|^{\beta}$, where cor is normally chosen as Pearson correlation and power β can enhance the co-expression similarity, which can be picked by scale-free topology criterion [Zhang and Horvath, 2005]. Originated from [Barabási and Albert, 1999] where the distribution of all nodes' connectivity k should follow the power law $p(k) \sim k^{\gamma}$ (this criterion can also be used in unweighted co-expression network construction). In order to capture the relative interconnections between two nodes in the network, `WGCNA` defined the topological overlap matrix (TOM) $\Omega = [\omega_{ij}]$ as the similarity measure.

$$\omega_{ij} = \frac{l_{ij} + a_{ij}}{min(k_i, k_j) + 1 - a_{ij}}, \tag{2.10}$$

where a_{ij} is the similarity between gene i and gene j, and k_i is the connectivity (degree) of gene i, defined as $\sum_j a_{ij}$. l_{ij} Modules detection is accomplished by a hierarchical

clustering on the topological overlap matrix Ω. `WGCNA` implements a dynamic tree cutting method [Langfelder et al., 2008] for choosing an optimal height of the dendrogram.

After getting modules from the co-expression network, `WGCNA` suggests relating modules with phenotypic traits, which contains a quantitative measurement of samples. Specifically, a module eigengene E is defined by the first principal component of given a module. A sample trait is used to determine the gene significance, by correlation of gene or module eigengene E. Such association could point out which module is responsible for a certain phenotypic feature (e.g. body weight). In real world applications, integrative gene set enrichment analysis also provide a guideline to explain the functions and processes associated with the module, denoted by a gene list. For the simulation study, the evaluation of modules detection keeps the same as in section 2.1.4. In addition, some other criteria measure the quality of modules identification such as normalized mutual information (NMI) [Estévez et al., 2009], rand index [Hubert and Arabie, 1985] can be used, since modules detection can be viewed as graph-based node clustering.

Weighted gene co-expression network has contributed to various applications, ranging from brain cancer genes identification [Horvath et al., 2006], Alzheimer disease pathways [Miller et al., 2010], cross-species transcriptional changes [Xue et al., 2013], etc [1]. Gene co-expression network constructed from latest transcriptomics and next-generation sequencing has continuously been proven useful in functional classification and genedisease predictions [van Dam et al., 2017].

2.3 Multilayer networks in computational biology

2.3.1 General multilayer networks analysis

Recent advances in network science have increasingly essential to move beyond simple graph and focus on more complicated but more realistic framework, and one of the rep-

[1]More theory and applied papers about `WGCNA` are at `https://labs.genetics.ucla.edu/horvath/htdocs/CoexpressionNetwork`

resentative models is multilayer network [Kivelä et al., 2014]. To extend existing research on single network layer to multilayer network is necessary since many interconnected systems consist of several layers, or could be better to expressed as multilayer networks. The term "multilayer network" has appeared as "multiplex network", "multislice network" or "multidimensional network" in other places, here we do not distinguish them from the topological perspective. A multilayer network in this thesis refers to a collection of single networks, with each network is a layer, and the interactions between nodes consist of intra-layer and inter-layer links. General concepts in single networks have been extended to multilayer networks, including the measures to characterize local and global properties [Battiston et al., 2014]. These tools and metrics provide basic descriptions of a multilayer system. [De Domenico et al., 2013] proposed a tensorial approach to study general multilayer networks, specialized in dynamical processes. A general form, as well as structural and dynamic properties of multilayer networks, was recently reviewed by [Kivelä et al., 2014].

For the community detection on multilayer networks, the first question needs to be verified is how to define a community in multilayer networks. [Mucha et al., 2010] generalized the modularity criterion [Newman, 2006] for the single network on multislice network and used the same computational heuristics that were available for the single network. [De Domenico et al., 2015] proposed compression of network flows based method, which can reveal smaller modules with more overlap that better capture the real organization on both synthetic data and real network. Both of these two works find the modules from the topological view. [Holme and Saramäki, 2012] discussed different fields about dynamic multilayer networks, in the name of temporal networks, where many concepts in static networks do not necessarily hold. [Hulovatyy et al., 2015] studied the structure and function of temporal networks with dynamic graphlets, a notion to describe statistically significant subgraphs.

Another branch of related research topics that use multiple networks to gain insight in network biology is the network differential analysis, previously reviewed in [Ideker and

Krogan, 2012]. By comparing the difference between the same set of biological components under different conditions can certainly reveal some important change in the network. This paradigm is close to multilayer network modeling but separates the "layers" without links across them, which may lose some temporal or spatial information. Instead, multilayer works provide a unified framework to capture both the relationships intra-layer and inter-layer.

The topics in multilayer biological networks can be categorized into the following several aspects, depending on what consists of each layer. Cross-species multilayer networks and general multilayer co-expression network are discussed here.

2.3.2 Cross-species multilayer networks analysis

In the field of computational biology, the idea of finding modules across multiple species, i.e., the conserved modules in weighted protein-protein interaction networks, has been reported. [Deshpande et al., 2010] proposed a greedy heuristic algorithm `neXus` to identify conserved active modules in two species in parallel. They use the average of activity scores across all genes in the connected module as module score, and take a greedy module growth strategy to find the heavy one. This method can be limited by selecting a starting point or the definition of module score. Thus the result is not guaranteed.

Zinman [Zinman et al., 2015] proposed `ModuleBlast`, a method to search active modules in multiple species and a criterion to distinguish conserved modules and species-specific modules, thus can simultaneous analysis of both conservation and divergence. `ModuleBlast` modified the module score (2.2) in `jActiveModule` [Ideker et al., 2002], and incooprated with edges scores E.

$$S_A = \frac{1}{\sigma_A} \frac{\sum_i (C_i - \beta_A \mu_A)}{\sqrt{k}} + W \frac{\sum_h E_h - \mu_{E(M)}}{\sigma_{E(M)}} \tag{2.11}$$

where the node scores parameters are the same as in (2.2), μ_A and σ_A are mean and SD in the normal distribution, C_i is the score of node i and k is the module size. Similarily,

$\mu_{E(M)}$ and $\sigma_{E(M)}$ are mean and SD of edge scores, W is a user-defined parameter to balance two parts. `ModuleBlast` used a greedy search [Nacu et al., 2007] to find heavy modules, and the starting seeds are selected as highly activated nodes.

The criterion to assess whether a module conserved or species specific is straightforward in `ModuleBlast`: to compute a differentiation score $Diff(S_A)$ for module A and activation score $Active(S_A, X)$ for module A and species X, and see whether $Diff(S_A)$ exceeds a pre-defined threshold value at the same time $Active(S_A, X)$ locate in certain interval. The definition of $Diff(S_A)$ between species X and Y is defined as

$$Diff(S_A) = \frac{\sum_i |C_{i\in X} - C_{i\in Y}|}{\sqrt{k}} \tag{2.12}$$

We can see that `ModuleBlast` takes a lazy strategy to mine conserved modules or species specific modules, which requires several user-defined parameters and threshold values after getting modules, and these values would make an impact on final results.

In constrast, [El-Kebir et al., 2015] proposed `xHeinz` based on previous published approach `Heinz`, which takes conserved degree as objective goal. Both `ModuleBlast` and `xHeinz` chose protein-protein interaction network as infrastructure for human and mouse, combined with omics data to measure the activities of proteins. The problem of identifying conserved active modules from two (connected) species networks [El-Kebir et al., 2015] is formally defined as

Problem 3. Given $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, $w \in \mathbb{R}^{|V_1 \cup V_2|}$, find a subset of nodes $V^\star = V_1^\star \cup V_2^\star$ with $V_1^\star \subseteq V_1$ and $V_2^\star \subseteq V_2$ such that activity score $\sum_{v\in V^\star} w_v$ is maximal and the nodes should be conserved, i.e. $|U^\star| \geq \alpha|V^\star|$ where $|U^\star| := \{u \in V_1^\star \mid \exists v \in V_2^\star : uv \in R\} \cup \{v \in V_2^\star \mid \exists u \in V_1^\star : uv \in R\}$. At the same time the induced subgraphs $G_1[x]$ and $G_2[x]$ are connected.

The trade-off between activity in each species and conservation across species is ex-

pressed as objectives and constraints in the following mathematical model:

$$\max \sum_{i \in V_1 \cup V_2} w_i x_i$$

Subject to

$$\begin{aligned} & m_u = \max_{uv \in R} \{x_u x_v\} && u \in V_1 \\ & m_v = \max_{uv \in R} \{x_u x_v\} && v \in V_2 \\ & \sum_{i \in V_1 \cup V_2} m_i \geq \alpha \sum_{i \in V_1 \cup V_2} x_i \\ & G_1[x], G_2[x] \text{ are connected,} \\ & x_v, m_v \in \{0,1\} && i \in V_1 \cup V_2 \end{aligned} \tag{2.13}$$

where $\mathbf{x} \in \{0,1\}^{|V_1 \cup V_2|}$ represent the nodes in module and $\mathbf{m} \in \{0,1\}^{|V_1 \cup V_2|}$ represent the conserved nodes in module. Parameter α is used to controls the trade-of between conserved part and species specific part. Being similar to `Heinz`, `xHeinz` also uses integer linear programming (ILP) to find the optimal solution of (2.13), with commercial library CPLEX. But the details of how to make sure that $G_1[x]$ and $G_2[x]$ are connected are omitted. The connectivity constraint may cause large extra overhead, since precisely controlling the connectivity requires more operation on status tracking table, such as `jActiveModule` in Cytoscape [Ideker et al., 2002; Shannon et al., 2003]. In theory `xHeinz` can be extended to multilayer network where the number of layers is greater than two, at a cost to increase the computational complexity.

2.3.3 Multilayer co-expression network

Regarding a general multilayer network case, [Li et al., 2011] proposed a tensor-based computational method to mine heavy subgraphs from multiple weighted gene co-expression network, where each shares the same set of genes. A heavy subgraph across many layers can be represented as node membership vector $\mathbf{x} \in \{0,1\}^n$ in which $x_i = 1$ means the i-th node in the recurrent heavy subgraph (RHS). The summed weight of all edges in RHS

can be written into a compact tensor form:

$$H(x,y) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m} a_{ijk}x_i y_j y_k \tag{2.14}$$

where element a_{ijk} represent the edge weight between i and j in the k-th network. $\mathbf{y} \in \{0,1\}^m$ is another the RHS membership vector, and $y_j = 1$ means the j-th network in RHS. Meanwhile, the size of RHS is constrained through vector norms $f(\mathbf{x})$ and $g(\mathbf{y})$. The summed weight maximization is transformed into a continuous optimization problem by relaxing integer constraints $\mathbf{x}$ and $\mathbf{y}$ to continuous constraints:

$$\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}_+^n,\mathbf{y}\in\mathbb{R}_+^m} H(x,y) \\ &\text{Subject to} \\ &f(\mathbf{x}) = 1 \\ &g(\mathbf{y}) = 1 \end{aligned} \tag{2.15}$$

They used the mixed norm $\ell_{0,\infty}(\mathbf{x}) = \alpha\|\mathbf{x}\|_0 + (1-\alpha)\|\mathbf{x}\|_\infty$ $(0 < \alpha < 1)$ to encode the characteristics of gene membership, where the optimal vector should contain equal non-zero values and the rest zero values. In practise $\ell_{0,\infty}$ was approximated by $\ell_{p,2}(\mathbf{x}) = \alpha\|\mathbf{x}\|_p + (1-\alpha)\|\mathbf{x}\|_2$ $(0 < p < 1)$, and they solved the non-convex problem (2.15) by Multi-Stage Convex Relaxation (MSCR) [Zhang, 2010]. The algorithm can be used to obtain multiple RHS with a module extraction strategy, i.e., after getting one module, the algorithm masks the edges in the module with zero weight and call the optimization procedure for problem (2.15) again.

2.4 Chapter summary

This chapter reviews existing works on active modules identification and the applications in multilayer biological networks. According to the literature, we first formally define the

basic form of active module identification on a weighted connected graph. Then we review the existing solutions on the problem into two categories based on the optimization method they used. At the end of this part, we introduce the criteria to evaluate the algorithms in active modules identification on connected graphs, including performance measures for simulated studies and functional enrichment based measures for real world applications. These criteria will be utilized in our proposed methods as well.

As many efforts have been made on identifying modules on protein-protein interactions network, we next review existing studies in gene co-expression networks, which can be modeled as complete graphs. Being different from connected graphs, modules detection on co-expression network is usually accomplished by graph based clustering. Due to the availability of massive gene expression data in various fields, gene co-expression networks analysis has achieved big success in many applications. We will introduce the idea of active modules on weighted gene co-expression network, using the paradigm of "basic network + activity measures", and develop algorithms to identify the modules in Chapter 4.

In order to apply active modules algorithms on multilayer intracellular networks, we need to gain a better understanding of multilayer networks first. Thus we review general multilayer networks analysis, followed by two cases in biological networks. The first example is to find conserved modules in cross-species weighted protein-protein interactions networks, and the second is to mine conserved modules in a general multilayer co-expression network. Based on these two applications, we propose our improved algorithms in Chapter 3 and Chapter 4 respectively, as the case studies of multilayer networks. Another special form of multiple networks comparison is discussed in Chapter 5.

CHAPTER 3

ACTIVE MODULES FOR PROTEIN-PROTEIN INTERACTIONS NETWORK

In the last chapter, we have already introduced active modules identification on protein-protein interactions networks. Meta-heuristic methods and exact approaches were reported before, while each has their limitation. This chapter describes our improvements upon existing research in modules identification, and the proposed method on single and multilayer dynamic PPI networks. Section 3.1 states the importance and our motivation of improvements on active modules identification on the connected graph such as protein-protein interaction networks, which is a well-defined combinatorial optimization problem. Section 3.2 describes the improvements the popular meta-heuristic methods. Section 3.3 proposes a new objective of active modules on PPI via graph partition, and its extension to multilayer PPI networks. And Section 3.4 summarizes the chapter.

3.1 Introduction

As introduced in previous chapters, PPI networks are usually considered as the primary choices in network biology due to the pervasive availability and reliability for many model species. The active modules are fundamental to PPI networks, as the fact that molecules can be related to regulatory and signaling mechanism associating with a given cellular response [Mitra et al., 2013]. And this association with given response utilizes the ac-

tivities of proteins, which is the main difference with modules merely detected from the topological structure of a PPI network.

As a hot topic, there has been intensive research using different intuitions to identify active modules from PPI networks, as Section 2.1 introduced. However, there are still shortcomings in existing methods, which left space for further improvement. These improvements can be summarized from two aspects: the technical view and the conceptual perspective.

From the technical view, there are shortcomings in two widely used classes of active module identification methods: heuristic methods and exact approaches. Generally speaking, heuristic methods could solve the NP Problem 1 at large size and get an acceptable result, but no guarantee of accuracy can be obtained. Furthermore, commonly used techniques such as binary encoding and standard operations in evolutionary algorithms may face problems on a graph problem, which poses specific requirements on the structure of the resulted module. Exact approaches, on the other hand, usually solve an approximation of Problem 1 with integer-linear programming (ILP) techniques [Dittrich et al., 2008], but the computational feasibility drops dramatically as the problem size increases. Other minor flaws for exact approaches lie at implementation, which normally needs professional solvers. This dependency may prevent a large group of potential users, who has no computer science background.

From the conceptual perspective, the definition of active module in Problem 1 only considers vertex activity (or extend to edges part) but ignores the other topological properties such as community structure [Newman, 2003]. The communities derived from the network topology may correspond to functional units. Furthermore, communities can cover a more wide range of all nodes in a PPI network while active modules usually focus on certain subset, which may overlap with connected differentially expressed genes (DEGs). And the number of DEGs is largely determined by the expression data. Gathering the limited number of DEGs as one module is not enough to handle tasks such as protein complexes prediction, where there are quite a few small protein modules to be

mined (see Section 3.3.2).

In this chapter, we try to provide two improvements upon the current active modules identification on PPI networks. The first ensures the connectivity of identified modules by the widely used heuristic methods, which is discussed in Section 3.2. The second describes a new objective of active modules combining active and topological properties of identified modules, which is discussed in Section 3.3. And this work is also extended to multilayer dynamic PPI networks in Section 3.3.2.

3.2 Active modules identification by an improved EA

This combinatorial optimization Problem 1 turns out to be NP-hard [Ideker et al., 2002], which is equivalent to finding a maximum weight clique in a weighted graph, a famous NP-complete problem [Karp, 1972]. As effective tools to solve combinatorial problems, metaheuristic algorithms have been widely applied to search satisfied solutions [Huang et al., 2012; Jia et al., 2012]. The original paper [Ideker et al., 2002] proposed to use simulated annealing (SA), a generic probabilistic metaheuristic to solve this problem. Other heuristic methods include extended simulated annealing [Guo et al., 2007], greedy algorithm [Ulitsky and Shamir, 2007, 2009], graph-based heuristic algorithm [Rajagopalan and Agarwal, 2005] and genetic algorithm (GA) [Klammer et al., 2010; Ma et al., 2011].

Binary encoding is the most common solution representation for active module identification using metaheuristic optimization algorithms such as SA or GA. In this encoding, the module in n-nodes network can be represented by membership vector $\mathbf{x} \in \{0,1\}^n$, where $x_i = 1$ means i-node belongs to the module. One of the prerequisites to use this representation is to ensure the connectedness of the solution, which is not only a biological requirement for resulting subgraphs (connected subgraph means reachable interactions inside the module), but also a computational constraint. Without the connectedness constraint, the maximal objective may correspond to a set of unrelated top-ranked nodes. Unfortunately, most related works mentioned above either did not consider this non-trivial

constraint, or did not tackle this aspect efficiently.

Despite the biologically insightful results obtained from the algorithm in [Ideker et al., 2002] to address Problem 1, the important detail was omitted in the paper: how to ensure the connectedness of the resulting subgraph after applying heuristic operators such as toggling, mutation or crossover. This detail is important because without ensuring the connectedness of a candidate solution, the identification of active modules could be trivial, i.e., a set of isolated top-ranked nodes. The connectivity constraint is also the main factor that increases the computational complexity of exact approaches reviewed in section 2.1.2. Otherwise, the integer linear programming can be applied more efficiently. For evolutionary algorithms (EAs) based on binary encoding, the connectivity constraint makes the result biologically meaningful as well. But the genetic algorithm in [Ma et al., 2011] did not take connectedness of solutions into consideration, which may result in unconnected modules.

In the source code provided by the original authors of jActiveModules, a plug-in for Cytoscape [Shannon et al., 2003], they employed a sophisticated way to check whether toggling one node of a membership vector is feasible, i.e., whether the toggling will affect the connectedness of the candidate solution, which makes the whole algorithm slow. Specifically, given a candidate solution, i.e., a subset of nodes, an additional HashMap has to be maintained to stores the pairwise elements $\{node, comp\}$, which indicates each node and its component (connected subnetwork), respectively, during the whole progress. After toggling, the algorithm will check this HashMap to see whether the operator affects the connectedness of resulted subnetworks. Such operations lead to both running time and memory overhead. How to ensure the connectedness is a non-trivial issue for evolutionary algorithms which take the binary encoding to represent the solution.

Another problem of using generic metaheuristic optimization algorithms is that the search operators, i.e., perturbation [Ideker et al., 2002], mutation and crossover [Ma et al., 2011], are not specifically designed for active module identification, which might result in mediocre search performance in terms of speed and accuracy. In some previous works, it

has shown that by incorporating local search operators into generic metaheuristic optimization algorithms, one can significantly improve the speed and accuracy of community detection in large scale biological networks [Liu et al., 2014; He et al., 2016].

The connectivity of identified modules also involves another issue, the module size, which is important to biological interpretation. Neither too small nor too large modules are desired, since small modules may provide limited information to individual genes while too big modules can contain too many functions. The properly sized module should also be connected.

In this section, we propose a novel active module identification algorithm based on a memetic algorithm. Specifically, a simple encoding/decoding scheme is conducted to ensure the connectedness of the identified active modules. Based on the scheme, we also design and incorporate a local search operator into the memetic algorithm to improve its performance.

3.2.1 Algorithms description

Decoding. We propose a simple but fast binary decoding scheme, which does not require the HashMap nor explicit operations when add or remove current nodes. Our binary encoding scheme is the same as used in [Ma et al., 2011], i.e., a binary vector of n binary values of which each represents the membership of the node ($x_i = 1$ means i-node belongs to the module). The key difference is the decoding scheme. But the previous work [Ma et al., 2011] did not consider the connectedness constraint. Specifically, we conduct the connected components finding (CCF) operation on the binary vector presented subset, and then extract the connected subnetworks. According to the Erdős-Rényi model [Erdos and Rényi, 1959], a random graph $G(n, p)$, where n is the number of vertices and p is the probability of the presence of edge between each pair of nodes, is expected to have a giant connected component containing certain fraction of the vertices, when $np > 1$. In other words, if we select a random set of vertices from a graph and ensure there are enough edges among the vertices at the same time, we may find a giant connected component from them

with high probability. Note that in the induced graph $G' = (V', E')$, $np = 2|E'|/(|V'|-1)$, which means the average number of edges should be more than half of the nodes. In the mainstream PPI databases such as BioGRID [Chatr-Aryamontri et al., 2015] and STRING [Szklarczyk et al., 2014], the average number of edges is larger than expected. Thus we can use various optimization techniques to get a larger set of nodes which are not necessarily connected, then find the connected component. The advantage is to avoid taking connectivity into consideration when identifying the maximal scored subgraph, which normally introduces high computational overhead.

Decoding scheme based on CCF operation is described in Algorithm 3, where Breadth-first search (BFS) is used to recursively find the node's neighbors.

Algorithm 3: Connected components finding based decoding algorithm

Input: A vector $\mathbf{x} \in \{0, 1\}^n$, where n is number of nodes in network.
Output: The list of components.

```
CCF: Connected components finding on x;
for each x_i == 1 in x do
    if node i is not visited then
        Include node i in current component;
        Component number increased;
        Breadth-first search (BFS) on node i;
    end
end
Output The list of components;
```

Since there are multiple connected subgraphs in a candidate solution, the fitness calculation can be flexible. In the simplest case, we can use the subgraph with the highest aggregated node score. However, no matter how we calculate the fitness function, genetic or meta-heuristics algorithms can be directly applied based on the encoding/decoding scheme. If we use SA, in each iteration, we decide to add or remove a randomly picked node by the same criterion: if toggling the state of the selected node c can increase s_A of the subnetwork A with the highest aggregated node score, then we choose to toggle it; otherwise to toggle it with certain probability p. Compared with the original mechanism of jActiveModules in Cytoscape, this decoding is computationally tractable and easy to

implement.

The connected components finding Algorithm 3 is actually based on breadth-first search (BFS) on a (sub)graph, requiring time complexity $O(|V'| + |E'|)$ where $|V'|$ and $|E'|$ are the number of nodes and edges of the current set respectively. Notice that this time complexity is only equivalent to one case to toggle a node in jActiveModules in theory.

Optimization. The evolutionary algorithm (EA) is an iterative algorithm that can be used in solving combinatorial optimization problems. Inspired by biological evolution, a typical EA uses operators such as selection, crossover, and mutation to improve the candidate solutions [Golberg, 1989]. Parameters for an EA are number of iterations T, population size P, crossover probability p_c and mutation probability p_m.

Memetic algorithm (MA) improved standard EA by enabling individuals to perform local refinements [Moscato et al., 1989]. Numerous effective local search (LS) methods have been developed and incorporated into MA to obtain state-of-the-art results in various applications [Ishibuchi et al., 2003; Zhu et al., 2007; Tang et al., 2009]. A recent review of MA can be found in [Neri and Cotta, 2012]. Algorithm 4 describes a common framework of MA, where the standard mutation operation is replaced by a local search operator. Being similar to conventional GA algorithms which partially prevent the "local optimum" problem by mutation and crossover mechanisms, Algorithm 4 uses an enhanced mutation step.

Algorithm 4: General framework of MA

```
1 Initialization: randomly initialize the population;
2 while not satisfied the stopping condition do
3 |   Evolutionary operations;
4 |   for each individual in population do
5 |   |   Perform local search with probability p_LS;
6 |   end
7 end
```

According to our encoding/decoding scheme, each candidate solution consists of several connected subgraphs, we define the highest score of these subgraphs as the fitness

of $\mathbf{x}$, denoted by $F(\mathbf{x})$. For multiple modules identification, we use a module extraction mechanism, i.e. to identify one active module each time and then extract it from the background network, which is left for next round. For the local search part, here we mainly consider a simple greedy search strategy. We pick all individuals in the population with probability p_{LS} and conduct M times of toggle on the current individual where $M < N$. Finally, we replace each chosen individual with the best scored one, followed by other genetic operators. More operations as in [Zhu et al., 2007] to conduct local search could be applied here.

It is necessary to make sure the identified module has reasonable **size** when toggling nodes. Both extreme small and large module can make the interpretation difficult. But the objective of module score itself cannot prevent large modules. Neither the original work [Ideker et al., 2002] nor the GA based method [Ma et al., 2011] proposed mechanisms to achieve reasonably sized modules. Furthermore, to maximize objective module score may lead to single gene module or very large component in practice. As long as one large module (e.g. containing 1,000 genes) is connected and has a high aggregated score, then this module may be found using general Algorithm 4.

Here we make a simple modification to the mutation operator in GA and local search operator in MA to constrain the module size to be desired: as long as the number of candidate genes (number of '1's in encoding vector) exceeds some threshold N_{max}, there will be no more potential nodes added to the subset. On the contrary, if the module size is going to be smaller than the predefined threshold N_{min}, there will be no more potential nodes removed out from the current subset.

The procedure of local search is described as in Algorithm 5. The whole procedure of MA for active module identification is combining Algorithm 4 and the local search strategy. For evolutionary operations in the whole procedure, we chose the commonly used one-point crossover.

Complexity. The computational complexity for memetic Algorithm 4 is $O(TP)$ without local refinements, where T is the number of ierations and P is the population

Algorithm 5: Local search for MA on active module identification

```
Procedure of local search in Algorithm 4;
for each individual in population do
    Select current individual x with probability p_LS;
    x_best = x;
    for i = 1→ M do
        Generate individual x' by toggle a random position j on x_best though the
         following procedure;
        if x_best_j == 1 and ∑x_best > N_min then
            x' = x_best by x_best_j = 0;
        end
        else if x_best_j == 0 and ∑x_best < N_max then
            x' = x_best by x_best_j = 1;
        end
        Conducting Algorithm 3 on x' and calculating the module score F(x');
        if F(x') > F(x_best) then
            x_best = x';
        end
    end
end
```

size. The expected computational complexity of Algorithm 4 with greedy search is thus $O(TP+TM(|V'|+|E'|))$ where M is the number of local search trails, $|V'|$ and $|E'|$ are the number of nodes and edges of a candidate solution subgraph respectively. If we consider almost half of the whole nodes may get involved in evolution and normally the number of edges $|E'|$ in subgraph approximately at the same level of the number of nodes $|V'|$, the simplified complexity of the whole algorithm should be $O(TP + TMN)$. Generally, the size of population P is small compared with the network size N, which makes the latter dominate the running time. And the number of local search trails M in each inner iteration also has an impact on the efficiency. In theory, the sophisticated mechanism of jActiveModule can also be used here, but it would make the fitness evaluation more difficult. And the space requirement is higher due to the HashMap.

3.2.2 Empirical evaluation

Module connectedness validation

First of all, we validate if the modules identified by proposed algorithm are connected. The baseline algorithm is a simple GA with basic binary encoding scheme without connectedness guarantee to search highly scored module in molecular networks. We use a simulated interaction network with 500 nodes and 1000 edges, to just validate the connectedness property. Figure 3.1 showed the resulted module, and the red nodes are in the subset of resulted module and gray ones are their neighbors but not included in the subset. We can see that the original subset is not connected at nodes like 185, 400 and 163 etc, which are isolated from a large set of red nodes. If we use the same GA algorithm with the proposed encoding mechanism in 3.2.1, we can get a different result as Figure 3.2 shows. With the same input and algorithmic parameters, the red nodes are now connected in the identified active module. The standard GA (modified from COSINE [Ma et al., 2011]) and visualization code are available at `https://github.com/fairmiracle/EAModules`.

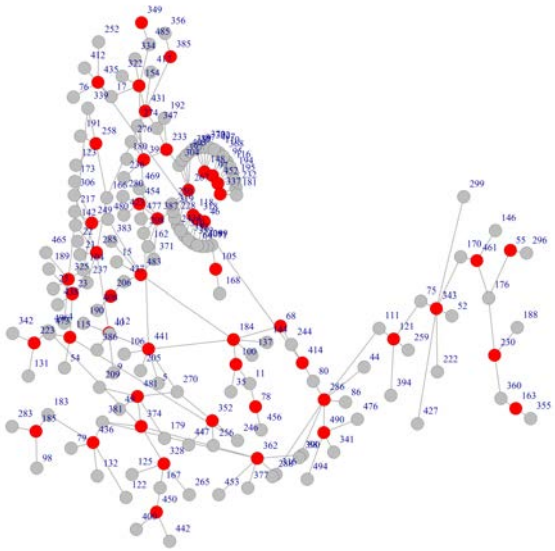


Figure 3.1: Modules identified by GA on simulated data. The red nodes are not connected though they are supposed to be.

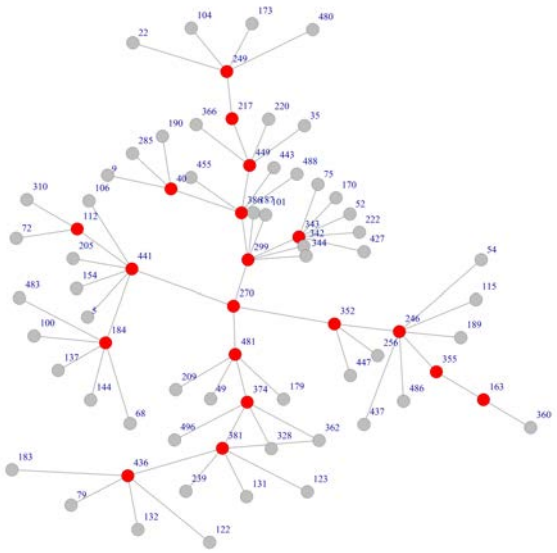


Figure 3.2: Modules identified by modified GA with proposed decoding scheme on the same simulated data as in Figure 3.1. The red nodes are connected.

Yeast PPI network

We first validate the proposed algorithm on a small real protein-protein network with 329 proteins in Yeast [Ideker et al., 2001]. The p-values on each node show the significance of gene expression changes in response to a single perturbation: a strain with a complete deletion of the GAL80 gene versus wild type yeast. The network structure data and expression values are available from Cytoscape sample data. The constructed network has 329 nodes and 358 edges. And the goal is to find a top-scoring subnetwork which shows a significant response to the perturbation.

We compare the performance of three algorithms using the encoding method in section 3.2.1: simulated annealing (SA), genetic algorithm (GA) and the proposed memetic algorithm (MA). In order to compare SA with other two EAs fairly, we run SA P (also the population size in GA and MA) times and select the best result, since SA is viewed as a single population GA. The number of iterations T for all algorithm is 10000, and temperatures decrease from 1.0 to 0.01 for SA. Other evolutionary parameters are crossover rate $p_c = 0.9$ for GA and MA, single point mutation rate $p_m = 0.9$ for GA and local search iterations $M = 10$ for MA. In GA and MA, we also reserve the best individual in each

iteration for stability. The actual fitness evaluations determined by these parameters are summarized in Table 3.1. We run each algorithm 50 times with randomly initialization and then compare the performance w.r.t highest module score and corresponding module size.

Table 3.1: A comparison between three algorithms based on the number of fitness evaluations.

Algorithm	Fitness per iteration	Iterations	Fitness in total
SA	1	1E+06	1E+06
GA	100	10000	1E+06
MA	1000	1000	1E+06

We compare the rate of convergence of three algorithms, to see how objective value improves along with iterations. We define the best objective value in population as the indicator in each iteration. According to Figure 3.3, MA reaches the stable objective earlier than GA. The local search scheme could make sure the performance of MA is no worse than basic GA, and the monotonic selection leads to early convergence compared with GA, at the cost of longer running time of the local search. Both GA and MA get higher objective than SA, which needs much more iterations to reach a high score.

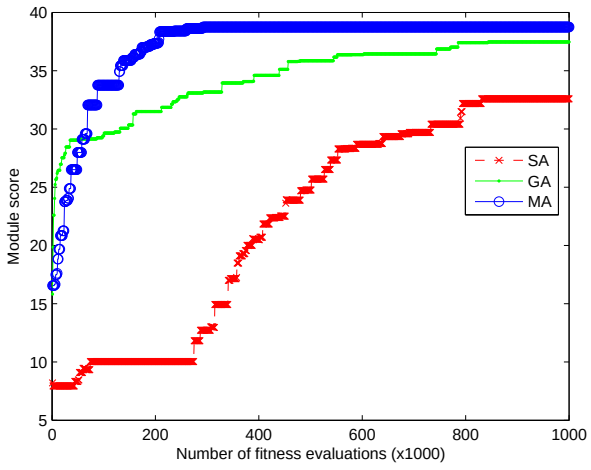


Figure 3.3: Convergence rate comparison of three algorithms: SA, GA and MA from one trail. Technically, we record the objective value every iteration of all three algorithms, but from Table 3.1, the total iterations are different. We sample the objective every 1000 fitness evaluations for three algorithms. MA is the first to reach the stable status and also the highest module score.

Human PPI network

In order to check the biological relevance of identified modules by proposed algorithm, we apply it to the real world protein-protein interactions (PPI) network. The background PPI network for *homo sapiens* is obtained from two updated databases: BioGRID [Chatr-Aryamontri et al., 2015] Release 3.4.138 and STRING v10.0 [Szklarczyk et al., 2014], specifically 9606.protein.links.v10.txt. The BioGRID for *homo sapiens* has 362,775 interactions while STRING stores 8,548,002 protein pairs, with a combined score ranging from 150 to 999 for each link. The gene expression profile comes from GEO35103 controlled by the differentiation of the Th17 cell, which is considered to play a key role in the pathogenesis of autoimmune and inflammatory diseases [Tuomela et al., 2012]. The expression profile contains 48,000 probes (genes), and 28,870 were kept after the following process: 1) remove probes those do not have gene symbols; 2) remove probes with more than 20% of missing values or NAs; 3) replace the rest of missing data with mean value of the row they belong to. Further, we select 5003 significantly expressed genes from all of them using limma [Smyth, 2005]. The gene filtering algorithm selects some potentially important candidates and reduces network size. Finally, we select PPI pairs according to match of expression probes.

For BioGRID we simply match the gene names for each probe of expression profile. But STRING uses the protein name (start with ENSP), thus we need to match that with official symbols (like ARF5) with database Ensembl Genes 84 [Flicek et al., 2014], and select the corresponded genes. The genes selection and construction procedure of PPI network from multiple data sources are shown in Figure 3.4 and related code are available at `https://github.com/fairmiracle/PPINet`.

The network constructed from BioGRID has 2,327 nodes and STRING has 1,602 nodes, with 1,480 nodes in common. Figure 3.5 and 3.6 show the largest connected component from both networks, and red nodes are the shared genes.

We execute Algorithm 4 on both networks, and use a module extraction method to identify multiple modules from this network, i.e. to identify one active module each

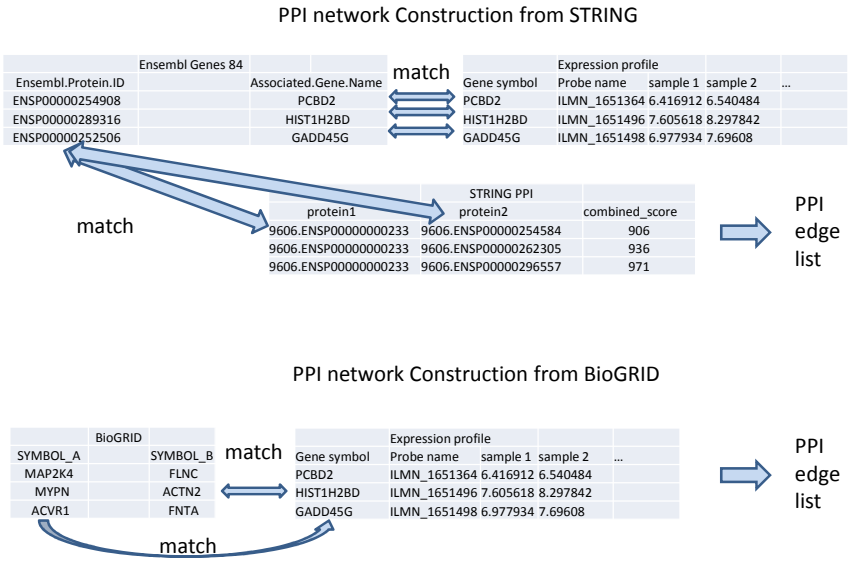


Figure 3.4: The construction process of PPI network from two updated databases STRING and BioGRID.

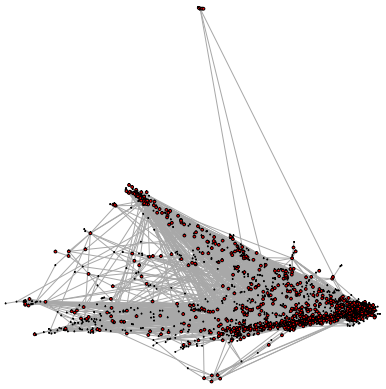


Figure 3.5: The largest connected component of PPI network from BioGRID

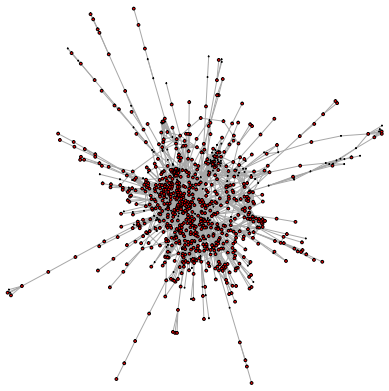


Figure 3.6: The largest connected component of PPI network from STRING

time and then extract it from the background network, which is left for next round. The largest size of each module is 100. The full gene symbols lists of modules are provided in supplementary materials (at `https://github.com/fairmiracle/EAModules/tree/master/examples/Supplementary`, where "GSE35103FromString_MA.txt" means the modules identified from STRING based PPI network using MA algorithm, and each module is stored as plain text by module score, gene ids, and official gene symbols). We can also see that under the same condition, MA could achieve higher scored modules than GA.

Generally speaking, larger module tends to be enriched by multiple biological functions, which may not be very relevant to each other. The first module identified from STRING PPI network contains 76 genes and according to GeneMANIA [Warde-Farley et al., 2010], among all potential links inside the module, there are 51.63% co-expression links, 33.59% are physical interactions and 4.16% are pathways. The top biological processes and pathways related to this module are listed in Table 3.2. We can see several

general responses found by STRING, and the hub nodes in this module shown as in Figure 3.7 also indicate general important genes related to receptor signaling and signal transduction (also see `http://bit.ly/2a87HTB`). While functions given by GeneMANIA show that these functions are intensively involved in Th17 cell differentiation. Several items are also claimed in a recent publication [Brummelman et al., 2016], which is consistent with the experimental settings.

Table 3.2: Enrichment analysis result of the first module identified by Algorithm 4 at 2h time point.

Biological Process (GO) given by STRING			
pathway ID	pathway description	count	FDR
GO.0007166	cell surface receptor signaling pathway	39	1.85E-17
GO.0007165	signal transduction	52	1.35E-16
GO.0044700	single organism signaling	51	1.11E-14
GO.0007154	cell communication	51	2.36E-14
GO.0051716	cellular response to stimulus	54	5.15E-14
KEGG pathway given by STRING			
5166	HTLV-I infection	10	6.45E-06
4630	Jak-STAT signaling pathway	8	1.22E-05
4380	Osteoclast differentiation	7	2.95E-05
5202	Transcriptional misregulation in cancer	7	0.000154
04151	PI3K-Akt signaling pathway	9	0.000194
Functions given by GeneMANIA			
Index	Function	FDR	Coverage
1	T cell differentiation	5.63e-12	13/90
2	lymphocyte differentiation	5.63e-12	15/144
3	leukocyte differentiation	6.95e-12	17/226
4	positive regulation of leukocyte activation	1.87e-11	15/166
5	positive regulation of cell activation	2.55e-11	15/172
6	regulation of leukocyte activation	1.01e-10	16/232
7	T cell activation	1.57e-10	16/241

The smaller module tends to play more specific roles in the process. Figure 3.8 plotted by GeneMANIA [Warde-Farley et al., 2010] shows the interactions between these 17 genes, and 87.84% of them are co-expression links according to previous studies. The function is more about pathways, like Fc-epsilon receptor signaling and Fc receptor signaling.

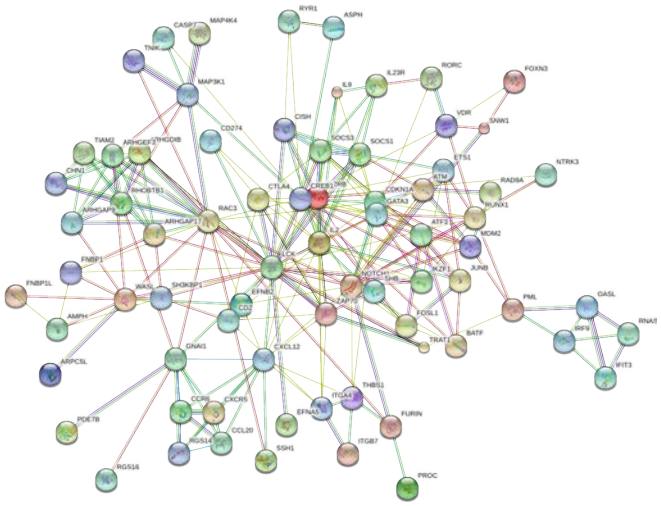


Figure 3.7: The first identified module plotted by STRING, where edges represent both known interactions including curated databases and experimentally determined and predicted interactions such as gene neighborhood and gene co-occurrence. Colored nodes standard for query proteins and first shell of interactors, and white nodes for second shell of interactors.

Related genes contained in this module are MAP3K1, MAP3K5 and MAP3K6, mitogen-activated protein kinase kinase, which play central roles in the regulation of cell survival and differentiation. The connection between MAP3k and Th17 differentiation is supported by [Suddason and Gallagher, 2016], through encoding MEKK1 which controls both B and T cell proliferation. And MEKK1 regulates Cdkn1b expression in Th17 cells. Other processes enriched by the module are also mentioned in a recent study [Cleret-Buhot et al., 2015].

Different sources of protein-protein interactions also make an impact. From the comparison between modules between BioGRID and STRING networks, we can see that they share some functions such as Fc-epsilon receptor signaling pathway, but they are not totally the same. Interactions in BioGRID largely rely on high-throughput datasets and previous studies, which makes the identified module less focused on some functions. Irreverent supporting materials make the set of genes has lower coverage and higher FDR, given by functional enrichment report by GeneMANIA. In contrast, STRING has many experimental and predicted interactions [Szklarczyk et al., 2014], and the combined score of links can further help to pick more reliable edges of PPI network. Identified mod-

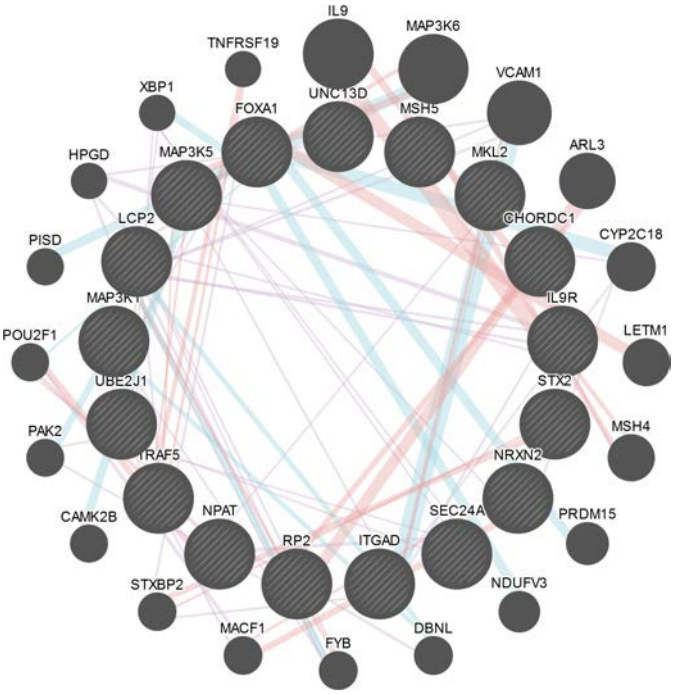


Figure 3.8: The relatively small module plotted by GeneMANIA, where most of the edges are co-expression links according to previous studies.

ules from this network tend to have more significant biological meanings. Take the first module (`http://bit.ly/2asI0Nw`) for example, gene ontology tells the hierarchical biological process of this module by starting with the regulation of tyrosine phosphorylation of the Stat3 protein. The Stat3 has been shown to be a master regulator of Th17 cell differentiation [Wei et al., 2007] and related immune pathways.

3.3 Active modules identification via convex optimization

The previous section aims to identify active modules based on the original definition of Problem 1, but improve the current heuristic methods from the computational perspective. However, like many existing works, node [Dittrich et al., 2008] or edge weights [Ulitsky and Shamir, 2009] (or both [Ma et al., 2011]) weight too much in the active module identification objective, which leads to modules without topological properties other than connectivity.

Some earlier results derived from the pure topological structure, such as communities [Girvan and Newman, 2002] or motifs [Milo et al., 2002], also make sense in terms of biological meaning [Palla et al., 2005]. The communities are fundamental important since they may correspond to functional units [Newman, 2013]. Modular structure in various PPI networks has been reported in [Spirin and Mirny, 2003; Chen and Yuan, 2006; Luo et al., 2006; Pizzuti and Rombo, 2014], and community detection techniques [Girvan and Newman, 2002; Fortunato, 2010] have been applied in a wide range of biological networks, including PPI networks, to find such topological modules. As a more recent example of 2016 Disease Module Identification DREAM Challenge [Organizers, 2016], participators were asked only to use network structural information to find disease modules, which were evaluated by pre-defined genome-wide association study (GWAS) sets. Quite a few teams only adopted basic communities detection methods which maximize the modularity [Newman, 2004b; Blondel et al., 2008] also achieved good performance.

Though the topology of a biological network does not always precisely reflect the function or even disease-determined regions [Barabási et al., 2011], which are the real concerns in biology. The topological modules and functional modules may have some overlapped components, but still vary on constitution. To better address these concerns, we can combine the **active modules** which show striking changes in molecular activity or phenotypic signatures that are associated with a given cellular response in biological networks [Mitra et al., 2013], and the **topological modules** which have obvious vertices or edges features such as dense interactions within groups, together to extract the targeted subnetworks.

In this section, we consider a new objective of representing an active module which is composed of two parts: the topological part and the active part. By combining these two we can reveal functional change and regulatory or signaling mechanisms. The topological property of a module is derived from graph partitioning, and the active property is highlighted by the higher expected average node score. As a result, we formulate the active modules identification on connected graphs as a constrained convex quadratic program-

ming problem, which can be solved efficiently by iterative methods. We also show that this method can be extended to multilayer networks, by conducting the same algorithm on an aggregation of multiple layers.

3.3.1 Optimization on single-layer connected graph

We aim to extract a module from a weighted network with both topological and active features. The problem is formally defined as:

Problem 4. Given a weighted graph $G = (V, E, W, \mathbf{z})$ where w_{ij} is the weight between vertex i and j, vertices weights z_i for each $i \in V$, find a connected subnetwork $S = (V_S, E_S)$ of G with significant separation from the rest in both edges interactions and vertices weight.

The separation from topological perspective has been intensively studied in graph theory. We thus start with the logic of spectral clustering (partitioning) on the graph (network) G. The *cut* for two disjoint subsets A and its complement $\overline{A}$ on G, defined as

$$cut(A, \overline{A}) = \sum_{i \in A, j \in \overline{A}} w_{ij}. \tag{3.1}$$

The basic criteria to measure the quality of a partition on graph G is to minimize $cut(A, \overline{A})$ over all possibilities of A. But there exists a trivial case when $A = V$ or $A = \emptyset$. Even when there is only one node or very few nodes in A, the $cut(A, \overline{A})$ is small, which is not preferred. Several modifications based on *cut* were proposed, such as the *ratio cut* [Wei and Cheng, 1989]:

$$R = \frac{cut(A, \overline{A})}{|A||\overline{A}|}. \tag{3.2}$$

[Zhao et al., 2011] proposed the community extraction framework based on *ratio cut*, incorporating with the assumption that the interactions in the extracted module should be denser. The same intuition had been formulated as the well-known *modularity* [Newman, 2004b], which also measures the quality of a partition instead of certain extraction module.

Adopting the actual-minus-expected paradigm on the average node score, we define the following objective to be minimized:

$$f = \frac{\sum_{i\in A, j\in\overline{A}} w_{ij}}{|A||\overline{A}|} - \lambda\sqrt{|A||\overline{A}|}\left(\frac{\sum_{i\in A} z_i}{|A|} - \frac{\sum_{i\in\overline{A}} z_i}{|\overline{A}|}\right). \tag{3.3}$$

The first part is the same as *ratio cut*, which is supposed to be minimized from the topological perspective. And $(\sum_{i\in A} z_i/|A| - \sum_{i\in\overline{A}} z_i/|\overline{A}|)$ in the second term is the gap between the expected node score in the module and the rest, which is supposed to be maximized (equivalent to minimize the negative) from the active perspective. We also add the penalty $\sqrt{|A||\overline{A}|}$ to this part, when $|A| = |\overline{A}|$ the penalty has a maximal value, which is consistent with the denominator of *ratio cut*. Parameter λ is the used to balance the consideration between edge part and node part. Without prior knowledge or preference on the two parts, we suggest the default value $\lambda = 1$.

Being similar to [Von Luxburg, 2007] (section 5, Graph cut point of view), we define the vector $\mathbf{x} = (x_1, x_2, ..., x_n)^\top$ where $n = |V|$, with entries:

$$x_i = \begin{cases} \sqrt{|\overline{A}|/|A|} & \text{if } i \in A \\ -\sqrt{|A|/|\overline{A}|} & \text{if } i \in \overline{A}. \end{cases} \tag{3.4}$$

Note the following two facts which are the same as in [Von Luxburg, 2007]:

$$\mathbf{x}^\top\mathbf{1} = \sum_i x_i = \sum_{i\in A}\sqrt{|\overline{A}|/|A|} - \sum_{i\in\overline{A}}\sqrt{|A|/|\overline{A}|} = 0. \tag{3.5}$$

and

$$\mathbf{x}^\top\mathbf{x} = \sum_i x_i^2 = \sum_{i\in A}(|\overline{A}|/|A|) + \sum_{i\in\overline{A}}(|A|/|\overline{A}|) = |V|. \tag{3.6}$$

The graph Laplacian $L = D - W$, and D is the diagonal degree matrix, in which

$D_{ii} = d_i = \sum_j w_{ij}$. Now we have

$$\begin{aligned}
&\mathbf{x}^\top L\mathbf{x} = \textstyle\sum_{i,j}^{n} w_{ij}(x_i - x_j)^2 \\
&= \textstyle\sum_{i\in A, j\in\overline{A}} w_{ij}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 + \sum_{i\in\overline{A}, j\in A} w_{ij}\left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \\
&= 2cut(A,\overline{A})\left(\tfrac{|\overline{A}|}{|A|} + \tfrac{|A|}{|\overline{A}|} + 2\right) = 2cut(A,\overline{A})\left(\tfrac{|\overline{A}|+|A|}{|A|} + \tfrac{|A|+|\overline{A}|}{|\overline{A}|}\right) \\
&= 2|V|R
\end{aligned} \tag{3.7}$$

where R is defined in (3.2) and

$$\begin{aligned}
&\mathbf{x}^\top\mathbf{z} = \textstyle\sum_i x_i z_i = \sum_{i\in A} x_i z_i + \sum_{i\in\overline{A}} x_i z_i \\
&= \textstyle\sum_{i\in A} z_i\sqrt{|\overline{A}|/|A|} - \sum_{i\in A} z_i\sqrt{|A|/|\overline{A}|} \\
&= \sqrt{|A||\overline{A}|}\left(\tfrac{\sum_{i\in A} z_i}{|A|} - \tfrac{\sum_{i\in\overline{A}} z_i}{|\overline{A}|}\right)
\end{aligned} \tag{3.8}$$

Combining equation (3.7), (3.8), and conditions (3.5) and (3.6), we can rewrite the objective (3.3) into the matrix form:

$$\text{minimize } f = \frac{\mathbf{x}^\top L\mathbf{x}}{2n} - \lambda\mathbf{z}^\top\mathbf{x} \quad \text{subject to} \quad \mathbf{x}^\top\mathbf{1} = 0,\ \mathbf{x}^\top\mathbf{x} = n. \tag{3.9}$$

Since the objective function is smooth and differentiable, we use a projected gradient [Lin, 2007] approach to find the local optimum, which works as follow. The gradient is used to determine the primary direction, followed by orthogonalization to satisfy the condition (3.5) and normalization step to satisfy the condition (3.6). The orthogonalization and normalization steps can be viewed as a projection operation $P(\cdot)$ on the candidate solution, which is the linear combination of previous solution and the gradient:

$$\mathbf{x}^{(k+1)} = P[\mathbf{x}^{(k)} - \alpha^{(k)}\nabla f(\mathbf{x}^{(k)})] \tag{3.10}$$

where $\alpha^{(k)}$ is the step-size in the k-th iteration, which is supposed to make the objective

to satisfy the non-monotone criterion, where σ is a pre-defined constant:

$$f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \leq \sigma \nabla f(\mathbf{x}^{(k)})^{\top}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}). \tag{3.11}$$

We want to pick the largest $\alpha^{(k)}$ to satisfy (3.11) in general, which needs $\alpha^{(k)}$ to scale efficiently. Here we directly adopt the following procedure [Lin, 2007]:

Algorithm 6: Searching proper step size for gradient based method.

```
Given scale factor 0 < β < 1, 0 < σ < 1, maximal (inner) iteration number N.;
if α^(k) satisfies (3.11) then
    repeatedly (maximal N iterations) increase it by α^(k) = α^(k)/β;
    until α^(k) does not satisfy (3.11) or x(α^(k)/β) = x(α^(k));
end
else
    repeatedly (maximal N iterations) decrease it by α^(k) = α^(k) · β;
    until α^(k) does satisfies (3.11);
end
```

Following [Calamai and Moré, 1987] we claim that any limit point of sequence $\{\mathbf{x}^{(k)}\}$ generated by the projected gradient procedure (3.10) is a stationary point of (3.3). According to the Proposition 1 of [Von Luxburg, 2007], the Laplacian L is positive semi-definite, the problem (3.9) is convex thus a local optimum is also the global optimum. The detailed convergence analysis is in Appendix B.

After getting the target set A, it may come to the case that the candidates nodes are not connected, due to the relaxation on $\mathbf{x}$, which was supposed to be a 0-1 vector. We conduct a connected components finding (CCF) operation on A as in [Li et al., 2017a] (described as in Algorithm 3 in section 3.2.1) to extract large component. The existence of giant component in a random set of vertices from a graph is supported by the Erdős-Rényi model [Erdos and Rényi, 1959]. In summary, the procedure is formally described as Algorithm 7.

For a large network, we may need to execute Algorithm 7 multiple times, to get proper number of modules. Every time we get the target set A, any connected component other than the giant component can be considered as a module, as long as the size is in proper

Algorithm 7: Module extraction via convex optimization

Input: The adjacency matrix W, node score vector $\mathbf{z}$, parameter λ, maximal candidate set size N_{max}, step-size $\alpha^{(1)}$, maximal iteration T, tolerate τ

Output: The extracted module node set A.

Laplacian: Compute diagonal degree matrix D and Laplacian $L = D - W$;
for *k=1,2,...,T* **do**
  $\alpha^{(k)}$ is picked by a non-monotone line search Algorithm 6;
  Gradient descent: $\mathbf{y} = \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x}^{(k)})$;
  Orthogonalization: $\mathbf{z} = \mathbf{1}^\top \mathbf{1} \cdot \mathbf{y} - \mathbf{y}^\top \mathbf{1} \cdot \mathbf{1}$;
  Normalization: $\mathbf{x}^{(k+1)} = \sqrt{n} \cdot \mathbf{z} / \|\mathbf{z}\|_2$;
  if $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|_2 < \tau$ **then**
    break;
  end
end
Determine the primary partition: $i \in A$ if $x_i > 0$ otherwise $i \in \overline{A}$;
if $|A| > N_{max}$ **then**
  Further division: Recursively call the algorithm;
end
else
  CCF: Conduct connected component finding (Algorithm 3) on A if it is not connected;
  Output the module vertices set.
end

range.

3.3.2 Aggregation on multilayer PPI network

The community structure has been explored on multilayer connected graphs. Various algorithms have been proposed to detect the communities on multilayer graphs, see the review [Kivelä et al., 2014]. One of the natural ideas is to extend the concept of modularity to multilayer case, as [Mucha et al., 2010] did. Taking the layer into consideration, they transformed the matrix calculation in original modularity into the tensor computation. The underlying assumption of this extension is that each layer shares a similar structure, and the word "multiscale" in the title of [Mucha et al., 2010] also indicates that each network shares a similar structure.

However, in some real applications, the multiple layers can be diverse. The sub-challenge 2 of Disease Module Identification DREAM Challenge [Organizers, 2016] required finding modules across multiple networks, including two protein-protein interaction networks, one signaling network, one co-expression network, one cancer network and one homology network. In such a multilayer network system, the size and topology are diverse with all nodes aligned. Due to the fact that there are very few shared edges among different networks, even no single edge shared by all networks, we can highlight the majority of the presence of edges by adopting the idea of **aggregated graph** or **consensus graph**. An aggregated graph is simply defined by the sum of all edge weight matrices, followed by proper cut-off to remove low valued entries. The intuition behind this aggregating is to enhance the concurrent edges across more layers and remove layer-specific edges. As a result, the aggregated graph encodes the conservation information of multiple layers. And the module (defined by the solution of (3.3)) extracted from this graph represents conserved subnetwork with maximal node activities across multiple layers. When each layer is a snapshot of a dynamic system, the extracted modules can be viewed as time-invariant subnetworks, which may be related to persistent biological processes.

The complete procedure to mine modules from multiple networks $G_1, G_2, ..., G_n$ is

describe as Algorithm 8.

Algorithm 8: Module extraction via convex optimization on multilayer network

Input: The adjacency matrices $W_1, W2, ..., W_n$, node score vectors $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n$ of n layers, edge weight cut-off threshold τ_1 and node weight cut-off threshold τ_2

Output: The extracted module node set A.

1 **Normalization**: Make sure $W_1, W_2, ..., W_n$ and $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n$ are normalized respectively;

2 **Aggregation**: $W = \sum_{i=1}^{n} W_i$, $\mathbf{z} = \sum_{i=1}^{n} \mathbf{z}_i$;

3 **Cut-off**: $W(W < \tau_1) = 0$, $\mathbf{z}(\mathbf{z} < \tau_2) = 0$;

4 **Module extraction**: Call Algorithm 7;

5 Output the module vertices set;

Why aggregation works. Here we still begin with two communities, one is the target, and the other is the background. Assume the real community structure that characterizes the separation between the target and the rest exists, in a graph G^*. The corresponding adjacency matrix is W^*. Now the problem is whether we can approximate G^* by the consensus graph, or approximate W^* by $1/L \sum_{i=1}^{L} W_i$.

We use the Frobenius norm (3.12) to measure the gap between two matrices:

$$\begin{aligned} \|A - B\|_F &= \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij} - b_{ij}|^2} \\ &= \sqrt{\|A^{(1)} - B^{(1)}\|^2 + \|A^{(2)} - B^{(2)}\|^2 + ...\|A^{(n)} - B^{(n)}\|^2} \end{aligned} \tag{3.12}$$

where the matrix $A, B \in \mathbb{R}^{m \times n}$, and $A^{(i)}$ is the i-th column of A.

Then the gap between G^* and the consensus graph can be expressed by:

$$\begin{aligned} g_1 &= \|\frac{1}{L} \sum_{k=1}^{L} W_k - W^*\|_F^2 \\ &= \sum_{i=1}^{n} \left(\frac{1}{L^2} \sum_{k=1}^{L} W_k^{(i)\top} W_k^{(i)} + \frac{2}{L^2} \sum_{k \neq j}^{L} W_k^{(i)\top} W_j^{(i)} + W^{*(i)\top} W^{*(i)} - \frac{2}{L} \sum_{k=1}^{L} W_k^{(i)\top} W^{*(i)} \right) \end{aligned} \tag{3.13}$$

Assume each layer encodes part of W^*, the expected gap between W^* and W_i can be

expressed by:

$$
\begin{aligned}
g_2 &= \frac{1}{L}\sum_{k=1}^{L}\|W_k - W^*\|_F^2 \\
&= \sum_{i=1}^{n}\left(\frac{1}{L}\sum_{k=1}^{L}W_k^{(i)\top}W_k^{(i)} + W^{*(i)\top}W^{*(i)} - \frac{2}{L}\sum_{k=1}^{L}W_k^{(i)\top}W^{*(i)}\right)
\end{aligned}
\tag{3.14}
$$

We can see that

$$
\begin{aligned}
& g_1 \leq g_2 \\
\iff & \frac{1}{L^2}\sum_{k=1}^{L}W_k^{(i)\top}W_k^{(i)} + \frac{2}{L^2}\sum_{k\neq j}^{L}W_k^{(i)\top}W_j^{(i)} \leq \frac{1}{L}\sum_{k=1}^{L}W_k^{(i)\top}W_k^{(i)} \\
\iff & 2\sum_{k\neq j}^{L}W_k^{(i)\top}W_j^{(i)} \leq (L-1)\sum_{k=1}^{L}W_k^{(i)\top}W_k^{(i)} \\
\iff & 0 \leq (L-2)\sum_{k=1}^{L}\|W_k^{(i)}\|^2 + \|\sum_{k=1}^{L}W_k^{(i)}\|^2
\end{aligned}
\tag{3.15}
$$

which means the aggregation reduces the gap to the optimal adjacency matrix W^*. Of course, the Frobenius norm only considers the summed edge weights. Precisely measuring the gap between the modular structure of two networks needs theories in network alignment [Elmsallati et al., 2016], which is challenging.

3.3.3 Empirical evaluation

Toy example

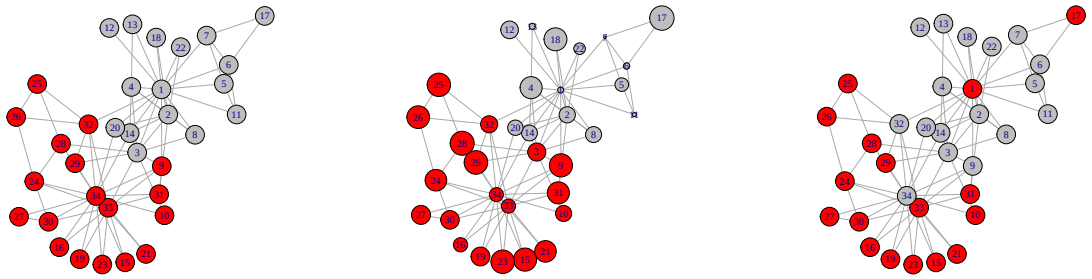
We first compare the proposed method with *ratio cut* minimization, which actually solves the graph partitioning without using the node score:

$$
\text{minimize } f = \frac{\mathbf{x}^\top L\mathbf{x}}{2|V|} \quad \text{subject to} \quad \mathbf{x}^\top\mathbf{1} = 0,\ \mathbf{x}^\top\mathbf{x} = n. \tag{3.16}
$$

According to Rayleigh-Ritz theorem [Von Luxburg, 2007; Lutkepohl, 1997], the solution is directly given by the eigenvector corresponding to the second smallest eigenvalue

of L, denoted as $\mathbf{u}$. The nodes of the extracted module are selected according to the sign of entries in $\mathbf{u}$, i.e. $i \in A$ if $u_i > 0$. We denote this method as the **spectral algorithm** in the following discussion.

We compare the spectral algorithm with Algorithm 7 on the widely used Zachary karate club network [Granovetter, 1973], as Figure 3.10 shows. Red nodes in three figures mean the members in the target module. We can see that spectral algorithm (see Figure 3.9b) not only failed to find the important node '34' in the target module but also wrongly include two nodes '1' and '17' from the opposite module. Furthermore, node '1' is considered to play a core role in the opposite module. The proposed method (see Figure 3.9b) successfully found all the nodes but introduced additional node '3', which also interacts with other nodes in the target module. In Figure 3.9b the node size indicates the assigned score, and the result validates the purpose of (3.9), which combines modular structure and high scored nodes. In other words, not all high scored nodes are included, such as the connected nodes '20', '1', '13' and '22'.



(a) The ground truth. (b) Partition by (3.9). (c) Partition by (3.16).

Figure 3.9: Comparison of the ground truth with partition by spectral algorithm (3.16) and proposed method (3.9), on the widely used Zachary karate club network. Red nodes mean the members in the target module. The node size in the middle figure means node score.

Results on Single-layer network

In order to check the biological relevance of identified modules by the proposed algorithm, we apply it to the real world protein-protein interactions (PPI) network. The human PPI network and p-values derived from differential expression and survival analysis [Rosenwald et al., 2002] come from the package `BioNet` [Beisser et al., 2010], which implements the integer programming based active module identification algorithm [Dittrich et al., 2008] as well as a heuristic algorithm. The heuristic algorithm can be viewed as an alternative of the Cytoscape plugin `jActiveModules` [Ideker et al., 2002]. Being different from the objective (3.3), both the exact approach [Dittrich et al., 2008] and the heuristics method [Ideker et al., 2002] aim to collect as more high-scored and connected nodes, but the exact approach relies on external library CPLEX.

We compare the heuristic method from `BioNet` and Algorithm 7 (using default parameters: $\lambda = 1$, $\alpha = 0.01$, $\tau = 1E - 9$, $T = 100$, $N_{max} = 100$) on the same PPI network with 2559 nodes and 7788 edges, and the nodes are scored by the Beta-Uniform-Mixture (BUM) model [Dittrich et al., 2008] based on expression p-values. `BioNet` identifies a module with 37 nodes and 44 edges, and the Algorithm 7 identifies a module with 54 nodes and 54 edges. And there are 20 nodes in common, most of which are high-scored nodes. In order to show the difference directly, we plot both modules in one figure, with different colors, as Figure 3.10 shows. We can see that in both modules, there are some low-scored nodes such as SMAD2, LYN, CDC2, NME1, serving as bridge nodes which connect server component. But the module identified by Algorithm 7 includes more low-scored nodes and their interactions.

The identified modules are enriched by biological processes and pathways related to the experimental settings. Since these two modules have different size, we use the false discovery rate (FDR) of enriched KEGG pathways to show the significance of enrichment analysis. Table 3.3 shows the top 5 pathways enriched by both modules, and we can see find more significant results. Furthermore, all pathways are related to cancer, which is consistent with the data source [Rosenwald et al., 2002].

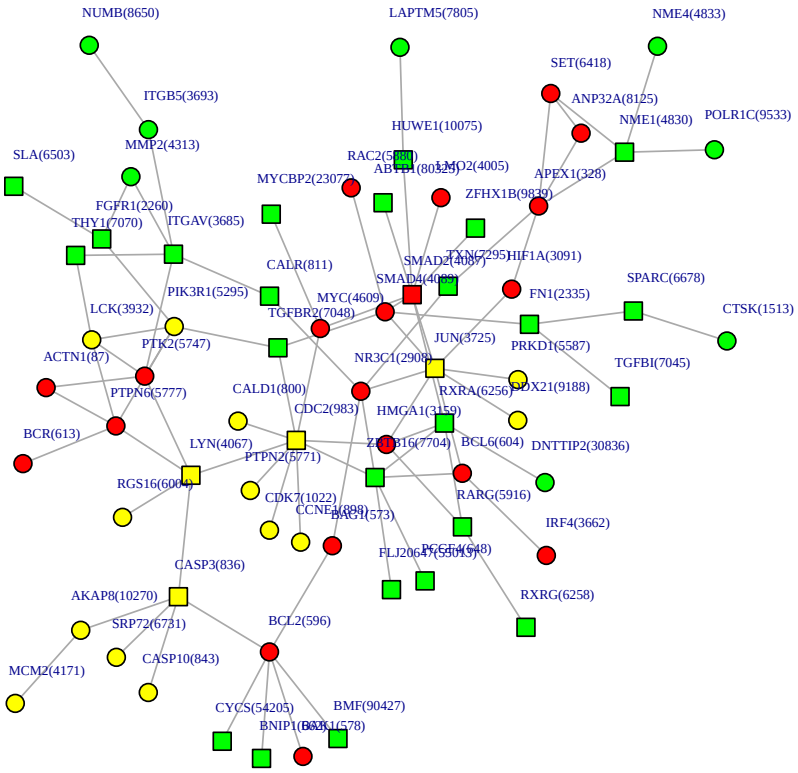


Figure 3.10: The identified modules by `BioNet` and Algorithm 7. The red nodes are shared by both methods, yellow for `BioNet` and green for Algorithm 7. The shape of nodes indicates score, squares indicate negative scores and circles for positive. Algorithm 7 is able to connect more low-scored regions which may have close relationship with the biological mechanisms.

Results on multilayer network

Multilayer dynamic PPI networks (DPPI or DPIN) are usually constructed by integrating static PPI network and a time-course gene expression data [Ou-Yang et al., 2014; Tang et al., 2011; Wang et al., 2013]. Each layer in a DPPI network represent a specific PPI of a time point, and the whole DPPI is supposed to model the dynamic properties of protein interactions. While the active modules across all layers can be viewed as complementary to those identified from static PPI [Chen et al., 2014], due to the time course data integration.

Network construction. Here we adopt the DPPI construction method by [Wang et al., 2013; Ou-Yang et al., 2014], where the topological structure of each layer is determined by both the static PPI network $G = (V, E)$ and the gene expression data $X \in \mathbb{R}^{N \times T}$, where $N = |V|$ is the number of proteins and T is the number of time points. The dynamic parts for time points $1 \leq t \leq T$ are derived from G and X. Specifically, a protein i is considered active at time point t, if the expression value exceeds the threshold $AT(i)$, defined in [Wang et al., 2013]:

$$AT(i) = \mu(i) + 3\sigma(i)(1 - F(i)), \tag{3.17}$$

where $\mu(i)$ and $\sigma(i)$ are the mean and standard deviation of protein i across T time points. $F(i) = 1/(1 + \sigma^2(i))$ is the weight factor. The necessary condition for protein i is connected with protein j at time point t is thus twofold: i and j are connected in static PPI network G, i.e. $e_{ij} \in E$ and i and j are both active at time t, i.e. $X_{it} \geq AT(i)$

Table 3.3: The top 5 enriched KEGG pathways of the modules identified by Algorithm 1, and heuristics algorithm which targets at a group of high-scored and connected nodes, on single human PPI network.

Algorithm 7			Heuristics algorithm		
ID	Description	FDR	ID	Description	FDR
5200	Pathways in cancer	5.05E-16	5200	Pathways in cancer	1.23E-9
5222	Small cell lung cancer	5.36E-9	5210	Colorectal cancer	1.23E-9
5210	Colorectal cancer	1.03E-8	5161	Hepatitis B	4.77E-7
5205	Proteoglycans in cancer	1.18E-8	4110	Cell cycle	5.43E-6
5202	Transcriptional misregulation in cancer	1.72E-8	4520	Adherens junction	9.32E-6

and $X_{jt} \geq AT(j)$. In addition, the node activity of each layer simply measured by the expression values.

The processed gene expression data, as well as DPPI construction, can be found in [Ou-Yang et al., 2014], where we have 2389 proteins in total and the expression data at 12 time points. The DPPI thus has 12 layers, proteins in each are scored by the expression level. Then we apply Algorithm 8 on this DPPI and set default parameters $\tau_1 = 6$ and $\tau_2 = 0$.

Results interpretation. First of all, we try to demonstrate the advantage of multilayer aggregation by comparing the result from the consensus graph and each single layer. We evaluate the result by f-measure [Li et al., 2010], which assesses the similarity between identified modules (protein complexes) $C_i \in P$ and reference complexes $C_j \in B$. We claim C_i and C_j match if $v_{ij} \geq 0.25$ [Wang et al., 2013]:

$$v_{ij} = \frac{|C_i \cup C_j|^2}{|C_i \cap C_j|}. \tag{3.18}$$

F-measure is defined based on precision and recall:

$$\begin{aligned} \text{precision} &= \frac{|\{C_i \mid C_i \in P \wedge \exists C_j \in B,\ C_j \text{ matches } C_i\}|}{|P|} \\ \text{recall} &= \frac{|\{C_j \mid C_j \in B \wedge \exists C_i \in P,\ C_i \text{ matches } C_j\}|}{|B|} \\ \text{F} - \text{measure} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \end{aligned} \tag{3.19}$$

Two widely used reference complexes CYC2008 [Pu et al., 2008] and MIPS [Mewes et al., 2004] are chosen as gold standard sets B. Furthermore, we use SPICi [Jiang and Singh, 2010] as baseline algorithm which was designed for detecting non-overlapping complexes from static PPI. Also, note algorithms for detecting overlapping complexes such as ClusterONE [Nepusz et al., 2012] and TS-OCD [Ou-Yang et al., 2014] would achieve higher F-scores since part of proteins might frequently appear in different modules, which increases the chance to be captured by reference sets according to (3.19).

Figure 3.11 shows the overall result of modules from each layer, consensus graph and the baseline algorithm SPICi [Jiang and Singh, 2010]. The purpose of this figure is twofold: identifying modules on the consensus graph is superior to each single one with the same algorithm and Algorithm 8 could achieve comparable result with other methods in terms of accuracy. Specifically, Algorithm 8 tends to achieve higher precision but lower recall compared with SPICi [Jiang and Singh, 2010]. A possible reason for failing to recall as more complexes lies at the CCF operation in Algorithm 7, which may miss some proteins in the reference set.

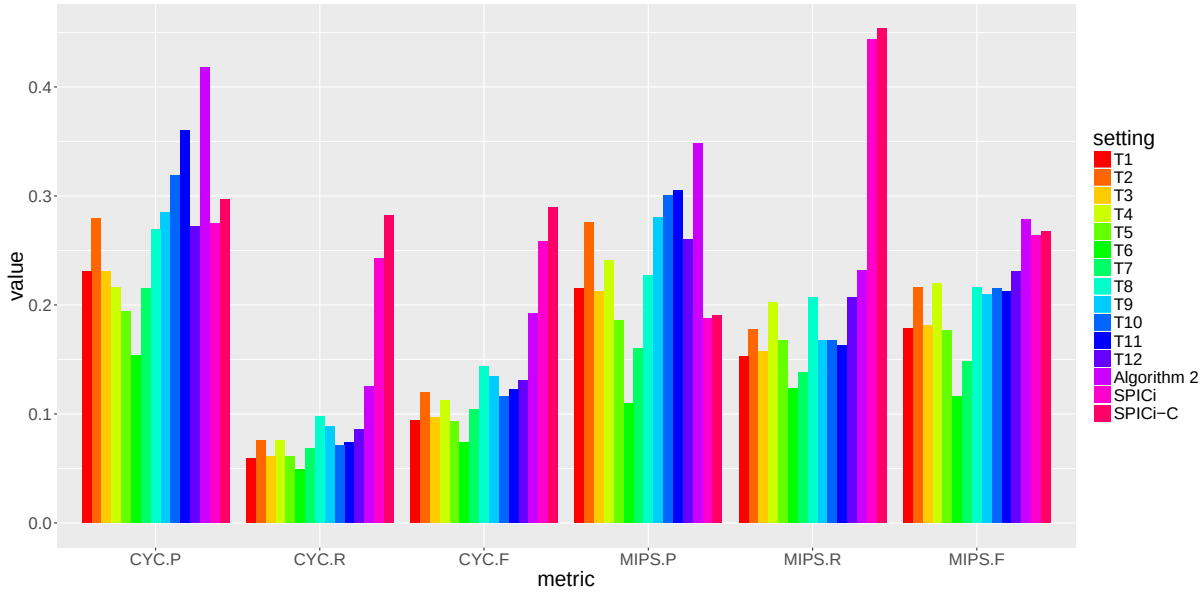


Figure 3.11: The F-measure of identified complexes from each time point layer, consensus graph and the baseline algorithm. 'CYC-P' means the precision evaluated by CYC2008 [Pu et al., 2008] and 'MIPS-R' means the recall evaluated by MIPS [Mewes et al., 2004]. 'F' is the F-measure defined as in (3.19). 'SPICi-C' is SPICi [Jiang and Singh, 2010] on the consensus graph.

In order to validate the biological meaning of the identified complexes, we conduct enrichment analysis with STRING [Szklarczyk et al., 2014], which provides interfaces to access the annotation resources as well as enrichment result. Furthermore, we compare the results of Algorithm 7 and SPICi [Jiang and Singh, 2010] in a statistical way, and count the number of complexes which are significantly enriched by at least one GO term (biological processes and KEGG pathways) at a given FDR ($\leq$0.05) cutoff. From Table 3.4 we can see that Algorithm 7 found relatively larger complexes on average, and the

resulted complexes are enriched by more KEGG pathways.

Table 3.4: Overview of the resulted complexes of Algorithm 7 and SPICi [Jiang and Singh, 2010]. *#Items* means the total GO terms of all complexes and *Ration* is proportion of complexes which are significantly enriched by at least one GO term at a given FDR ($\leq$0.05).

Method	Modules	Avg. size	Enrichment		
				BP	KEGG
SPICi	298	4.21	#Items	8482	599
			Ratio	0.81	0.72
Algorithm 7	277	5.06	#Items	7114	1088
			Ratio	0.81	0.88

3.4 Chapter summary

This chapter studies active modules identification on PPI networks and proposes two novel algorithms.

The first addresses the connectivity issue of resulted module based on the popular heuristic methods. Connectivity insurance is vital for both computation and biological explanation. We introduce a simple strategy based on a connected components finding (CCF) procedure on the binary encoding of module membership, thus connectivity is guaranteed. Based on encoding/decoding scheme, we propose a memetic algorithmic framework embedded with local search operators. Empirical studies on real networks show the effectiveness and efficiency of this strategy.

The second suggests a new objective of active modules combining topological properties and active part of a identified module, which is solved by a convex optimization approach. We also generalize the proposed method to the multilayer dynamic PPI networks, using an aggregation of multiple single layers. A straightforward argument for the aggregation operation is given. Experiments on real-world data show the advantages of the optimization approach as well as the aggregation of multilayer networks.

CHAPTER 4

ACTIVE MODULES FOR GENE CO-EXPRESSION NETWORK

This chapter focuses active modules identification on the weighted gene co-expression network (WGCN). Section 4.1 introduces the basic WGCN analysis and states the motivation of active modules on WGCNs. Section 4.2 describes the active modules identification method on single layer WGCN. And Section 4.3 extends the single layer case to the multilayer cases, including two-layer cross-species and multilayer dynamic WGCN. After the main work Section 4.4 discusses several related issues around the proposed method. Finally, Section 4.5 summarizes the chapter.

4.1 Introduction

We have already seen active module identification algorithms have been developed to integrate omics data and protein-protein interaction (PPI) networks or metabolic networks which are constructed from prior knowledge databases [Ideker et al., 2002; Chuang et al., 2007; Qiu et al., 2010; Ma et al., 2011; Dittrich et al., 2008; Chen et al., 2017].

However, most of the existing active module identification algorithms, including our own work [Li et al., 2017a,b] described in the previous Chapter can only work with protein-protein interaction (PPI) or metabolic networks. These networks are constructed from prior knowledge databases, which might not be comprehensive and accurate. Moreover,

for some non-model species or some new model species such as Daphnia, their PPI or metabolic networks are not available, which limited the application of active module identification algorithms.

In contrast, gene co-expression network is a pure data-driven gene network, which only relies on gene expression profile. From a gene expression profile of N genes, a $N \times N$ similarity matrix is calculated, in which each element measures how similar the expression level of a pair of genes change together. Usually, the similarity matrix is converted into an unweighted adjacency matrix by replacing the elements which are above a certain threshold (i.e. significant co-expressions) with 1 and replacing the remaining elements with 0. However, choosing an appropriate threshold is difficult. In this paper, we will focus on the weighted gene co-expression network (WGCN), which is a fully connected graph constructed directly from the similarity matrix.

Module identification from WGCNs is the first but crucial step for gene co-expression network analysis. Most traditional module detection methods for gene co-expression networks were based on gene clustering, i.e., grouping similar genes based on their correlations or edge weights into clusters as modules [Zhang and Horvath, 2005]. These identified co-expression modules are considered to participate in some biological process [Zhang and Horvath, 2005], and those with significant biological meaning are regarded as functional modules. However, because the clustering based module identification methods cover all genes without considering their activity, the identified modules might not be informative to reveal the dynamic mechanisms associating with a given cellular response. It is suggested to associate the modules identified by `WGCNA` with phenotypic traits [Langfelder and Horvath, 2008], which possibly include gene activities. However, since the module identified by WGCNA contain a large number of genes, it is not easy to associate them with their gene expression levels.

We hypothesize that by identifying active modules which consider gene activities in WGCNs, we will be able to reveal not only the dynamic biological processes but also the regulatory signaling mechanisms underlying a given cellular response. However, rigorous

definition of **active modules** in WGCNs has not been proposed. Our key criteria to define an active modules in WGCN is that it should be significantly different from random subnetworks at two perspectives: 1) From the topological point of view, the nodes in the active module should be densely connected with each other, i.e., significantly co-expressed, which is quantified by the module score based on edge weights. 2) From the regulatory and signaling mechanism point of view, an active module should show a significant change in molecular activity which can be measured by the module score based on the activities, i.e., expression levels of the genes (node scores).

In this chapter, we develop the first active module identification algorithm `AMOUNTAIN` for WGCNs. The aim of this algorithm is to identify active modules to reveal not only the dynamic biological processes but also the regulatory and signaling mechanisms underlying a given cellular response. To this end, we propose a new definition of the active module in a WGCN. Based on the definition, we formally formulate active modules identification problem in single-layer WCGNs and generalize the problem to multilayer WCGNs.

We evaluate the proposed framework on both simulated data and real-world data, including multiple species and time-course gene expression datasets. The results indicate that the identified active modules can reveal not only the dynamic biological processes but also the regulatory and signaling mechanisms that underlie a given cellular response. We provide `AMOUNTAIN` as a Bioconductor package which requires minimal dependencies.

4.2 Continuous optimization on single network

In order to identify active modules from WGCNs, which are essentially weighted and fully connected graphs, we need to modify the definition of active modules in problem (1). Our idea is to consider the node scores of the genes as the measures their activities under certain conditions as same as in problem (1), while we also consider the topology or co-expression relationship among those genes as indicated by their edge weights. More specifically, we aim to find an active module or subgraph of size k (otherwise it corresponds

to a trivial case containing all top-scored nodes) that have both maximal aggregated node score and maximum aggregated edge weight, which can be formally defined as:

Problem 5. Given a complete graph $G = (V, E)$, with vertex weight $\mathbf{z}_v \in \mathbb{R}$ for each $v \in V$ and non-negative edge weights $W = [w_{ij}]$ for each edge (i, j), find a subgraph T of size k with large vertices weight $\sum_{i \in T} z_i$ and also edges weights $\sum_{i,j \in T} w_{ij}$.

Problem 5 is essentially a simplified problem of (K_1, K_2)-Recurrent Heavy Subgraph (RHS) problem [Li et al., 2011] but with additional node scores. (K_1, K_2)-RHS problem considers multilayer co-expression networks, which is also discussed in detail in the next section. A module can be represented by a membership vector $\mathbf{x} \in \{0, 1\}^n$, where $x_i = 1$ means the i-gene belongs to the module. Thus the optimization is naturally expressed as:

$$\begin{aligned} &\max_{\mathbf{x}} \; S = \mathbf{x}^\top W \mathbf{x} + \lambda \mathbf{z}^\top \mathbf{x} \\ &\text{Subject to} \\ &\qquad \sum_{i=1}^{n} x_i = k \\ &\qquad x_i \in \{0, 1\}, \qquad\qquad i = 1, ..., n, \end{aligned} \tag{4.1}$$

where parameter λ controls the trade-off between edges score and nodes score.

4.2.1 Continuous optimization formulation

The NP-hardness of equation (4.1) can be proved by reducing it to the well-known k-clique problem (See Appendix C.1), which is NP-complete. To solve this NP-hard problem, similar to [Dittrich et al., 2008], one might apply linear relaxation and then use integer programming methods. However, the time complexity is not guaranteed, especially for large-scale networks. Alternatively, if we relax the integer constraints of $\mathbf{x}$ to continuous constraints [Wang and Xia, 2008; Li et al., 2011] and control the module size by introducing a vector norms of $\mathbf{x}$, specifically, in solution $\mathbf{x} \in \mathbb{R}^n_+$ when $x_i > 0$ means the i-th node is in the module, it becomes a nonnegative and equality constrained quadratic

programming (QP) problem (4.2), which can be solved by various existing continuous optimization techniques in polynomial time.

$$\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}_+^n} F(\mathbf{x}) = \mathbf{x}^\top W\mathbf{x} + \lambda\mathbf{z}^\top\mathbf{x} \\ &\text{Subject to,} \\ &f(\mathbf{x}) = 1, \end{aligned} \tag{4.2}$$

where $f(\mathbf{x})$ is the vector norm. The ℓ_p-norm ($p > 0$) of $\mathbf{x}$ is defined as $(\sum_i |x_i|^p)^{1/p}$.

The choice of vector norm affects the structure of the solution (4.2). For example, the ℓ_0-norm and ℓ_1-norm can produce a sparse solution which corresponds to modules with small size. This is desirable since we aim to identify smaller modules which are easy to verify in the follow-up experiments. Since the optimization of ℓ_0-norm is also a NP-hard combinatorial problem, the ℓ_1-norm has been widely used as an alternative [Donoho, 2006]. However, ℓ_1-norm tends to produce a too sparse solution which is again not desired.

The elastic net penalty [Zou and Hastie, 2005], which is a linear combination of ℓ_1 and ℓ_2, i.e. $\alpha\|\mathbf{x}\|_1 + (1-\alpha)\|\mathbf{x}\|_2^2$, has been introduced. In the context of the least square problem with elastic net penalty, $\alpha = 1$ corresponds to lasso [Tibshirani, 1996] and $\alpha = 0$ corresponds to ridge regression. Therefore, the elastic net is considered to enjoy the advantages of both lasso and ridge regression, i.e. the sparsity and accuracy, by tuning the parameter α.

In `AMOUNTAIN`, we employ the elastic net penalty [Zou and Hastie, 2005] to control the sparsity and improve the efficiency of our module identification algorithm. Therefore, the equation (4.2) becomes

$$\begin{aligned} &\max_{\mathbf{x}\in\mathbb{R}_+^n} F(\mathbf{x}) = \mathbf{x}^\top W\mathbf{x} + \lambda\mathbf{z}^\top\mathbf{x} \\ &\text{Subject to,} \\ &f(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + (1-\alpha)\|\mathbf{x}\|^2 = 1. \end{aligned} \tag{4.3}$$

Optimization method

We use a projected gradient method to solve (4.3) since the objective function is smooth and differentiable and the constraint, i.e., elastic net, is strictly convex. In addition to gradient ascend to find the local maximum, the projected gradient method employs projection operation to project the current candidate solution to the nearest point in the convex feasible region [Lin, 2007; Gong et al., 2011]. The projected gradient method is guaranteed to converge to the stationary points of the problem (4.3) [Calamai and Moré, 1987]. Specifically, We used the following sequence to approximate the final solution:

$$\mathbf{x}^{(k+1)} = P_C(\mathbf{g}), \tag{4.4}$$

where $\beta^{(k)}$ is the step size which can be fixed or tunned to improve the convergence rate [Lin, 2007]. P_C is the Euclidean projection of a vector $\mathbf{g} = \mathbf{x}^{(k)} + \beta^{(k)}\nabla F(\mathbf{x}^{(k)})$ on the convex set C, and the subproblem is thus defined as:

$$P_C(\mathbf{g}) = \arg\min_{\mathbf{x}\in\mathbb{R}^n_+} \frac{1}{2}\|\mathbf{x}-\mathbf{g}\|_2^2 \; s.t. \; \alpha\|\mathbf{x}\|_1 + (1-\alpha)\|\mathbf{x}\|_2^2 = 1. \tag{4.5}$$

Solving subproblem (4.5) involves a root finding procedure [Gong et al., 2011] which can be done in linear time, and the details can be found in Appendix C.2. The Euclidean projection based optimization for problem (4.3) is summarized as Algorithm 9.

Algorithm 9: Euclidean projections optimization

Input: Network edge weight $W \in \mathbb{R}^{n\times n}$, node score $\mathbf{z} \in \mathbb{R}^n$ and initial solution $\mathbf{x}^{(0)} \in \mathbb{R}^n_+$ which is randomly sampled from the uniform distribution [0,1] and then projected to the feasible region.

Output: Module indicator vector $\mathbf{x}$

1 **while** *Convergence or reach maximal iterations* **do**
2 | Update $\mathbf{g}$ in equation (4.5) by the gradient of $F(\mathbf{x})$ in equation (4.3);
3 | Solve optimal $\mathbf{x}$ in equation (4.5) by Algorithm 14 in Appendix C.2;
4 **end**

In order to identify multiple modules from one network, similar to [Zhao et al., 2011; Liu et al., 2014], we can find N modules by running the Algorithm 9 N times, with each

time simply extracting a module and subsequently delete the module from background network. The general procedure for identifying N modules from given gene expression profile can be summarized as Algorithm 10.

Algorithm 10: Active modules identification of GCN

Input: Gene expression profile $X \in \mathbb{R}^{n \times p}$, number of modules M
Output: M modules
1 **Construction**: Construct a weighted gene expression network G;
2 **Nodes scores**: Perform gene differential analysis to calculate fold-changes or p-values and assign to the genes as node scores;
3 **for** *iterations less than* M **do**
4 | **Solving**: Find solution $\mathbf{x}$ for (4.1) using algorithm (9);
5 | **Musk**: Delete the nodes in $\mathbf{x}$ and corresponding edges from G;
6 **end**

4.2.2 Empirical evaluation

Synthetic Data

Several related works have used to artificially generate data [Rajagopalan and Agarwal, 2005; Langfelder and Horvath, 2008] in order to test their algorithms in single network module identification. However, in our study, the simulated networks should have not only a clear topological structure but also consists of node scores. We follow [Li et al., 2011] to construct gene co-expression networks for simulation study as follows: Let n be the number of genes, and edge weights, as well as node score, follow the uniform distribution in the range $[0, 1]$. A module contains k genes inside which the edge weights as well as node score follow the uniform distribution in range $[\theta, 1]$, where $\theta = \{0.5, 0.6, 0.7, 0.8, 0.9\}$.

Table 4.1 shows the best performances of the MSCR algorithm in [Li et al., 2011] (details described as Algorithm 15 in Appendix section C.3.1) and Algorithm 9 in different networks. Our results shows that `AMOUNTAIN` can achieve better accuracy than the non-convex approach in [Li et al., 2011], especially when network size is larger.

Figure 4.1 shows the how parameters could affect the performance of both methods in the same network when networks size $n = 1000$ and real module size $k = 100$. We can

Table 4.1: Best performances of two methods in different networks, where n is the size of network and k is the size of true module. F score values (mean ± std) are calculated based on 50 runs.

Network specification	F score		Running time (s)	
	MSCR	AMOUNTAIN	MSCR	AMOUNTAIN
n=100, k=20	0.90±0.049	0.97±0.025	0.192	0.062
n=100, k=50	0.97±0.02	1.00± 0	0.186	0.068
n=500, k=100	0.92±0.017	1.00± 0	1.149	0.871
n=500, k=200	0.84±0.024	0.97± 0.01	1.147	0.83
n=1000, k=50	0.61±0.104	1.00± 0	2.877	1.916
n=1000, k=100	0.90± 0.027	0.99± 0.003	2.866	2.692

see that `AMOUNTAIN` is less sensitive to the parameters, i.e., it can accurately identify the ground true modules with a range of different parameters. We also tested the robustness of the proposed method by introducing small perturbations to the edge scores and node scores.

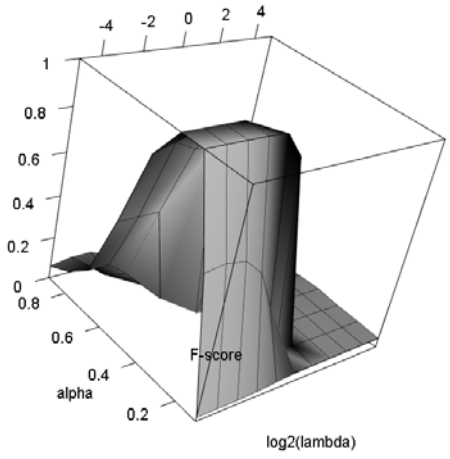


Figure 4.1: Parameters selection for module identification on large network with 10000 nodes and the module contains 100 nodes. With grid search we find the optimal $\alpha \in [0.3, 0.4]$ and $\log(\lambda) \in [-5, -1]$ lead to $F = 1$.

Real-world Data

We downloaded the gene expression profiles of human Th17 cell differentiation (GSE35103) from Gene Expression Omnibus (GEO) [Edgar et al., 2002]. The dataset was collected to

identify transcriptional changes induced by vitro polarization of human cord blood CD4+ cells towards Th17 subtype with combination of IL6, IL1b and TGFb [Tuomela et al., 2012]. There are 57 samples, consisting of 3 biological replicates of time series data (0, 0.5, 1, 2, 4, 6, 12, 24, 48 and 72 hours) of Th17 polarized cells and control Th0 cells [Pramila et al., 2002].

In order to deal with missing or invalid values, we discarded probes with more than 20% missing values or NAs, and replaced them with positions in a valid probe with k-nearest neighbors (KNN, $k = 10$) [Troyanskaya et al., 2001] output of the rest samples of that probe. We did not filter out genes by only selecting significantly expressed genes using a linear model, as `xHeinz` does [El-Kebir et al., 2015], because Algorithm 10 requires more information about genes correlation relationship to construct co-expression networks. Furthermore, the whole objective in (4.2) is consisted of two parts, and the gene activities only contribute part of it.

Algorithm 9 requires a weighted gene co-expression network as input. Here we just use the co-expression matrix as the edge score, where each entry W_{ij} in the symmetric means the correlation value of gene i and j, using all samples. The node score vector $\mathbf{z}$ is computed using package limma [Smyth, 2005] by different time points. In each time point, the expression level measurement p-values represent gene activities for Algorithm 9. As we want to maximize the objective, p-values are replaced by z-scores in practice. Correlation based similarity requires as many samples while gene activities are closely related to certain conditions, including the exposed time period.

We first run the algorithm on the single layer co-expression network. In order to investigate how the parameter λ in the objective (4.1) actually affects the identified module, we try different values on the same network and keep the module size strictly to be 100 by tuning the parameter α. From equation (4.1) we can see that larger λ adds more weights of the nodes activities, and in the extreme case the algorithm only finds top DEGs but ignores the genes correlation. On the contrary, very small λ leads to the module owning maximal edges weights but without considering the nodes scores. Figure 4.2 shows the

actual percentages of top DEGs contained in the modules identified by using different λ values, which is consistent with above claims.

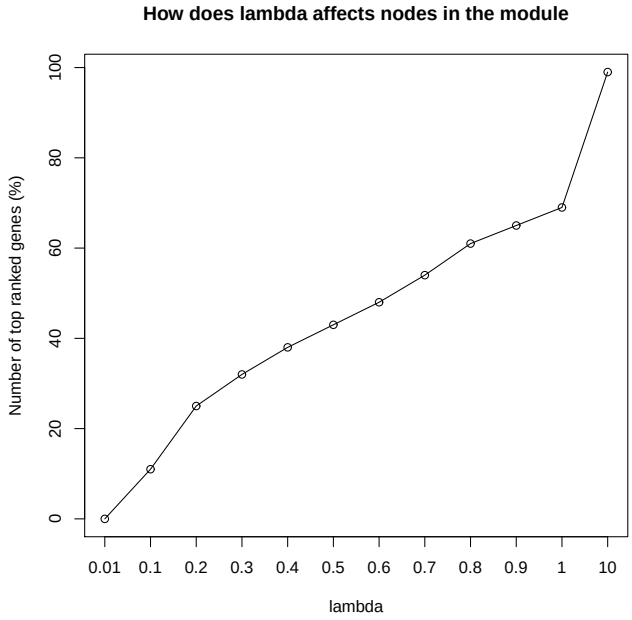


Figure 4.2: The proportion of differentially expressed genes in the first identified module, using different λ value in the objective (4.2). The module size is strictly constrained to be 100.

We first applied `AMOUNTAIN` with default $\lambda = 1$ to the Th17 single layer WGCN constructed in Material section. We provided the identified modules as gene lists in Supplementary files "**Table SS1.xlsx**".

Since cellular signaling mechanisms involve protein-protein interactions (PPIs) that transmit information, we first investigate whether the co-expression active module modules identified by `AMOUNTAIN` are enriched by PPIs. To this end, we used PPI enrichment analysis provided by PPI database STRING [Szklarczyk et al., 2014]. These PPIs include curated databases and experimentally determined and predicted interactions such as gene neighborhood and gene co-occurrence. Our results showed that 33 of 100 modules had significant PPI enrichment ($p<0.05$). If we relax the maximal module size to 500, the number of `AMOUNTAIN` modules with significant PPI enrichment is 50 (Supplementary file **Table SS2.xlsx**). These results indicate that `AMOUNTAIN` was able to find co-expression modules that are enriched by known PPI, which might reveal some signaling mechanisms

of a given cellular response.

In addition to PPI enrichment analysis, we also conducted KEGG pathways enrichment analysis to check whether the identified modules are enriched by known signaling pathways. The numbers of modules significantly enriched by KEGG pathways are 88 and 90 out of 100 identified modules, for maximal size 50 and 500 constraints respectively. The top enriched KEGG pathways include 1) Influenza A, which inhibits Th17 pathway activation by secondary bacterial challenge [Kudva et al., 2011]; 2) Hepatitis C, a common virus infection that could introduce Th17 cells [Rowan et al., 2008]; 3) Prolactin signaling, which may induce the production of Th17 [Hau et al., 2014] and 4) Jak-STAT signaling, which plays a central role in orchestrating of immune system, especially for cytokines involved in T helper cell differentiation [O'Shea et al., 2009; Seif et al., 2017]. (See supplementary file "**Table SS4**.xlsx" (size 50-100) and "**Table SS4**.xlsx" (size 50-500) for all modules).

We investigated the biological function of the identified modules using functional enrichment analysis with STRING [Szklarczyk et al., 2014]. Our results show that 55 out of 100 identified modules (with module size 50-100) are enriched by at least one GO term (biological processes) at a given FDR ($\leq$0.05) cutoff. If we relax the maximal module size to 500, 62 modules that are significantly enriched (FDR $\leq$0.05) by GO terms are found.

We listed the first 10 modules with the PPI and GO enrichment information in Table 3.3. We can see except for the 10th module, all the top 9 modules were enriched by PPI and biological progresses (See supplementary file "**Table SS3**.xlsx" (size 50-100) and "**Table SS4**.xlsx" (size 50-500) for all modules). We also found that there is a strong correlation between PPI and biological progresses, i.e., modules enriched by more protein interactions tend to have more significant GO terms.

Among these 10 modules, the first identified module was enriched by biological progresses and pathway related to Th17 differentiation in the early stage [Tuomela et al., 2012]. For example, we found that this module consisted of several important transcription factors such as STAT1/STAT2/STAT3, which are known regulators of the Th17

Table 4.2: Overview of top 10 modules identified from single layer WGCN of human, at 2 hour time point. PPI P-value indicates if there are significant known protein interactions in the module.

ID	Size	PPI P-value	Representative GO term (BP) and description	P-value
1	161	0	GO:0019221, cytokine-mediated signaling pathway	2.03E-18
2	190	0	GO:0044711, single-organism biosynthetic process	1.43E-9
3	294	4.02E-3	GO:0032479, regulation of type I interferon production	9.51E-6
4	150	1.54E-6	GO:0050860, negative regulation of T cell receptor signaling pathway	1.66E-6
5	234	0	GO:0000278, mitotic cell cycle	3.31E-21
6	301	0	GO:0000122, negative regulation of transcription from RNA polymerase II promoter	3.12E-8
7	248	4.88E-5	GO:0051726, regulation of cell cycle	1.77E-10
8	182	9.13E-3	GO:0006955, immune response	4.04E-8
9	73	3.31E-13	GO:0002764, immune response-regulating signaling pathway	1.01E-5
10	54	0.37	None	None

differentiation process [Durant et al., 2010]. These regulators were surrounded by other genes in the same cytokine signaling pathway (See Table SS1).

It is worth mentioning that the first identified module overlaps with the 29 differentially expressed genes (DEGs) identified by `limma` [Smyth, 2005], (See in Table SS1). As shown in Figure 4.3, there are 26 nodes (annotated with green color) shared between the first module identified by `AMOUNTAIN` and the DEGs. There are 3 DEGs were not included in the first identified module (annotated with red color). However, they were included in the second module identified by `AMOUNTAIN`. We speculate the reason for the significant overlap is partly because Th17 differentiation exhibits a high level of activity in the early stage [Tuomela et al., 2012]. However, from a different case study on ankylosing spondylitis disease samples (See Supplementary A2), the first identified module was different from DEGs.

Figure 4.3 shows the first identified module by `AMOUNTAIN` and the DEGs. The nodes shared between them were annotated with green color. Nodes only belonged to the module and DEGs were annotated with gray and red colors, respectively. We applied the cuff-off threshold (correlation coefficient ≤ 0.8) to delete those edges with low correlations. From the figure, we can see the goal of Algorithm 10: to find a module composed of high scored nodes (expressed genes), low scored nodes (ordinary genes) and their correlations.

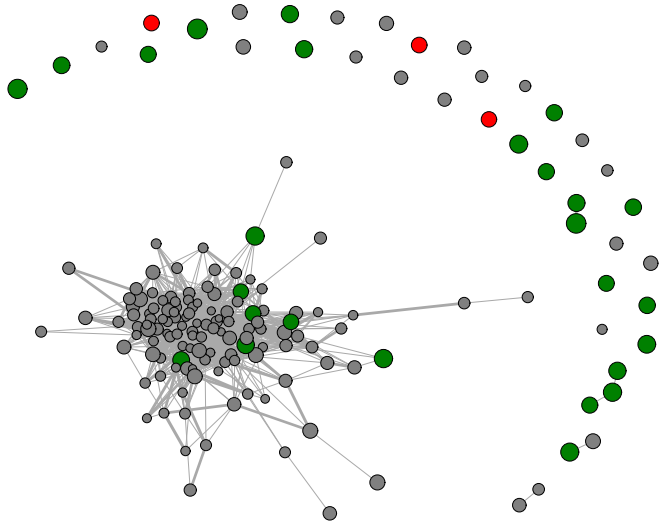


Figure 4.3: The first identified active module of single layer co-expression network, and the DEGs. In the module, node sizes are proportional to the intensities of gene activities and edge widths to the correlation coefficients (cut-off at 0.8). The green nodes are shared by identified module and DEGs, and the red nodes are only in DEGs.

4.3 Continuous optimization on multilayer network

We start from a simple case where inter-layer interactions only exist between neighborhood layers, then derive a compact form for multilayer networks without inter-layer links.

4.3.1 Algorithms

Multilayer network with inter-layer interactions

We first generalize the single layer active module identification problem to a two-layers WGCN. We define a two-layer active module as two modules in two different networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ but connected by inter-layer edges. The inter-layer edges were defined by $A = [a]_{ij} \in \mathbb{R}^{n_1 \times n_2}$ where n_1 and n_2 are the numbers of nodes in G_1 and G_2. The two-layer WGCN active module identification problem is formally defined as

Problem 6. Given two complete graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, with vertices weights $\mathbf{z}_{1v} \in \mathbb{R}$ for each $v \in V_1$ and $\mathbf{z}_{2v} \in \mathbb{R}$ for each $v \in V_2$. And non-negative edges weights $W_1 \in \mathbb{R}^{n_1 \times n_1}$ for edges in G_1 and $W_2 \in \mathbb{R}^{n_2 \times n_2}$ for edges in G_2. The inter-layer

interactions were measured by $A = [a]_{ij} \in \mathbb{R}^{n_1 \times n_2}$. The goal is to find two subgraphs $T_1 \in G_1$ and $T_2 \in G_2$ which both have large vertices weights and edges weights as well as intensive interaction with each other.

We use two variables $\mathbf{x}$ and $\mathbf{y}$ to represent the memberships of active modules in two different networks, $x_i > 0$ means the i-th node in the first network is in the module. Thus the optimization problem can be expressed as an extension to (4.2),

$$\begin{aligned} \max_{\mathbf{x}\in\mathbb{R}_+^{n_1},\mathbf{y}\in\mathbb{R}_+^{n_2}} \quad & F = \mathbf{x}^\top W_1 \mathbf{x} + \lambda_1 \mathbf{z}_1^\top \mathbf{x} + \mathbf{y}^\top W_2 \mathbf{y} \\ & + \lambda_2 \mathbf{z}_2^\top \mathbf{y} + \lambda_3 \mathbf{x}^\top A \mathbf{y} \\ \text{Subject to} \quad & \\ & f_1(\mathbf{x}) = 1 \\ & f_2(\mathbf{y}) = 1, \end{aligned} \tag{4.6}$$

where $f_1(\mathbf{x})$ and $f_2(\mathbf{y})$ are the vector norms on two vectors respectively. For simplicity we use the same Elastic net penalty $f(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + (1-\alpha)\|\mathbf{x}\|_2^2$ for both $\mathbf{x}$ and $\mathbf{y}$.

There is another parameter λ_3 in (4.6) controlling how much the inter-layer links affect the resulting modules. Take multi-species for example; large λ_3 can lead to conserved modules across different species which may reveal some gene conservation in response to certain changes. Conversely, small λ_3, to the extreme case, $\lambda_3 = 0$ makes the inter-layer information playing no role, thus leading to two independent module identification processes.

In order to solve (4.6) we use the alternating optimization, i.e. iteratively optimizing one variable while fixing another each time [Lin, 2007]. Dealing with one variable has the same form as in (4.2). And each iteration in the procedure can be simply expressed as:

- Find $\mathbf{x}^{(k+1)}$ such that $F(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)}) \leq F(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$ and,
- Find $\mathbf{y}^{(k+1)}$ such that $F(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}) \leq F(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)})$

Multilayer network without inter-layer interactions

The complete algorithm to find multiple modules in the two-layer network shares the same structure of Algorithm 10. In a multilayer situation, the rationale remains the same as in the two layer case, alternating optimization can be used as the same way. Otherwise a more compact tensor computational paradigm [Li et al., 2011] can be more efficient without inter-layer links consideration. Being different from [Li et al., 2011] here we use node activities to search modules and the method is based on elastic net regularization. The multilayer network module identification problem is formally defined as

Problem 7. Given an L-layers network with each layer a complete graph $G = (V, E)$ where $|V| = n$. The vertices weight and non-negative edges weight in the i-th layer are $\mathbf{z}_i \in \mathbb{R}^n$, $W_i \in \mathbb{R}^{n \times n}$ respectively. The goal is to find a conserved subgraph T with large vertices weight $\sum_{k=1}^{L} \sum_{i \in T} z_i^{(k)}$ and also edges weights $\sum_{k=1}^{L} \sum_{i,j \in T} w_{ij}^{(k)}$.

The corresponding objective function is

$$\begin{aligned} \max_{\mathbf{x} \in \mathbb{R}_+^n} F &= \sum_{k=1}^{L} (\mathbf{x}^\top W^{(k)} \mathbf{x} + \lambda_k \mathbf{z}^{(k)\top} \mathbf{x}) \\ \text{Subject to} \\ f(\mathbf{x}) &= 1 \end{aligned} \tag{4.7}$$

where λ_k controls the trade-off between edges weights and nodes weights in the k-th layer and $f(\mathbf{x})$ is the vector norm such as Elastic net penalty $f(\mathbf{x}) = \alpha \|\mathbf{x}\|_1 + (1-\alpha)\|\mathbf{x}\|_2^2$. The optimization method follows the same way as in Algorithm 9.

4.3.2 Empirical evaluation

Cross-species network

Inspired by `xHeinz` [El-Kebir et al., 2015], in addition to *Homo sapiens* dataset GSE35103, we used *Mus musculus* Th17 cell differentiation dataset (GSE43955) for the multilayer

cross-species co-expression study. The original papers [Yosef et al., 2013] and [Tuomela et al., 2012] reported the expression profiles identification controlled by the differentiation of Th17 cell.

The cross-species co-expression network, with the expectation to find evolutionarily conserved modules. Following the case study in `xHeinz` [El-Kebir et al., 2015], we used the two gene expression datasets (GSE35103 and GSE43955) to construct a two layer cross-species network, of which each layer is the co-expression of a species. The inter-layer connections were defined by the orthology information, obtained from Ensembl 85 [Yates et al., 2015] (`http://www.ensembl.org`). We used the associated gene name as the unique identifier for each gene (node) in both human and mouse, and the corresponding orthologous mapping table were embedded into this two-layer network. After gene expression data pre-processing and orthologous selection, we get 28870 genes in human layer and 22192 genes in mouse layer. There were 11039 links between two layers, standing for confident orthologous mapping pairs. To keep as many inter-layer links, we do not filter out low confident orthologous scores.

We applied `AMOUNTAIN` to the two-layer cross-species co-expression network of human and mouse Th17 differentiation. We selected an early time point 1 hour because the following reasons: 1) there are more activities in the early phase of Th17 differentiation in both in mouse [Ciofani et al., 2012; Yosef et al., 2013] and human [Tuomela et al., 2012]. And 2) there are enough replicates for both species at this time point. We did not select 2h as done in `xHeinz` because there is only one replicate for the mouse at both time points, which can not be used to calculate of node score.

As mentioned at section 4.3.1, λ_3 in equation (4.6) controls the number of inter-layer links or conserved genes in the resulting modules. The detailed modules identified under different λ_3 are shown in Table 4.3 and we can see that as λ_3 increases the number of inter-layer links (conserved genes) also increases.

We applied our algorithm to the cross-species network of human and mouse Th17 differentiation at 1 hour with default value $\lambda_1 = \lambda_2 = 1$. We set $\lambda_3 = 1000$ (by Table

4.3). The identified 100 modules with their size and shared inter-layer links information are stored in supplementary Table SS 5. Due to the space limit, we selected the first identified conserved module for further analysis.

In the first identified conserved module, there are 52 and 57 genes in human and mouse layers, respectively. There are 52 conserved genes which include several key genes such as STAT2/SOCS3/IRF1 in TH17 cell differentiation [Zhu et al., 2010; Yang et al., 2013; Karwacz et al., 2013]. The completed gene list of the 2-layer modules is provided in supplementary file "**Table SS1**.xlsx".

In order to illustrate the potential conserved signaling mechanisms, we overlaid known interactions from STRING to the (nodes) genes extracted from the two layers as shown in Figure 4.4 and 4.5. (See the corresponding co-expression modules in Figure S6 and Figure S7 in supplementary text A1). We can see that genes in both layers centered around STAT2/Stat2 and IRF1/Irf1, which are the key transcription regulators of Th17 cell differentiation [Zhu et al., 2010; Yang et al., 2013; Karwacz et al., 2013].

Our functional and KEGG pathway enrichment results show that both modules share some common pathways such as the JAK-STAT signaling pathway and those pathways are relevant to Th17 differentiation. The detailed top enriched biological functions and KEGG pathways enriched by modules from both species are listed in Table 4.4.

Table 4.3: Modules identified from two species, using objective (6) in main body when $\lambda_1 = 1$ and $\lambda_2 = 1$. The second and third columns are module size and last column are the number of conserved genes.

λ_3	Human	Mouse	Conserved
1	51	50	5
10	67	78	9
100	63	61	14
1000	52	57	52
10000	55	53	53
100000	68	63	63
1000000	71	69	69

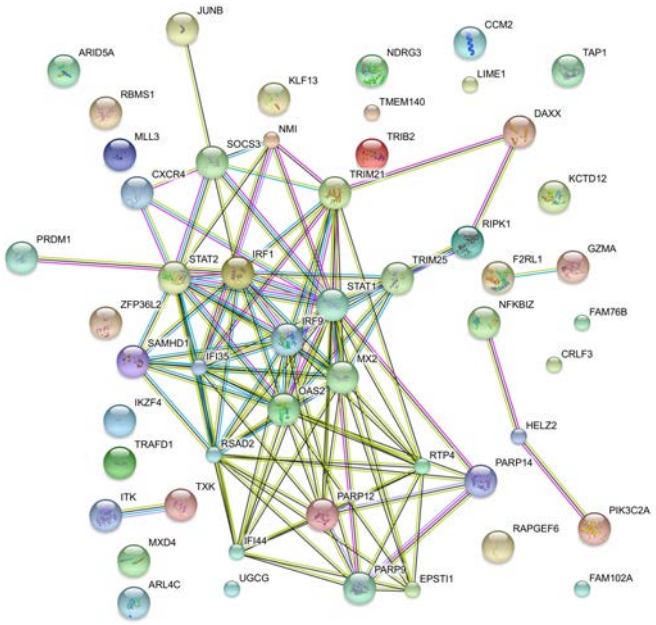


Figure 4.4: The first identified module in the human layer at 1 hour time point, plotted by STRING, where edges represent the known interactions. Colored nodes standard for query proteins and first shell of interactors, and white nodes for second shell of interactors.

Multilayer dynamic network

In order to evaluate the performance of our algorithm on multilayer networks with more than two layers, we constructed a dynamic co-expression network of human Th17 differentiation based on dataset GSE35103. We aim to apply `AMOUNTAIN` to this dynamic co-expression network to identify conserved active modules. We, therefore, selected the expression data of several time points (such as 0.5h, 1h, 2h, 4h, 6h or 12h, 24h, 48h) in GSE35103, each time point as one layer. Ideally layer x should be constructed from the samples belong to time point x. But there are only three replicates of each time point which makes the correlation values suspicious. Therefore we use all samples of these time points to construct the co-expression network for each layer and calculate gene activities from corresponding time points. In other words, each layer shares the same edges scores but with different nodes scores.

The node scores of our dynamic co-expression network is calculated from the gene expression profiles at three later time points, i.e., 12h, 24h and 48h. This is because Th17 differentiation showed that the effective secretion of Th17 hallmark cytokines only

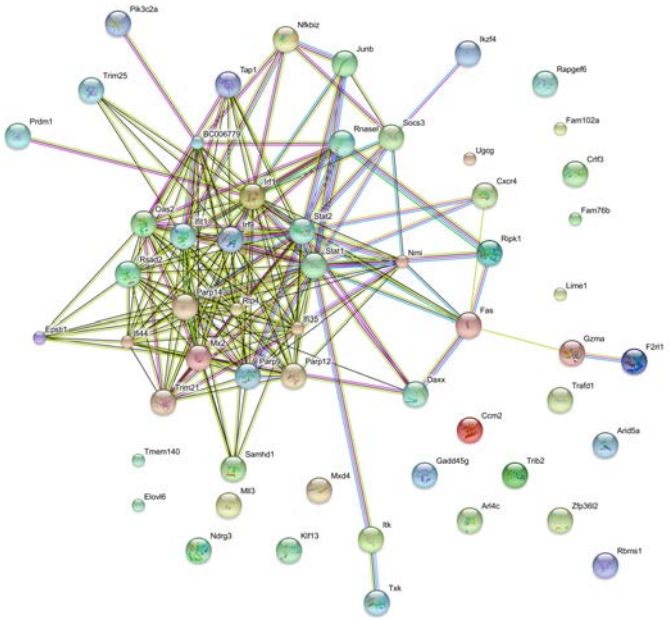


Figure 4.5: The first identified module in the mouse layer at 1 hour time point, plotted by STRING, interactions are denser compared with human layer. Key transcriptional factors Stat2/Irf1 are densely surrounded by interactions.

happens after several days of polarization [Tuomela et al., 2012; Yosef et al., 2013]. In essence, we constructed a 3-layer dynamic co-expression network of three later time points during human Th17 differentiation.

In order to unveil the dynamic regulatory and signaling mechanisms of the Th17 differentiation, we applied our `AMOUNTAIN` algorithm to the Th17 dynamic multilayer network (See Materials). `AMOUNTAIN` can readily identify conserved modules across the later time points (12h, 24h, and 48h). Also, to depict the dynamic changes of co-expression networks, we identified time point specific modules by applying `AMOUNTAIN` to each layer. Figure 4.6 shows the first identified conserved module and Figure S8-S10 in supplementary text A1 show the three time point specific modules respectively.

The conserved module identified from the three-layers dynamic network includes several signature genes of Th17 lineage commitment, e.g., RORC and RUNX1 [Lazarevic et al., 2011; Villarino et al., 2010], which shown significantly different gene expression profile compared with Th0 group (See Sheet 1 of "**Table SS5**.xlxs" in supplementary files). We found that in the conserved module, RORC gene always interacted with VDR

Table 4.4: The first identified module at 1 hour for human and mouse, when $\lambda_3 = 1000$ in the objective function and 52 conserved genes are found.

Human		Mouse	
Biological Process (GO) given by STRING			
Function	FDR	Function	FDR
Type I interferon signaling pathway	4.59e-12	Innate immune response	1.61e-9
Cellular response to type I interferon	4.59e-12	immune response	1.95e-08
Response to virus	6.91e-10	defense response	2.07e-08
Cytokine-mediated signaling pathway	6.91e-10	defense response to virus	4.32e-08
defense response to virus	9.91e-09	immune system process	5.47e-08
KEGG pathway given by STRING			
KEGG pathway	FDR	KEGG pathway	FDR
Hepatitis C	2.46e-04	Influenza A	2.88e-09
Influenza A	4.89e-04	Hepatitis C	4.43e-09
Herpes simplex infection	4.89e-04	Herpes simplex infection	4.43e-09
Osteoclast differentiation	1.08e-03	Measles	7.85e-05
Jak-STAT signaling pathway	0.034	Osteoclast differentiation	7.24e-04

(Vitamin D Receptor), which is very relevant to T cell development and differentiation [Chang et al., 2010; Kongsbak et al., 2015]. Another interesting finding is that, in the conserved module, RORC gene also interacted with BHLHE40, a transcription factor that controls cytokine production by T cells [Lin et al., 2014].

By comparing the conserved modules with the three time point specific modules, we found that some genes only connected to RORC at specific time points. For example, RBPJ was only identified in the time point specific module from the network at 24h, which is a known regulator of the Notch signaling pathway [Tanigaki and Honjo, 2010] and play an important role in lineage fate decisions in cells. The above result indicates that the co-expression active modules identified from the dynamic multilayer network using our algorithm can provide insights into the dynamics of Th17 lineage commitment.

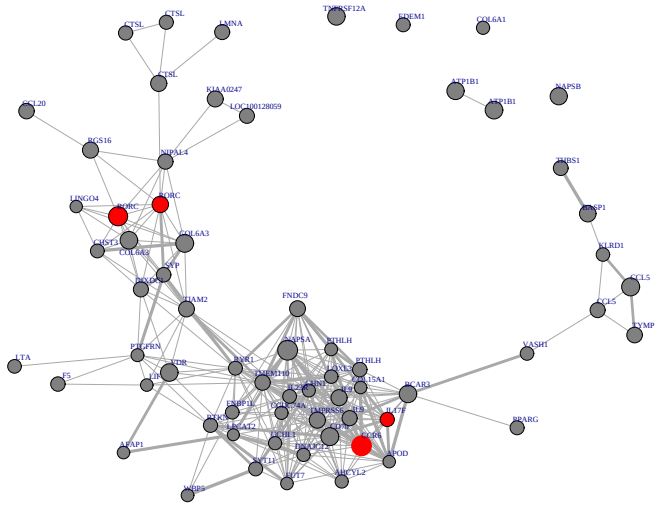


Figure 4.6: The first identified conserved module for a three layer network where each layer represents nodes from 12h, 24h and 48h respectively. The red nodes are two probes of gene RORC, a signature gene w.r.t. Th17 lineage commitment. Plotted by `igraph` [Csardi and Nepusz, 2006].

4.4 Discussion

4.4.1 Related works

Although algorithms in [Zinman et al., 2015; El-Kebir et al., 2015] can identify evolutionarily conserved PPI modules from 2-layer cross-species networks, to our best knowledge, general active module identification algorithms for multilayer gene co-expression networks do not exist. The most relevant algorithm is the algorithm in [Li et al., 2011], which was proposed to identify heavy recurrent modules from multiple gene co-expression networks. However, this algorithm differs from `AMOUNTAIN` in the follow perspectives: 1) The algorithm [Li et al., 2011] only considers edge weights while `AMOUNTAIN` considers both edge weights and node scores (hence the modules are called active modules); 2) The algorithm in [Li et al., 2011] is specifically designed for multi-slice (multiplex) networks, which share exactly the same set of nodes, while `AMOUNTAIN` algorithm is designed for multilayer networks which are more general. For example, different layers could have different sets of nodes, and the inter-layer interactions can be considered; 3) the algorithm in [Li et al., 2011] is based on a non-convex regularization while `AMOUNTAIN` algorithm adopts a convex

regularization which is more efficient to achieve sparsity.

4.4.2 The optimization problem

Maximizing the constrained quadratic function (4.3) with indefinite matrix is NP-hard [Burer and Letchford, 2009]. In a different context of shape matching in computer vision, [Rodola et al., 2013] solved the same problem, i.e., the objective function (4.3) using a projected gradient method. The only difference is the procedure to solve the subproblem (4.5) since their target solution was not sparse as us.

[Li et al., 2011] solved a similar problem using power method, followed by a normalization step. We have shown our projected gradient method performed better than the power method in terms of convergence rate and accuracy in section 4.2.2.

4.4.3 Difference with `WGCNA`

Both `WGCNA` [Langfelder and Horvath, 2008] and `AMOUNTAIN` take the weighted gene co-expression network as input, it is necessary to point out the difference. Although `WGCNA` was not proposed to identify active modules, we can relate phenotypic traits such as gene expression levels, to clusters. Compared with `AMOUNTAIN`, which identifies more modules with significant PPI enrichment with fewer genes, `WGCNA` identified modules with a large proportion of isolated genes, although they might have similar biological functions. This is the main difference between `WGCNA` and `AMOUNTAIN`: the former partitions the whole weighted network and group the genes with similar biological functions (except one 'grey' module for unrelated genes), but the latter aims to extract active modules with significant node activities, which include high scored genes, ordinary genes and their interactions. These active modules could be used as hypotheses of the signaling and regulatory mechanisms of a given cellular response [Mitra et al., 2013]. Due to the size controlling of the active modules, `AMOUNTAIN` can identify small modules which facilitate follow-up experiments to test the hypotheses.

4.5 Chapter summary

This chapter describes `AMOUNTAIN`, a general and efficient active modules identification algorithm for single layer and multilayer WGCNs. The proposed algorithm is based on a new definition of active modules in WCGNs. This definition enables us to formulate the module identification problem that not only considers the correlation between genes but also their activity. We also generalized the active module identification problem in single layer WGCNs to multilayer WGCNs. Another main contribution is the continuous optimization formulation of the problem, which achieves better efficiency when dealing with large-scale networks.

We provide `AMOUNTAIN` as an R package which is freely available at Bioconductor. We expect `AMOUNTAIN` algorithm can be applied to a wide range of problems that involve identifying dynamic and evolutionary mechanisms associating with a cellular response.

CHAPTER 5

NETWORKS COMPARISON

Last two chapters describe the main topics of this thesis, the active modules identification on two classes of biological networks. These networks, as well as the modules, are from the same context, in other words, they normally model the system under the same condition. However, in some real world problems, there is a need to explore the organizational mechanism under different conditions, such as multiple biological stresses or environmental changes. In this chapter, we extend the modules identification to the networks comparison, which compares the multiple biological networks constructed from multiple conditions via modules. Section 5.1 introduces the background of this problem, including related works. Section 5.2 describes how to conduct networks comparison in details. And Section 5.3 evaluates the proposed method on simulated data as well as real world data. Finally, section 5.4 summarizes the chapter.

5.1 Introduction

In addition to simple weighted gene co-expression network analysis, there is a need to integrate a set of samples belonging to different biological stresses. In research area like ecology, we may need to explore mechanisms of complex biological processes exposed to multiple biotic and abiotic environmental perturbations [Orsini et al., 2016]. The ideal case is to construct each condition-specific network with samples just from that condition.

However, the number of samples belonging to each condition is limited due to the costs. It is required to make use of these samples more effectively.

Several previous works went beyond individual gene differential analysis. GSCA [Choi and Kendziorski, 2009] detects a set of differentially co-expressed (DC) genes. DICER [Amar et al., 2013] uses a probabilistic framework to detect DC gene sets. Both take genes as individuals and did not provide a systematic view (at the network level) of expression profiles. DINGO [Ha et al., 2015] estimates group-specific networks by calculating differential scores between each pair of two genes, which is focused on individual edges in the networks. DINA [Gambardella et al., 2013] can identify condition-specific modules from a collection of condition-specific gene expression profiles, which requires a certain number of samples for each biological conditions.

Biological network comparison [Sharan and Ideker, 2006; Pržulj, 2007] or network alignment [Kuchaiev et al., 2010] is also a related but more general topic, which takes the overall network as an object to tell the difference. Comparing sample-specific or context-specific networks [Gao et al., 2016] can provide a picture of different conditions or dynamics. Previous works such as differential co-expression network [Hsu et al., 2015] takes the whole network as an object, or overlaying the differentially expressed genes to the co-expression networks [Lui et al., 2015]. The ideal case is to construct each condition-specific network with samples just from that condition. However, the number of samples belonging to each condition is limited due to the costs. It is required to make use of these samples more effectively.

Following the three steps of network analysis:

- Network construction: to construct the network from various data, but a WGCN is purely constructed from microarray or RNA-Seq.
- Network analysis: to extract basic topological features and high-order patterns, such as modules.
- Result interpretation: to associate these features or patterns with biological process,

pathways or diseases.

We propose modules differential analysis (`MODA`) to fill the gap between existing weighted gene co-expression network analysis methods and the need to explore gene expression behaviors affected by multiple conditions. We extend the individual gene expression analysis to gene modules, taking the modules identified from weighted gene co-expression networks as basic units. Specifically, `MODA` allows us to construct a collection of networks from multiple conditions and compare the differences between the networks based on the modules, which is finally associated with known biological process, pathways or diseases by enrichment analysis. The whole pipeline is designed to gain insights from transcription profiles under different conditions at the system level.

5.2 Modules differential analysis

We implemented `MODA` as an R Bioconductor package, aiming to establish a pipeline from gene expression profile to biological explanations. The input of `MODA` is gene expression profile $X \in \mathbb{R}^{n \times p}$ from multiple conditions, where n is the total number of experimental samples, and p is the number of genes. X_{ij} means the expression value of the j-th gene in i-th sample. n samples are divided into k groups according to the experimental conditions (e.g., different chemicals or concentrations of the same chemical), each has a limited number of samples. The goal is to explore possible biological functions associated with these conditions. We describe how `MODA` works in the following three parts.

5.2.1 Networks construction

We use the same scheme of `WGCNA` to construct one weighted network. Given the expression profile X, the entries of adjacency matrix are defined as

$$s_{ij} = |cor(x_i, x_j)|^{\beta}, \tag{5.1}$$

where the power parameter β is picked up based on the scale-free topology criterion [Zhang and Horvath, 2005].

As mentioned above, the desired number of samples for constructing a condition-specific network should be large enough. While in practice, it is expensive to make many samples/replicates for just one condition. In the RNA-Seq data set obtained from two natural genotypes of Daphnia magna [Orsini et al., 2016], there are three replicates for each environmental perturbation.

Inspired by [Kuijjer et al., 2015], we use a sample-saving approach to construct condition-specific co-expression networks for each single condition, which works as follows. Assume network N_b is background, generally containing samples from all conditions, is constructed based on the correlation matrix from all samples. Then condition-D-specific network N_1 is constructed from all samples **except** samples belonging to the certain condition D [Kuijjer et al., 2015]. Thus for both networks, we have enough samples to calculate relatively reliable correlation coefficients weights for network edges. The differences between network N_b and N_D is supposed to be introduced by the effects of condition D. The rationale behind this criteria is based on the mechanism of correlation, i.e., which samples can make an impact on the correlation coefficients while others may not? Figure 5.1 illustrates an extreme example of how the additional two samples may affect the correlation between vector X and Y.

The given samples are divided into k groups by the conditions, from which we construct a background network N_b and a set of condition-specific networks $N_i, i = 1, 2, ..., k$ as such. The goal is to compare N_b with each N_i. Note that power parameter β in equation (5.1) should keep the same in all condition-specific networks N_i as in N_b, which can be determined by all samples when constructing N_b.

5.2.2 Modules detection

After getting a set of networks, the second step is to identify modules from each network. As a general framework, `MODA` provides several options in this stage. The default one is

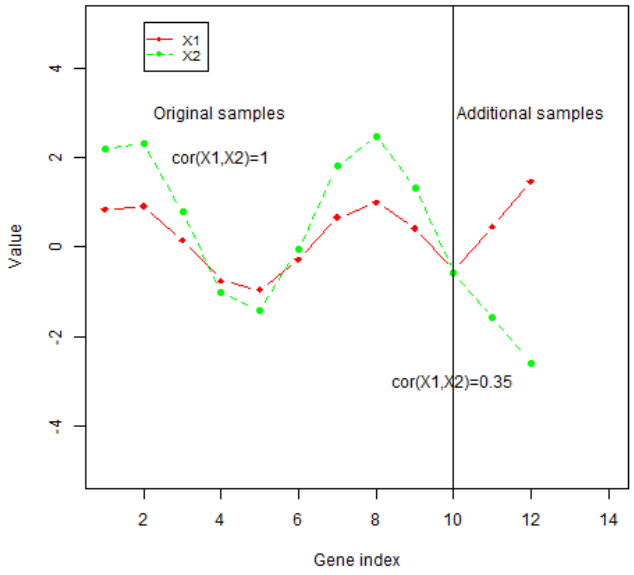


Figure 5.1: Scatter plot of variable X_1 and X_2. Removing or adding the last two data points would make an impact on their correlation coefficient.

based on `WGCNA`, which is based on hierarchical clustering but embedded with a mechanism to pick the optimal cutting height. If the resulted module size is too large to interpret, we can also use several other methods by leveraging existing tool `igraph` [Csardi and Nepusz, 2006] combined with a mechanism to control the module size.

Hierarchical clustering

`WGCNA` adopts the classical hierarchical clustering on a special considered Topological Overlap Matrix (TOM) [Zhang and Horvath, 2005], and the cutting height of a hierarchical clustering tree is determined by the dynamic tree cut method [Langfelder et al., 2008]. In `MODA`, we use an automatic method to determine the optimal cutting height based on the quality of modules. Inspired by the concept of partition density of link communities [Ahn et al., 2010], our method searches for the optimal cutting height that can maximize the average density of resulting modules. Here we simply define the module density [Zhang

and Horvath, 2005] as the average edge weights in module A as:

$$Density(A) = \frac{\sum_{i \in A} \sum_{j \in A, j \neq i} a_{ij}}{n_A(n_A - 1)}, \tag{5.2}$$

where a_{ij} is the weight (normally the similarity) between gene i and gene j, and n_A is the number of genes in A. We can also use the modularity Q of weighted network A [Newman, 2004a] as the criterion to pick the height of hierarchical clustering tree:

$$Q = \frac{1}{2m} \sum_{i,j} [a_{ij} - \frac{k_i k_j}{2m}] \sigma(c_i, c_j), \tag{5.3}$$

where m is the number of edges and k_i is the connectivity (degree for unweighted network) of gene i, defined as $\sum_j a_{ij}$. And $\sigma(c_i, c_j) = 1$ only when gene i and j are in the same module. The optimal cutting height of the dendrogram can make the average modularity to be maximal. It turns out such criteria works for real-world data and there exists the unique optimal height in many cases. The complete module detection and average density is shown in Figure 5.2.

Community detection

Clustering based modules identification methods are limited from two aspects. 1) Hierarchical clustering simply categorizes "similar" genes together but ignore the possible community structure of the networks, which is verified in different biological networks [Girvan and Newman, 2002]. 2) The module size is difficult to control by cutting the dendrogram. Normally there are several large modules which make them difficult to interpret. `MODA` provides alternatives to use mainstream methods to find the communities on a weighted network [Fortunato, 2010], by incorporating with `igraph` R package. Furthermore, the module size is guaranteed by introducing a recursive way to find modules from large modules. Take the popular Louvain algorithm [Blondel et al., 2008] for example, `MODA` identifies the modules through the Algorithm 11.

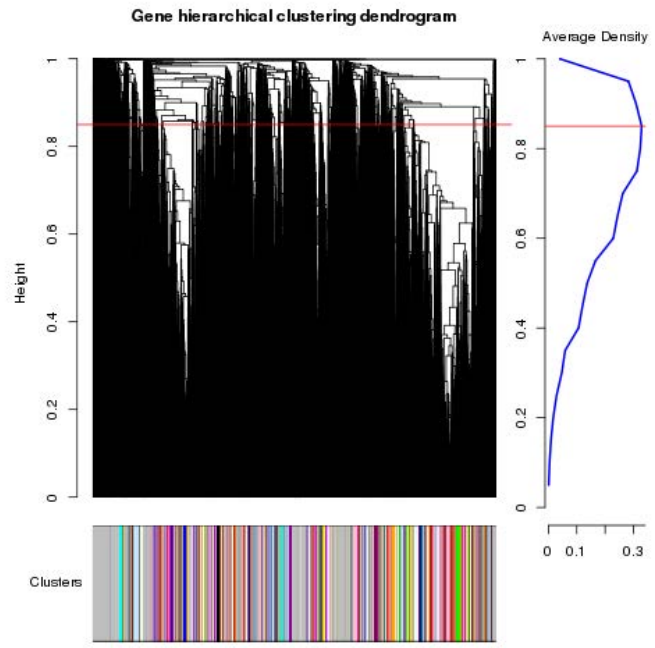


Figure 5.2: An score based function to pock the optimal height of hierarchical clustering tree in MODA. The right hand figure shows the cutting maximizes average partition density of resulted modules.

5.2.3 Network comparison

After networks construction and modules detection for each network, we have already represented a network as a collection of modules. The principle of MODA is to compare each condition-specific network N_d with the background network N_b, and find which module in N_b has large or small overlapped parts with modules in N_d. We define a similarity matrix $S \in \mathbb{R}^{n_1 \times n_2}$, where n_1 is the number of modules of N_b and n_2 is the number of modules of N_d. Each entry S_{ij} means the similarity between the i-th module from the network N_b (denoted by $N_b(A_i)$) and j-th module from the network N_d (denoted by $N_d(A_j)$). One of the most straightforward ways is just considering the vertex set of two modules since the biological evaluation is mainly considered from the gene list. And this metric is specified by a Jaccard similarity coefficient,

$$S_{ij} = \frac{N_b(A_i) \cap N_d(A_j)}{N_b(A_i) \cup N_d(A_j)}. \tag{5.4}$$

Other criteria that can be used for comparing two (sub)graphs can also be used here

Algorithm 11: Recursive modules identification

Input: The weighted graph g, maximal module size N_{max} and minimal module size N_{min}

Output: The list of modules.

```
Modules detection: Obtain k modules on g by Louvain algorithm;
for each modules g_i, i in 1,2,..k, do
    if |V(g_i)| > N_max then
        Call the algorithm and take g_i as input;
    else if |V(g_i)| < N_min then
        Discard;
    else
        Save the module as node list;
end
```

to replace equation (5.4) as long as it is a metric, which should satisfy non-negativity, symmetry and triangle inequality. Available extensions applicable on graphs include common graph similarities [Koutra et al., 2011], mutual information [Escolano and Hancock, 2014], special designed metric [Xu et al., 2013] and network properties comparison [Pržulj, 2007].

Since entries of S are non-negative, we use the sum of each row of S (vector denoted by $\mathbf{s}$) to indicate how much degree of modules in N_b are affected by the corresponding condition. The higher s_i means the module i in N_b may just be responsible for general stress. Especially when some s_i keeps relatively high compared with all N_j ($j = 1, 2, ..., k$ by removing one condition each time), showing these modules have little association with any specific conditions. We call such modules as **conserved modules**. On the contrary, lower s_i means module i in N_b is very different from the modules in N_j, which may indicate this module has some connection with the corresponding condition. Especially when some s_i in N_b only keeps relatively small compared with just one network N_j, showing this modules may be affected by the specific condition j. And we call such modules as **condition-specific modules**. There are two parameters to determine in which case modules should be annotated as **conserved modules** or **condition-specific modules**. When fixed cutting-off does not work well, an adaptive choice can be used: 1) θ_1, the threshold that defines a specific module is set to be $min(\mathbf{s}) + \theta_1$; and 2) θ_2, the

threshold that defines a conserved module across conditions is $max(\mathbf{s}) - \theta_2$. An example of resulted bar plot of all *interested* modules is shown as Figure 5.6, where short bars above indicate **condition-specific modules** and long bars below mean **conserved modules**. Figure 5.3 shows the overview of `MODA`, showing each step mentioned above.

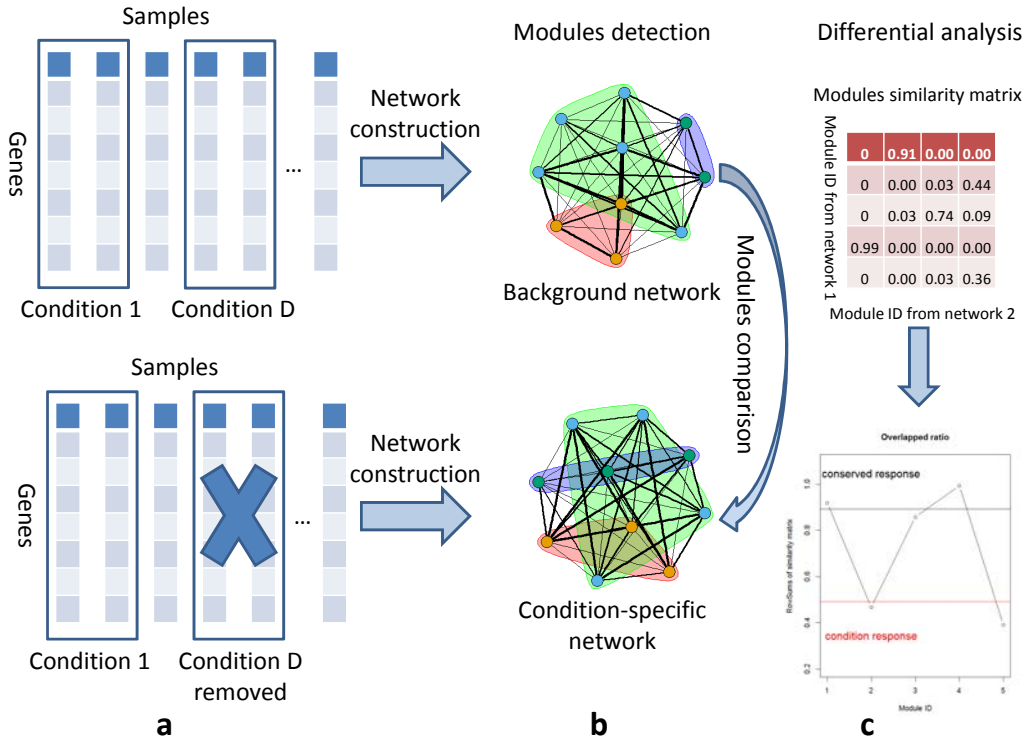


Figure 5.3: Overview of `MODA`. (a) Condition-specific networks construction from a set of samples. (b) Modules identification using various methods. (c) Module differential analysis based on the similarity of each two modules from background network and condition-specific network.

After determining which module may be condition-specific or conserved, we get a module as a gene list. The condition-specific modules are supposed to have relationship with corresponding conditions, while conserved modules may correspond to general responses. The principle of interpretation suggested by `MODA` is based on GWAS evaluation, i.e., associating the biological processes, pathways or diseases with the module by correlating the gene lists with multiple GWAS datasets. Blast2GO [Conesa et al., 2005] searches functional annotations on genomics, providing a direct way to associate biological pro-

cesses to gene sequences. There are also several integrative tools available, such as gene list enrichment analysis tool Enrichr for human [Chen et al., 2013], which does not only provide the pathway and gene ontology enrichment analysis but also has a visualization tool with each. DAVID [Huang et al., 2008] is also a commonly used one for multiple species, and there is an R Webservice interface [Fresno and Fernández, 2013] enabling the interpretation step automatic. GeneMANIA [Warde-Farley et al., 2010] provides an user-friendly web interface for generating hypotheses about gene function with multiple species.

5.3 Empirical evaluation

5.3.1 Simulated study

The purpose of simulated study is to validate `MODA` on gene expression profile with known modular structure. The basic synthetic gene expression data is generated by the following logic: given desired correlation matrix $C \in \mathbb{R}^{n\times n}$ with n genes which have a clear modular structure that all genes are equally divided into k groups according to the similarities. Then we conduct the Cholesky decomposition on C such that $C = LL^T$, where $L \in \mathbb{R}^{n\times n}$ is the lower triangular matrix. Finally, we project L on random matrix $A \in \mathbb{R}^{m\times n}$ to get desired gene expression matrix $X \in \mathbb{R}^{m\times n}$, which has the rough modular structure defined by correlation C. Let the gene number $n = 500$ in the simulation.

We include two simulated datasets in the R package. The genes in $datExpr1$ are divided into five clusters, and each has size 100. The first three clusters of $datExpr2$ are generated in the same way of $datExpr1$, but the last cluster of $datExpr2$ uses randomly picked genes from $datExpr1$. The level plot of correlation matrix of X_1 is shown in Figure 5.4. The other dataset X_2 is part of X_1 by removing samples that distinguish the last two clusters, which makes the genes divided into 4 clusters and the last has size 200. The level plot of correlation matrix of X_2 is shown in Figure 5.5. Then we can compare these

two networks with proposed method to see which genes were affected.

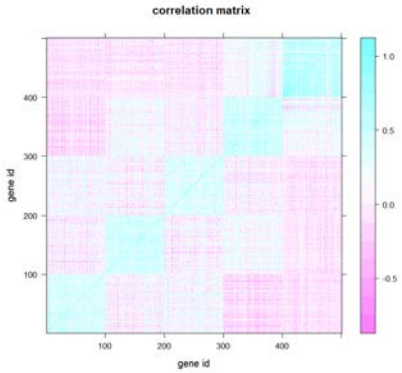


Figure 5.4: Levelplot of correlation matrix calculated on simulated gene expression profile X_1. There are five modules, each module has roughly identical size 100. But the last two modules contain similar samples.

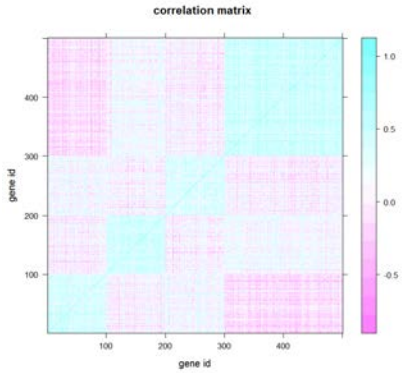


Figure 5.5: Levelplot of correlation matrix calculated on simulated gene expression profile X_2. There are four modules, and each of three has the size 100 but the last module has larger size 200.

The result (stored as plain text by `MODA`) also show that modules that contain gene id from 1-100, 101-200 and 201-300 have large overlap with those of network 2, while gene id from 301-500 have least overlap with network 2. The facts are consistent with data generation settings, which validate that the rationale of constructing condition-specific network, and `MODA` reveals the relationship between modules and external conditions.

5.3.2 Application to Ecology

`MODA` has been used as the main computational pipeline to analyze the early transcriptional response of Daphnia magna to twelve environmental perturbations, including biotic and abiotic stressors [Orsini et al., 2017]. The study associated uncharacterized genes within co-responsive modules to genes of known function and to specific environmental perturbations.

5.3.3 Application to Nanotoxicology

To validate MODA in real-world, we reanalyzed published gene expression data [Poynton et al., 2012], which investigated gene expression profiles of Daphnia magna, a freshwater crustacean and common indicator species for toxicity, after acute exposure to different sublethal concentrations of silver nanomaterials using a 15k oligonucleotide microarray (GSE35150). The two nanomaterials used, AgNO3 and AgNPs, induced divergent expression profiles that were interpreted by the authors as suggestive of different mode of toxicity. The study found a total of 466 significantly differentially expressed (DE) genes.

There are 15k genes across 60 samples in total, which are further divided into 7 conditions: three different concentrations of AgNO3, two concentrations of citrate-coated silver NP and two concentrations of PVP-coated silver NP, making the input of MODA. After getting modules in all networks, we use the similar way as in [Poynton et al., 2012] to predict the module functions. Specifically, the module genes (denoted by DMXXXXXX) were mapped into Entrez Gene IDs, and then Batch Entrez (https://www.ncbi.nlm.nih.gov/sites/batchentrez) was used to find the corresponding Gene Bank accessions and nucleotide sequences in FASTA format. Based on these sequences, Gene ontology (GO) terms and annotation were done through Blast2Go [Conesa et al., 2005]. The functions of the module genes thus can be summarized from the GO terms. The full tables of description and GO of each module are appended in supplementary Table S1. The parameters we used in pipeline can be found in Table 5.1.

Table 5.1: Parameters in the pipeline of MODA, on Daphnia magna microarray data GSE35150.

Steps	Parameters
Network construction	$\beta = 6$ in (5.1)
Modules detection	Maximal module size 200, minimal 50
Functional interpretation	BLAST Expatiation value 1E-5, others default

It turns out there are 118 modules in the background network, which are compared with modules from seven conditions respectively. The frequency of each module is annotated as condition-specific or conserved is shown in Figure 5.6, based on module similarity

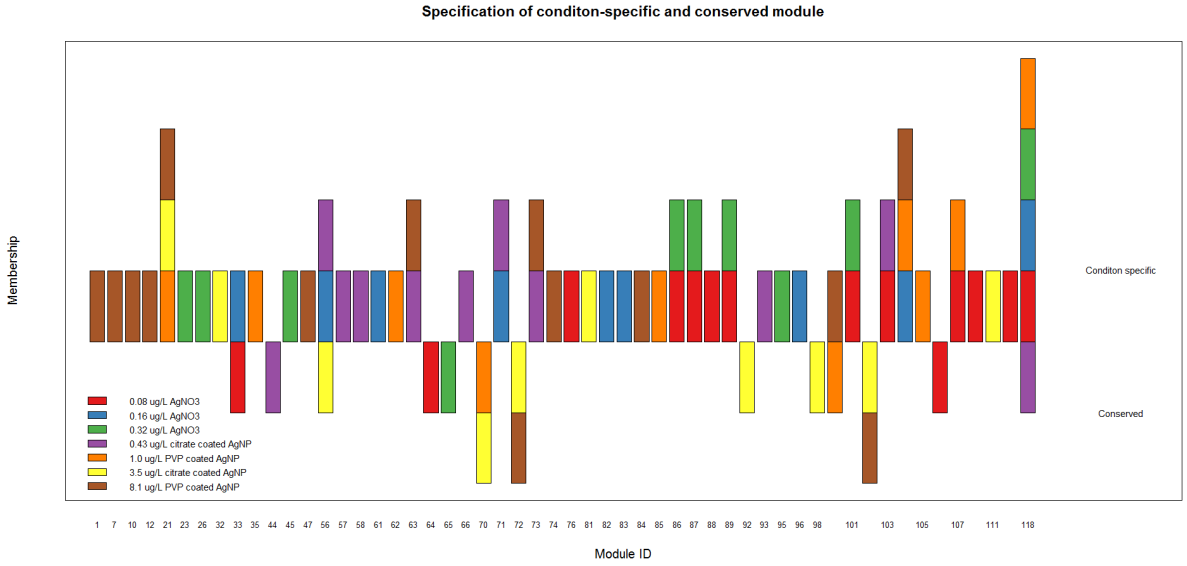


Figure 5.6: A combination bar plot of statistics of frequency about which module can be condition specific and conserved. Modules shown at the bottom are from background network, and both modules from background network and each condition-specific networks are identified by Louvain algorithm with module size constrained to be 50-200.

through Equation (5.4). There are a few condition-specific modules that show response to only one certain condition, and conserved modules also shared by AgNO3 or AgNP.

By examining up-regulated or down-regulated individual genes, [Poynton et al., 2012] suggests that AgNO3 and AgNPs exposures cause distinct changes in gene expression. Being different from individual genes study, we explore the differences at the module level, which is the motivation of network analysis. There are some consistent findings between `MODA` and [Poynton et al., 2012]. For instances, Module #1 show condition specific response to high PVP AgNP, which is enriched by ubiquitin. And [Poynton et al., 2012] concludes that AgNP exposures caused the upregulation of genes involved in pathways. Module #105 show condition specific response to low PVP AgNP and enriched by deoxyribonucleoside diphosphate metabolic process, which is consistent with AgNP exposures may have effects to protein metabolism [Poynton et al., 2012]. Proteolysis is found in AgNP condition-specific modules #1, #7 and #81, is also suggested as AgNP's effect in [Poynton et al., 2012]. As for AgNO3 exposures, processes related to energy production or developmental processes are wanted. Module #112 and #61 are enriched

with cellular growth, and both are identified as AgNO3 condition specific. Module #95 is enriched with oxidation-reduction process and is categorized as AgNO3 condition specific. In addition, Module #72 is a conserved module across two conditions: high cit (citrate-coated). AgNP and high PVP AgNP, which indicates the corresponding functions are not affected by AgNP. This module is also enriched regulation of multicellular organism growth.

We also find something beyond the consistency, mainly from condition specific modules that recognized in multiple conditions, which indicate gene expression behaviors under AgNO3 and AgNP share similar aspects. According to [Poynton et al., 2012], the lowest concentration of AgNP (1 μg/L) that was reported as not sharing any DE gene with the other exposures, while both module #104 and #118 are identified as shared across two silver nanoparticles condition-specific by `MODA`. These modules are enriched by functional gene categories identified in the original publication. e.g., proteolysis, appeared in module #118, is supposed to be caused by AgNP, and oxidation-reduction process which is affected by AgNO3, also appeared in module #118. Module #104 is also the module containing the most DE genes, as shown in Figure 5.7, where the interactions among DE are dense. Both [Poynton et al., 2012] and Table S1 reveal this module is related to nervous system development, and still, most of the gene functions remain as unknown.

5.4 Chapter summary

In this chapter, we propose the modules differential analysis (`MODA`) for weighted gene co-expression networks, given the gene expression profiles of multiple biological conditions. Specifically, we use a sample-saving approach to construct a set of condition-specific networks, which are compared with background network. We conducted the network comparison by modules detection on each network and then compare the modules. The effectiveness of `MODA` is validated on both synthetic data and real-world data. Modules identified from Daphnia magna gene expression profiles under multiple silver nanoparticles

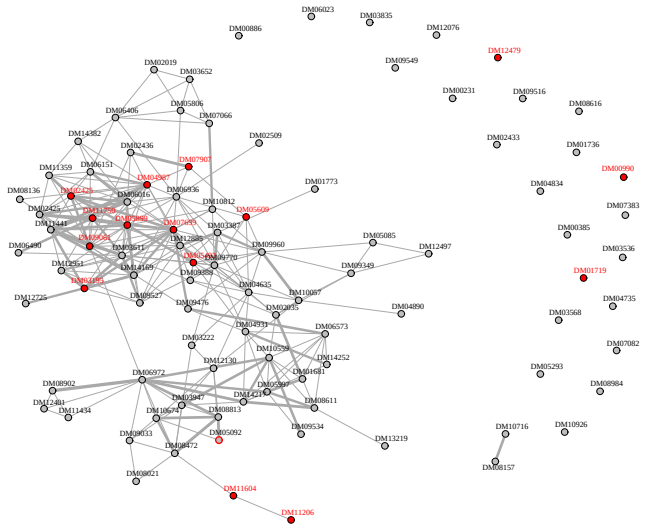


Figure 5.7: Gene co-expression network of module 104, in which the red genes are DE genes and edge width means the correlation coefficients. Edges with correlation value less than 0.7 are removed for visualization. Plotted by R igraph package.

reveal consistent conclusions with previous study, as well as some new findings. Module functions may indicate more sophisticated toxicity mode of different chemicals.

`MODA` is a general framework which focuses on modules comparison in weighted co-expression network. The current comparison only includes node set similarity, but it may be meaningful to take edges into consideration for both modules identification and comparison. Other methods in network construction and modules detection can also be included in the following development.

CHAPTER 6

CONCLUSIONS AND FURTHER WORK

This thesis provides a general framework for modeling and analyzing biological networks, focusing on active modules identification on multilayer intracellular networks. After literature reviewing of existing works on this topic in Chapter 2, the two main kinds of networks, protein-protein interactions network and gene co-expression network which has been widely used in computational systems biology, are intensively studied in Chapter 3 and 4. The main problem on such networks is modules identification, which is designed to gain insight to living mechanism at system level. Two basic optimization algorithms, meta-heuristic and continuous optimization are discussed to tackle the well-defined problems. Chapter 5 describes a package to compare different networks via modules differential analysis, which can be viewed as a following up step of modules identification on different networks. This chapter summarizes the contributions of the thesis and points out potential directions related to this topic.

6.1 Conclusions

The central problem discussed in this thesis is identifying active modules from basic biological networks and their multilayer forms. Starting with the concept of active modules, we propose several novel algorithms to mine active modules from different biological networks. Instead of summarizing the contributions according to the order they appeared in

Network	Graph	Active module requirement
PPI	Connected	Connected, with highest summed node score
Dynamic PPI	Connected	Connected, with highest summed node score
Weighted co-expression	Complete	High nodes score, high correlation
Multilayer weighted co-expression	Complete	With additional high inter-layer links

Table 6.1: Differences between active modules identification on intracellular networks discussed in this thesis.

the thesis, we summarize the contributions according to the different levels of solving the piratical problems. And we conclude the thesis by mixing works from different chapters but fitting into the following levels: formal problem definition, optimization algorithms (including implementations), multilayer extension and biological explanation.

6.1.1 Definition of active modules

Although [Ideker et al., 2002] stated the first definition of active modules in molecular interaction networks Problem 1, we may come with more specific requirements for different networks in reality. The general idea of the contribution about formally defining active modules consists of two aspects: the topological aspect and the active aspect. Both Problem 5 and Problem 4 follow the idea, based on which we define the modules on different multilayer networks. The difference between active modules on different networks depends on the characteristics of the networks themselves. Specifically, modules identification on connected graphs is more evolved with traditional graph partition theory, which poses more explicit requirement on topological structure. While modules detection on co-expression network is more like extracting a group of highly interacted nodes which also show activity. Table 6.1 highlights some specific requirements.

The first contribution of this thesis is to formally define active modules on several widely used intracellular networks and their multilayer forms, according to the characteristics and requirements in Table 6.1. These requirements are transformed into several well-defined optimization problems. Abstract as they are, kinds of existing methods in other fields become available to solve them.

6.1.2 Efficient algorithms

Optimization on the graph is generally hard due to the structural and other consideration. The optimization problem of module detection is essentially under the umbrella of combinatorics, which deals with discrete variables corresponding to the module membership in identification. We proposed several efficient algorithms to work with such problems based on two principles:

- Meta-heuristic algorithms. As a traditional technique working on combinatorial problems and discrete variables, meta-heuristic algorithms have been proven effective. We use a memetic algorithm to solve a module identification on basic PPI network in Section 2 of Chapter 3, with adaptation in terms of connectedness and module size constraint.

- Continuous relaxation. When dealing with large-scale problems, strategies in heuristics may comprise efficiency heavily. Continuous relaxation is another principle used in this thesis, including Algorithm 7 in Chapter 3 and Algorithm 10 in Chapter 4. The benefit of continuous relaxation lies at computational feasibility and efficiency. And the quality of the solution is guaranteed with some additional conditions. Thanks to the recent development in mathematical optimization, the proposed methods can be implemented easily.

Besides the two different principles, we integrate some (existing) efficient algorithms to work out a framework for network comparison in Chapter 5, which makes it more open to different users and wider application scenarios.

As in an applied multidisciplinary area, people may demand hands-on tools other than algorithms. We thus implemented these algorithms as more general-purpose and open-source packages in public platforms such as Bioconductor and GitHub. The efficient algorithms and packages can be viewed as the second contribution of this thesis.

6.1.3 Computational interpretation

After getting a module as a group of genes or proteins, the next step is biological interpretation. The most common and basic tool for this step is functional enrichment analysis, which is a computational approach that determines whether a given set of genes are significantly associated with biological functions. With enrichment analysis, we successfully relate the identified modules with biological meanings, which are consistent with experiments settings in the real-world examples in Chapter 3, 4, and 5.

In addition to basic enrichment analysis, we also establish several new computational pipelines in the interpretation step. These pipelines are much inspired by the Disease module DREAM Challenge we participated in 2016, which required to relate identified module with knowing disease. As an open challenge, the submitted modules are evaluated primarily based on the number of discovered modules that are significantly associated with complex traits and diseases [Organizers, 2016]. Specifically, it counts the number of modules with a significant enrichment p-value in at least one GWAS at a given false discovery rate (FDR) cutoff. We adopt the same idea and derive a pipeline for more widely used biological processes and KEGG pathways in Chapter 3 and Chapter 4, which is able to evaluate the proposed methods statistically. And these pipelines are reusable in other applications as well, aiming to reduce the dependence on domain knowledge as much as possible. We thus regard the efforts in biological interpretation through a (pure) computational approach as the third contribution of this thesis.

6.2 Limitations

Although the thesis contributes the field of network biology by defining several problems and developing efficient algorithms to solve them, there exists limitations in the study. We mainly discuss the technical limitations of the proposed algorithms in this section as some of the conceptual limitations will be mentioned in the next section.

6.2.1 Limitations of active modules on PPI network

Chapter 3 proposes two different algorithms for identifying active modules on PPI networks. Each inherits the limitations from their superclasses. Specifically, the memetic algorithm (Algorithm 4) provides no guarantee on the solution in finite running time. And the overhead is linear to the population size which makes it expensive when larger population is needed. The convex optimization approach (Algorithm 7) relaxes the original 0-1 programming into the continuous space, which makes it hard to estimate the gap between the solution in the relaxed space and the optimum.

Another shared weak point of these two algorithms compared with those in Chapter 4 and Chapter 5 is that they were not implemented as standard packages, which reduces the chances for other people to use.

6.2.2 Limitations of `AMOUNTAIN`

Chapter 4 describes `AMOUNTAIN`, a unified framework for identifying active modules on weighted gene co-expression networks. Algorithm 9 is shown to be robust to the input WGCNs and not sensitive to the parameters, but how to pick these parameters can be a problem. A crucial decision which will affect the performance is the size of the active modules. In our current implementation of `AMOUNTAIN`, the users need to determine the module size according to the preference or prior knowledge. It might be helpful to determine the module size rigorously and automatically.

From the input perspective, the current algorithm cannot deal with general multilayer networks, i.e., networks of networks with arbitrary inter-layer interactions. The compact form of matrix computation in the multilayer network case makes it easier to implement, but also make it harder to handle more general networks.

6.2.3 Limitations of MDOA

Chapter 5 describes MODA, a Bioconductor R package to conduct modules differential analysis, mainly for WGCNs. The package provides several modules detection algorithms, and there are numerous more algorithms can be integrated but no one beats all the rest for all data. How to pick a proper detection algorithm for given input data is to be explored.

Another limitation lies at modules comparison stage. Currently, MODA only uses Jaccard index to measure the similarity between two modules (subnetworks), which provides limited information on the structural properties. Furthermore, the network edges properties are ignore by this comparison, which might help to reveal important mechanism of these subnetworks.

6.3 Future Work

There are few related works which can be considered in the future, and potentially improve the works in the thesis.

6.3.1 Overlapping and hierarchical modules detection

The active modules identification methods on single layer networks described in the thesis are closely related to non-overlapping graph partitioning. Even the primary objective in Chapter 3 is derived from a graph partitioning perspective, which makes the proposed algorithm not able to deal with overlapping complexes detection. While overlapping and hierarchical community structure has been discussed in general networks [Lancichinetti et al., 2009]. Naturally, the existence of overlapping and hierarchical modules in biological networks are also possible [Nepusz et al., 2012; Ou-Yang et al., 2014].

Overlapping complexes in PPI networks are considered to have advantages over non-overlapping complexes in terms of certain explanation, especially when several important proteins may get involved in multiple complexes in a disease process. And the hierarchy

of biological functions has been seen in Gene Ontology [Maere et al., 2005; Huang et al., 2008], which can be captured by hierarchical modules. How to derive efficient algorithms to detect overlapping and hierarchical modules is thus important, and potentially improve the works described in this thesis, in terms of both empirical evaluation and biological explanation.

6.3.2 Heterogenous multiple layers

The multilayer networks discussed in Chapter 3 and Chapter 4 are the natural extension of single layer networks, especially the multilayer dynamic networks, in which different layers share the same set of vertices and probably similar structure. Even in the cross-species network Chapter 4, both the two layers represent the gene levels of two species, with the orthologous mapping between them. We can call such multilayer networks **homogeneous** systems. While in reality, there is also a need to model **heterogenous** multilayer networks, in which different layers can be various, such as protein layer and metabolic layer interact in cell. As a more specific example, [Rieckmann et al., 2017] describes the immune system as a multilayer social network, where the layers represent immune cell types, organs, and tissues, etc. Compared with multilayer dynamic PPI or cross-species networks, the inter-layer links are more general in this case, which cannot be processed with algorithms in Chapter 4 currently.

In theory, the existing method can be used with a layer-by-layer or neighbour-by-neighbour strategy, but it is limited to find global conserved modules. That partly explains why a unified framework is preferred. As a more broadly model of multilayer networks, heterogenous layers with arbitrary inter-layer interactions requires new theories and algorithms.

6.3.3 High-order graph features

The active module in this thesis is defined as a group of node with 1) high summed node weights, and 2) high edge weights or significantly more edge interactions. The idea is applied in both connected graphs in Chapter 3 and complete graphs in Chapter 4, which are characterized by direct interactions between vertices. In other words, the active module identification only leverages the **first-order** information of graphs in the thesis. But recent advances in graph embedding [Goyal and Ferrara, 2017] suggests that the **second-order** or even higher-order information other than direct connections on graphs may help to derive more robust and precise methods.

As an instance in computational biology, modeling high-order biological networks or extracting high-order features from existing networks may shed some light on unsolved problems in this field, or improve current methods in terms of accuracy and robustness.

6.3.4 General networks comparison

As a following-up operation to active modules identification, Chapter 5 describes `MODA`, a network comparison method via modules differential analysis. Currently, it only supports weighted gene co-expression networks due to the close connection with popular tool `WGCNA`. But more general network comparison or network alignment is also of interest and attracts much attention in computational biology [Kuchaiev and Pržulj, 2011]. Current works on networks comparison tend to take the whole network as an individual and focus on global properties. But it may lead to limited information when comparing large-scale networks. The idea of `MODA` can be incorporated with existing network comparison methods from two aspects: 1) Applying `MODA` to more general networks other than weight gene co-expression networks and taking the whole network as an individual, but conducting modules identification at the same time. 2) Applying more existing network comparison algorithms or graph similarity metrics when comparing modules in `MODA`, which is supposed to provide more information other than Jaccard index.

APPENDIX A

A BRIEF INTRODUCTION TO SEVERAL ALGORITHMS

Chapter 2 reviews several representative algorithms for active modules identification, which can be categorized into two distinct classes: meta-heuristic methods and exact approaches. Here we give a brief introduction of each representative algorithm belonging to these two classes at a basic level.

A.1 Simulated annealing

Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function [Wikipedia, d]. The optimization procedure is inspired by annealing in metallurgy, which names the algorithm. Proposed in 1979 and developed in 1980s [Kirkpatrick et al., 1983], SA has been a widely used in numerous applications.

The basic idea of SA is to explore the solution space by slowly decreasing in the probability of accepting worse solutions. Being similar physical systems, the goal is to achieve a state when the internal energy is minimized. This task is accomplished by the following components: an initial state, the neighbors of a state, acceptance probabilities and the annealing procedure. The corresponding iterative optimization details can vary from one application to another, but the terminologies can be roughly matched in the following Table A.1:

Table A.1: How simulated annealing is inspired by annealing in metallurgy (* or achieves maximal iteration)

	Annealing	Optimization
Goal	Minimal internal energy	Minimal objective
Initialization	Initial state	Initial solution
New solution	Neighbours of a state	Strategies
Keep new solution or not	Acceptance probabilities	Parameters
Procedure	Annealing	Iteration
Stopping condition*	Energy zero	Convergence

A more general case of simulated annealing other than Algorithm 1 used in active modules identification is described as Algorithm 12.

Algorithm 12: General framework of simulated annealing

Input: The maximal iterations number K_{max}, Temperature, Neighbour and Probability functions
Output: The final state s.

Initialization: $s = s_0$;
for k = *1,2,..*K_{max} **do**
  $T \leftarrow$ Temperature(k/K_{max});
  Pick a random neighbour, $s_{new} \leftarrow$ Neighbour(s);
  if *Probability(Energy(s),Energy(*s_{new}*),T)* $\geq$ *random(0,1)* **then**
    $s \leftarrow s_{new}$;
end
Output s

A.2 Genetic algorithm

Genetic algorithm (GA) is a metaheuristic inspired by the process of natural selection, which also belongs to and remains as the foundation to the larger class of algorithms: evolutionary algorithms (EA). These algorithms have been commonly used in optimization and searching problems, and employed in a wide range of applications.

The basic idea of GA is to use of a population of candidate solutions evolving towards to a better solution to an optimization problem. The quality of a solution is evaluated by a fitness function, to maximize (or minimize) which is also the goal the GA want to achieve. This task is accomplished by the following components: an initial group of solutions, the

Table A.2: How genetic algorithm is inspired by natural selection

	Natural selection	Optimization algorithm
Goal	Survival	Minimal objective
Initialization	Wild population	Generated individuals
New solution	Variation	Mutation
	Differential reproduction	Crossover
	Heredity	Selection
Fitness	Potential to survival	Objective function
Procedure	Natural choice	Evolving
Stopping condition	None	Convergence or max iter

genetic operators (crossover, mutation and selection), and the evolving procedure. Being similar to Table A.1, the terminologies of GA and optimization can be roughly matched in the following Table A.2:

As the genetic operators (mainly the crossover) can be conducted on a binary represented individual, the binary encoding has naturally become the most common choice. A more general case of genetic algorithm (also referred as simple genetic algorithm) is described as Algorithm 13.

Algorithm 13: Simple genetic algorithm

Input: The maximal iterations number K_max, Temperature, Neighbour and Probability functions
Output: The final state s.

Initialization: $s = s_0$;
for k = *1,2,..*K_{max} **do**
  $T \leftarrow$ Temperature(k/K_{max});
  Pick a random neighbour, $s_{new} \leftarrow$ Neighbour(s);
  if *Probability(Energy(s),Energy(*s_{new}*),T)* $\geq$ *random(0,1)* **then**
    $s \leftarrow s_{new}$;
end
Output s

A.3 Linear programming

Linear programming (LP) [Wikipedia, c] is an optimization method dealing with linear models. Specifically, LP is for the optimization of a linear objective subject to linear

constraints. The standard form of an LP is expressed as

$$\begin{aligned} &\text{maximize} && \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to} && A\mathbf{x} \leq \mathbf{b}, \\ &\text{and} && \mathbf{x} \geq \mathbf{0}, \end{aligned} \tag{A.1}$$

where $\mathbf{c}$, $\mathbf{b}$ are given vectors and A is known matrix.

As the linear function is convex and when the linear constraints define the convex and bounded polyhedron, every local optimum of an LP is also the global optimum. Two popular algorithms have been used to solve LPs: the simplex algorithm [Dantzig, 1947] and the interior point method [Karmarkar, 1984]. Both algorithms are efficient for general LPs while the interior point method enjoys a lower worst-case complexity [Karmarkar, 1984]. Along with many other variants and improvements, these algorithms have been included in many different open-source or commercial solvers. Therefore LP is generally a mature technique.

In Chapter 2, we have also seen the integer linear programming (ILP) [Wikipedia, b] which takes integer variables and linear objective or constraints. The standard form of ILP is expressed as

$$\begin{aligned} &\text{maximize} && \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to} && A\mathbf{x} \leq \mathbf{b}, \\ & && \mathbf{x} \geq \mathbf{0}, \\ &\text{and} && \mathbf{x} \in \mathbb{Z}^n, \end{aligned} \tag{A.2}$$

where $\mathbf{c}$, $\mathbf{b}$ are given integer vectors and A is matrix. If some entries in $\mathbf{x}$ are not integers, the problem is also referred as mixed integer linear programming (MILP).

Integer programming is proven to be NP-hard by reducing it to a VERTEX COVER problem, one of the Karp's 21 NP-complete problems [Karp, 1972]. Various approximation algorithms have been developed. Just some problems to be solved but represented by ILP, the ILP itself is solved by two categories of algorithms:

- Exact algorithms. Linear relaxation method relaxes the integer variables into real values as linear programming (LP) is a relatively easy to solve. Then we can round the entries of a real solution vector to get the integer solution.

- Heuristic methods. The ILP is NP-hard which means the problem is intractable thus heuristic can be used instead. Methods like tabu search and simulated annealing have been applied to ILP.

As an example in active modules identification which applys exact algorithms to solve ILP, `heinz` [Dittrich et al., 2008] uses an existing exact algorithm [Ljubić et al., 2006] to solve the Problem 2.

APPENDIX B

SUPPLEMENTARY TO CHAPTER 3

B.1 Convergence analysis of Algorithm 7

Here is the detailed Convergence analysis of Algorithm 7 in Chapter 3, which aims to solve the following constrained quadratic programming problem:

$$\text{minimize } f = \frac{1}{2}\mathbf{x}^\top L\mathbf{x} + \lambda \mathbf{z}^\top \mathbf{x} \quad \text{subject to} \quad \mathbf{x} \in \Omega, \tag{B.1}$$

where the graph Laplacian $L = D - W$, and D is the diagonal degree matrix, in which $D_{ii} = d_i = \sum_j w_{ij}$. And $\mathbf{z} \in \mathbb{R}^n$ is the vertices weight, $\Omega = \{\mathbf{x} \ : \ \mathbf{x}^\top \mathbf{1} = 0, \ \mathbf{x}^\top \mathbf{x} = n\}$ is the feasible region.

Proposition B.1.1. (Proposition 1 of Von Luxburg [2007]) L is symmetric and positive semi-definite.

Proof. By the definition, given any $\mathbf{y} \in \mathbb{R}^{n\times n}$, we have

$$\begin{aligned}
\mathbf{y}^\top L\mathbf{y} &= \mathbf{y}^\top D\mathbf{y} - \mathbf{y}^\top W\mathbf{y} = \sum_{i=1}^{n} d_i y_i^2 - \sum_{i,j=1}^{n} y_i y_j w_{ij} \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} d_i y_i^2 - 2\sum_{i,j=1}^{n} y_i y_j w_{ij} + \sum_{i=j}^{n} d_j y_j^2\right) \\
&= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(y_i - y_j)^2 \geq 0.
\end{aligned} \tag{B.2}$$

□

Since L is positive semi-definite, the quadratic programming problem (B.1) is convex [Boyd and Vandenberghe, 2004]. We use gradient based method to solve it. Specifically, we use the following projected gradient to make the solution located in the feasible region.

$$\mathbf{x}^{(k+1)} = P(\mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x}^{(k)})), \tag{B.3}$$

where $\alpha^{(k)} > 0$ is the step size at k-th iteration and $P(\cdot)$ is the projection

$$P(\mathbf{g}) = \arg\min\{\|\mathbf{x} - \mathbf{g}\| : \mathbf{x} \in \Omega\}. \tag{B.4}$$

which projects the vector $\mathbf{g}$ onto the set $\Omega \subset \mathbb{R}^n$.

For problem (B.1), we consider the second constraint $\mathbf{x}^\top \mathbf{x} \leq 1$ is equivalent to $\mathbf{x}^\top \mathbf{x} = 1$, since the optima occurs at the boundary. More importantly, the former constraint makes it a convex set which brings optimization properties, thought we use the latter in practical computation. In summary, the projection is conducted by orthogonalization and normalization on the gradient of f at the k-th iteration:

$$\begin{aligned} \mathbf{y} &= \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x}^{(k)}) \\ \mathbf{z} &= \mathbf{1}^\top \mathbf{1} \cdot \mathbf{y} - \mathbf{y}^\top \mathbf{1} \cdot \mathbf{1} \\ \mathbf{x}^{(k+1)} &= \sqrt{n} \cdot \frac{\mathbf{z}}{\|\mathbf{z}\|_2} \end{aligned} \tag{B.5}$$

The two-step projection is shown as Figure B.1. Both the orthogonalization and normalization steps are straightforward and efficient.

For general purpose, we can also write (B.3) into the function w.r.t. α:

$$\mathbf{x}^{(k)}(\alpha) = P(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)})), \tag{B.6}$$

[Bertsekas, 1976] proposed the first practical procedure called Armijo rule to determine

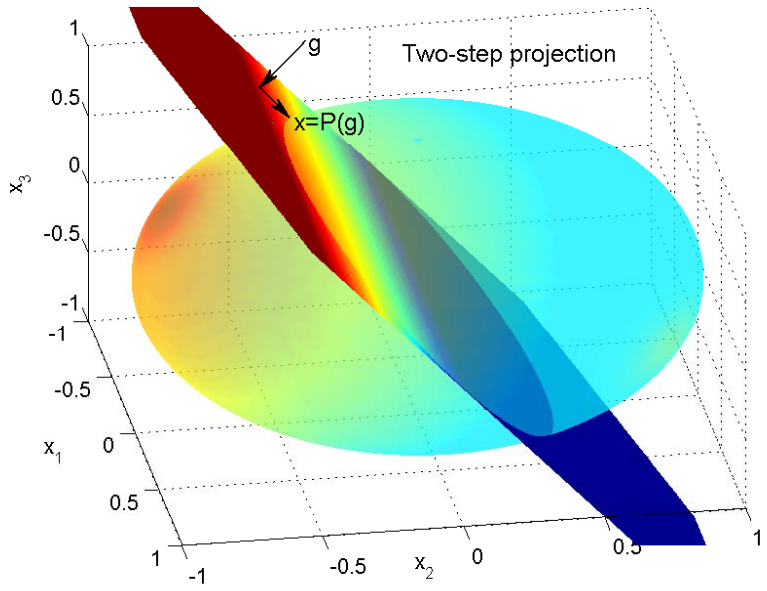


Figure B.1: The Euclidean projection of a vector $\mathbf{g} \in \mathbb{R}^n$ onto the feasible region $\Omega = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{x}^\top \mathbf{1} = 0, \ \|\mathbf{x}\|_2^2 \leq 1\}$, which is a intersection of a hyperplane and a ℓ_2-ball, and the boundary of ℓ_2-ball in the $(n-1)$-dimensional space.

the step size: given $0 < \beta < 1$, $0 < \gamma$ and $0 < \mu < 1$, pick $\alpha^{(k)}$ via:

$$\alpha^{(k)} = \beta^{m_k} \gamma, \tag{B.7}$$

where m_k is the smallest nonnegative integer such that

$$f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)}) + \mu(\nabla f(\mathbf{x}^{(k)}), \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) \tag{B.8}$$

[Calamai and Moré, 1987] further to propose the step size should satisfy (B.7) and given positive constants γ_1, γ_2, μ_2 such that,

$$\alpha^{(k)} \geq \gamma_1 \ or \ \alpha^{(k)} \geq \gamma_2 \bar{\alpha} \tag{B.9}$$

where $\bar{\alpha}$ satisfies

$$f(\mathbf{x}^{(k)}(\bar{\alpha})) > f(\mathbf{x}^{(k)}) + \mu_2(\nabla f(\mathbf{x}^{(k)}), \mathbf{x}^{(k)}(\bar{\alpha}) - \mathbf{x}^{(k)}) \tag{B.10}$$

Based on these conditions, [Calamai and Moré, 1987] generalized the existing theories by then and proposed the theorem about convergence properties of projected gradient method:

Theorem B.1.2. (Theorem 2.4 of Calamai and Moré [1987]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously ditterentiable on Ω, and $\mathbf{x}^{(k)}$ the sequence $\{\mathbf{x}^{(k)}\}$ generated by (B.3) and (B.9). If some sequence $\{\mathbf{x}^{(k)} : k \in K\}$ is bounded then

$$\lim_{k \in K, k \to \infty} \frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\alpha^{(k)}} = 0. \tag{B.11}$$

Moreover, any limit point of $\{\mathbf{x}^{(k)}\}$ is a stationary point of (B.1).

As stated above, problem (B.1) is convex because of L, a local optimum is also the global optimum. The effectiveness of Algorithm 7 is thus guaranteed.

B.2 Supplementary modules and GO terms

Supplementary modules and GO terms are stored at `http://www.cs.bham.ac.uk/~dxl466/st/ModuleExtraction.zip`.

- "Algorithm2.txt": modules identified by Algorithm 2 on consensus graph.
- "PPI_complexesx.txt": modules identified by Algorithm 2 on time-x layer. graph.
- "spici.txt": modules identified by SPICi Jiang and Singh [2010] on the static network.
- "spici-c.txt": modules identified by SPICi on the consensus graph.
- "Table SS1.xlsx" Enriched GO terms by Algorithm 2 on consensus graph.
- "Table SS2.xlsx" Enriched GO terms by SPICi on the static network.

APPENDIX C

SUPPLEMENTARY TO CHAPTER 4

C.1 NP-hardness of problem (5)

The graph in Problem 5 can be summarized as a complete network with vertices weights and edges weights. The NP-hardness of Problem 5 can be proved by considering finding heaviest subnetworks on the following four simplified cases:

- 1), Complete network with identical vertices weights but non-identical edges weights.
- 2), Incomplete network with identical vertices weights and edges weights.
- 3), Incomplete network with identical vertices weights but non-identical edge weights.
- 4), Incomplete network with identical edges weights but non-identical node weights.

Theorem C.1.1. Searching for heaviest module on all the cases above are NP-hard, thus problem (5) is NP-hard.

Proof. A more simplified version of case 1) corresponds to a k-maximal spanning tree problem, which asks for a tree covers exactly k nodes with maximal edge weights. Here we ask for a complete subgraph instead of a subtree. The NP-hardness of edge-weighted k-cardinality tree problem has been proven in [Fischetti et al., 1994] and [Woeginger, 1992].

Case 2) can be viewed as a special case of case 3) which is also can be reduced to the well-known NP-complete problem k-clique. The NP-hardness proof is available in the Supplementary Text S1 of [Li et al., 2011].

Case 4) corresponds to the Maximum-Weight Connected Subgraph Problem (MWCSP). The NP-hardness can be proven by reducing the well-known NP-complete problem MINIMUM COVER to the problem. The proof sketch is available in Supplementary information of [Ideker et al., 2002].

Finding heaviest subnetworks on all above different networks are NP-hard and special case of Problem 5, which means problem (5) is also NP-hard. □

For completeness, finding a heaviest subgraph on a complete graph with identical edges weights but non-identical vertices weights is a trivial task. Just sorting the vertices weights and picking the highest k ones is not NP-hard, thus we did not list this case above.

C.2 Piecewise root finding for Euclidean projection on elastic net

Here are the detailed steps to solve the following problem:

$$P_C(\mathbf{y}) = \arg\min_{\mathbf{x}\in\mathbb{R}^n_+} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 \; s.t. \; \alpha\|\mathbf{x}\|_1 + (1-\alpha)\|\mathbf{x}\|_2^2 = 1. \tag{C.1}$$

We have the optimal solution of (C.1) as follows:

$$x_i^* = \max\left(0, \frac{y_i - \alpha\theta^*}{1 + 2(1-\alpha)\theta^*}\right), \tag{C.2}$$

where θ^* is the optimal lagrange multiplier.

The problem turns out to find a proper θ^* to satisfy $\alpha\|\mathbf{x}^*\|_1 + (1-\alpha)\|\mathbf{x}^*\|_2^2 = t$. Using the same strategy of Gong et al. [Gong et al., 2011], we formulate the problem of Euclidean projections on the elastic net constraint set as a root finding problem. Being

slightly different from that in [Gong et al., 2011], we use the original form of elastic net penalty, i.e $f(\mathbf{x}) = \alpha\|\mathbf{x}\|_1 + (1-\alpha)\|\mathbf{x}\|_2^2$, which makes it easy to control the module size by tuning the parameter α: the larger α leads to a smaller module.

$$\sum_{i=1}^{n}\left(\alpha \max\left(0, \frac{y_i - \alpha\theta}{1+2(1-\alpha)\theta}\right) + (1-\alpha)\frac{\max(0, y_i-\alpha\theta)^2}{(1+2(1-\alpha)\theta)^2}\right) = t. \tag{C.3}$$

The solution θ is then transformed to the root of following function,

$$\begin{aligned} &f_{en}(\theta) = t(1+2(1-\alpha)\theta)^2 - \\ &\sum_{i=1}^{n}\left(\alpha(1+2(1-\alpha)\theta)\max(0, y_i-\alpha\theta) + (1-\alpha)\max(0, y_i-\alpha\theta)^2\right) \\ &= a_\theta\theta^2 + b_\theta\theta + c_\theta, \end{aligned} \tag{C.4}$$

where $a_\theta = (\alpha^3-\alpha^2)|R_{\mathbf{y},\theta}| - 4t(1-\alpha)^2$, $b_\theta = -\alpha^2|R_{\mathbf{y},\theta}| - 4t(1-\alpha)$ and $c_\theta = \alpha\sum_{i\in R_{\mathbf{y},\theta}} y_i + (1-\alpha)\sum_{i\in R_{\mathbf{y},\theta}} y_i^2 - t$, and $R_{\mathbf{y},\theta} = \{i | y_i \geq \theta\}$. $|R_{\mathbf{y},\theta}|$ is the cardinality of the set. The root finding procedure of (C.4) can be achieved by the following sequence $\{\theta_k\}$:

$$\theta_k = \frac{-b_{\theta_{k-1}} + \sqrt{b_{\theta_{k-1}}^2 - 4a_{\theta_{k-1}}c_{\theta_{k-1}}}}{2a_{\theta_{k-1}}}. \tag{C.5}$$

The algorithm for problem (C.1) is summarized as algorithm 14.

Algorithm 14: Euclidean projections on elastic net

Input: $\mathbf{y} \in \mathbb{R}^n$, $0 < \alpha < 1$, $t > 1$ and $0 < \theta_0 < y_n$
Output: $\mathbf{x}$

```
while not reach maximal iterations or f_en(θ_k) == 0 do
    Update θ by;
    (C.5) set x according to (C.2);
end
```

For the optimization procedure in a two-layer network, each single layer module identification share the same algorithm as we adopt the alternating optimization strategy.

C.3 Comparison with non-convex optimization

C.3.1 Modified algorithm on single layer WGCN

As stated in the main text, the method we propose is inspired by [Li et al., 2011]. In order to compare it with our algorithm, we can also use the mixed norm $\ell_{0,\infty}(\mathbf{x}) = \alpha\|\mathbf{x}\|_0 + (1-\alpha)\|\mathbf{x}\|_\infty$ $(0 < \alpha < 1)$ to encode the characteristics of gene membership. In practice $\ell_{0,\infty}$ was approximated by $\ell_{p,2}(\mathbf{x}) = \alpha\|\mathbf{x}\|_p + (1-\alpha)\|\mathbf{x}\|_2$ $(0 < p < 1)$, and [Li et al., 2011] solved the non-convex problem with only edge weights by Multi-Stage Convex Relaxation (MSCR) [Zhang, 2010]. Here we adopt the similar strategy, the solution of problem (2) in the main text is a stationary point of the following optimization problem:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathbb{R}_+^n} \ \mathbf{x}^T W\mathbf{x} + \lambda\mathbf{z}^T\mathbf{x} - \mu f(\mathbf{x}), \tag{C.6}$$

where μ is the lagrange multiplier.

Following the analysis in [Zhang, 2010], let $\mathbf{h}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ be a specific convex vector function, such as $h(x) = x^h$ $(h \geq 1)$, and we assume there exists a concave function $\bar{f}_{\mathbf{h}}(\mathbf{u})$ such that $f(\mathbf{x}) = \bar{f}_{\mathbf{h}}(\mathbf{h}(\mathbf{x}))$ holds. The regularization function can be rewritten as

$$f(\mathbf{x}) = \inf_{\mathbf{v}\in\mathbb{R}^n}[\mathbf{v}^T\mathbf{h}(\mathbf{x}) - f_{\mathbf{h}}^*(\mathbf{v})]. \tag{C.7}$$

using concave duality, function $f_{\mathbf{h}}^*(\mathbf{v})$ is given by

$$f_{\mathbf{h}}^*(\mathbf{v}) = \inf_{\mathbf{u}\in\mathbb{R}^n}[\mathbf{v}^T\mathbf{u} - \bar{f}_{\mathbf{h}}(\mathbf{u})]. \tag{C.8}$$

Thus Lagrangian (C.6) can be rewritten as

$$\left[\hat{\mathbf{x}}, \hat{\mathbf{v}}\right] = \arg\max_{\mathbf{x},\mathbf{v}} \ \mathbf{x}^T W\mathbf{x} + \lambda\mathbf{z}^T\mathbf{x} - \mu\mathbf{v}^T h(\mathbf{x}) + \mu f_h^*(\mathbf{v}) \tag{C.9}$$

We solve (C.9) by an alternative optimization method, i.e. each time fix one variable

and optimize another until convergence. When fixing $\mathbf{v}$, the entries of $\mathbf{x}$ are given by (C.10) according to gradient of (C.9). Here we enforce $f(\mathbf{x}) = 1$ by normalization, thus avoid tuning lagrange multiplier μ.

$$\hat{x}_i = \left[\frac{2x_i \sum_j w_{ij}x_j + \lambda x_i z_i}{h\hat{v}_i}\right]^{\frac{1}{h}} \tag{C.10}$$

When fixing $\mathbf{x}$, the solution of $\mathbf{v}$ is determined by the gradient of $\bar{f}_{\mathbf{h}}(\mathbf{u})$, which is equal to $\nabla_{\mathbf{x}} f(\mathbf{x})/\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x})$ since $f(\mathbf{x}) = \bar{f}_{\mathbf{h}}(\mathbf{h}(\mathbf{x}))$ and the chain rules, thus the solution is

$$\hat{v}_i = \frac{\alpha}{h}(\sum_j |x_j|^p)^{\frac{1}{p}-1}|x_i|^{p-h} + \frac{1-\alpha}{h}(\sum_j x_j^2)^{\frac{1}{2}-1}|x_i|^{2-h} \tag{C.11}$$

The algorithm is summarized as

Algorithm 15: Alternating optimization for problem (C.6) using MSCR

Input: Co-expression network edge weight $W \in \mathbb{R}^{n\times n}$, node score $\mathbf{z} \in \mathbb{R}^n$ and initial solution $\mathbf{x}^{(0)} \in \mathbb{R}^n$ which is sampled from uniform distribution [0,1].

Output: Module indicator vector $\mathbf{x}$

```
Initialize v̂_j = 1;
while Convergence or reach maximal iterations do
    Update x̂ by (C.10), then x is normalized by x_i ← x_i / (α‖x‖_p + (1−α)‖x‖_1);
    Update v̂ by (C.11);
end
```

C.4 Supplementary Tables

Supplementary Tables SS1-6 mentioned in Chapter 4 are available online. See `http://www.cs.bham.ac.uk/~dxl466/st/SupplementaryFiles_AMOUNTAIN.zip`.

APPENDIX

LIST OF REFERENCES

Ahn, Y.-Y., Bagrow, J. P., and Lehmann, S. (2010). Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764.

Amar, D., Safer, H., and Shamir, R. (2013). Dissection of regulatory networks that are altered in disease via differential co-expression. *PLoS Comput Biol*, 9(3):e1002955.

Aoki, K., Ogata, Y., and Shibata, D. (2007). Approaches for extracting practical information from gene co-expression networks in plant biology. *Plant and Cell Physiology*, 48(3):381–390.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.

Assenov, Y., Ramírez, F., Schelhorn, S.-E., Lengauer, T., and Albrecht, M. (2008). Computing topological parameters of biological networks. *Bioinformatics*, 24(2):282–284.

Backes, C., Rurainski, A., Klau, G. W., Müller, O., Stöckel, D., Gerasch, A., Küntzer, J., Maisel, D., Ludwig, N., Hein, M., et al. (2012). An integer linear programming approach for finding deregulated subgraphs in regulatory networks. *Nucleic acids research*, 40(6):e43–e43.

Ballouz, S., Verleyen, W., and Gillis, J. (2015). Guidance for rna-seq co-expression network construction and analysis: safety in numbers. *Bioinformatics*, 31(13):2123–2130.

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

Barabási, A.-L., Gulbahce, N., and Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68.

Barabasi, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113.

Battiston, F., Nicosia, V., and Latora, V. (2014). Structural measures for multiplex networks. *Physical Review E*, 89(3):032804.

Beisser, D., Brunkhorst, S., Dandekar, T., Klau, G. W., Dittrich, M. T., and Müller, T. (2012). Robustness and accuracy of functional modules in integrated network analysis. *Bioinformatics*, 28(14):1887–1894.

Beisser, D., Klau, G. W., Dandekar, T., Müller, T., and Dittrich, M. T. (2010). Bionet: an r-package for the functional analysis of biological networks. *Bioinformatics*, 26(8):1129–1130.

Benjamini, Y. and Hochberg, Y. (1995). 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.

Bertsekas, D. (1976). On the goldstein-levitin-polyak gradient projection method. *IEEE Transactions on automatic control*, 21(2):174–184.

Bhalla, U. S. and Iyengar, R. (1999). Emergent properties of networks of biological signaling pathways. *Science*, 283(5400):381–387.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C. I., Gómez-Gardenes, J., Romance, M., Sendina-Nadal, I., Wang, Z., and Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Boyle, E. I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M., and Sherlock, G. (2004). Go:: Termfinder, an open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2007). On finding graph clusterings with maximum modularity. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 121–132. Springer.

Brazma, A. and Vilo, J. (2000). Gene expression data analysis. *FEBS letters*, 480(1):17–24.

Brummelman, J., Raeven, R. H., Helm, K., Pennings, J. L., Metz, B., van Eden, W., van Els, C. A., and Han, W. G. (2016). Transcriptome signature for dampened th2 dominance in acellular pertussis vaccine-induced cd4+ t cell responses through tlr4 ligation. *Scientific reports*, 6.

Bryant, W. A., Sternberg, M. J., and Pinney, J. W. (2013). Ambient: active modules for bipartite networks-using high-throughput transcriptomic data to dissect metabolic response. *BMC systems biology*, 7(1):26.

Burer, S. and Letchford, A. N. (2009). On nonconvex quadratic programming with box constraints. *SIAM Journal on Optimization*, 20(2):1073–1089.

Calamai, P. H. and Moré, J. J. (1987). Projected gradient methods for linearly constrained problems. *Mathematical programming*, 39(1):93–116.

Carchiolo, V., Longheu, A., Malgeri, M., and Mangioni, G. (2011). Communities unfolding in multislice networks. In *Complex Networks*, pages 187–195. Springer.

Chang, S. H., Chung, Y., and Dong, C. (2010). Vitamin d suppresses th17 cytokine production by inducing c/ebp homologous protein (chop) expression. *Journal of Biological Chemistry*, 285(50):38751–38755.

Chatr-Aryamontri, A., Breitkreutz, B.-J., Oughtred, R., Boucher, L., Heinicke, S., Chen, D., Stark, C., Breitkreutz, A., Kolas, N., O'donnell, L., et al. (2015). The biogrid interaction database: 2015 update. *Nucleic acids research*, 43(D1):D470–D478.

Chatr-aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N. K., O'Donnell, L., Oster, S., Theesfeld, C., Sellam, A., et al. (2016). The biogrid interaction database: 2017 update. *Nucleic Acids Research*, page gkw1102.

Chen, B., Fan, W., Liu, J., and Wu, F.-X. (2014). Identifying protein complexes and functional modules from static ppi networks to dynamic ppi networks. *Briefings in bioinformatics*, 15(2):177–194.

Chen, E. Y., Tan, C. M., Kou, Y., Duan, Q., Wang, Z., Meirelles, G. V., Clark, N. R., and Maayan, A. (2013). Enrichr: interactive and collaborative html5 gene list enrichment analysis tool. *BMC bioinformatics*, 14(1):128.

Chen, J. and Yuan, B. (2006). Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, 22(18):2283–2290.

Chen, W., Liu, J., and He, S. (2017). Prior knowledge guided active modules identification: an integrated multi-objective approach. *BMC systems biology*, 11(2):8.

Choi, Y. and Kendziorski, C. (2009). Statistical methods for gene set co-expression analysis. *Bioinformatics*, 25(21):2780–2786.

Chuang, H.-Y., Lee, E., Liu, Y.-T., Lee, D., and Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular systems biology*, 3(1):140.

Ciofani, M., Madar, A., Galan, C., Sellars, M., Mace, K., Pauli, F., Agarwal, A., Huang, W., Parkurst, C. N., Muratet, M., et al. (2012). A validated regulatory network for th17 cell specification. *Cell*, 151(2):289–303.

Cleret-Buhot, A., Zhang, Y., Planas, D., Goulet, J.-P., Monteiro, P., Gosselin, A., Wacleche, V. S., Tremblay, C. L., Jenabian, M.-A., Routy, J.-P., et al. (2015). Identification of novel hiv-1 dependency factors in primary ccr4+ ccr6+ th17 cells via a genome-wide transcriptional approach. *Retrovirology*, 12(1):1.

Conesa, A., Götz, S., García-Gómez, J. M., Terol, J., Talón, M., and Robles, M. (2005). Blast2go: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, 21(18):3674–3676.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5):1–9.

Dantzig, G. B. (1947). Maximization of a linear function of variables subject to linear inequalities. *Activity Analysis of Production and Allocation*, pages 339–347.

De Domenico, M., Lancichinetti, A., Arenas, A., and Rosvall, M. (2015). Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, 5(1):011027.

De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M. A., Gómez, S., and Arenas, A. (2013). Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022.

De Las Rivas, J. and Fontanillo, C. (2010). Protein–protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS Comput Biol*, 6(6):e1000807.

Deshpande, R., Sharma, S., Verfaillie, C. M., Hu, W.-S., and Myers, C. L. (2010). A scalable approach for discovering conserved active subnetworks across species. *PLoS computational biology*, 6(12):e1001028.

Dittrich, M. T., Klau, G. W., Rosenwald, A., Dandekar, T., and Müller, T. (2008). Identifying functional modules in protein–protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231.

Donoho, D. L. (2006). Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306.

Durant, L., Watford, W. T., Ramos, H. L., Laurence, A., Vahedi, G., Wei, L., Takahashi, H., Sun, H.-W., Kanno, Y., Powrie, F., et al. (2010). Diverse targets of the transcription factor stat3 contribute to t cell pathogenicity and homeostasis. *Immunity*, 32(5):605–615.

Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210.

El-Kebir, M., Soueidan, H., Hume, T., Beisser, D., Dittrich, M., Müller, T., Blin, G., Heringa, J., Nikolski, M., Wessels, L. F., et al. (2015). xheinz: An algorithm for mining cross-species network modules under a flexible conservation model. *Bioinformatics*, page btv316.

Elmsallati, A., Clark, C., and Kalita, J. (2016). Global alignment of protein-protein interaction networks: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(4):689–705.

Erdos, P. and Rényi, A. (1959). On random graphs i. *Publ. Math. Debrecen*, 6:290–297.

Escolano, F. and Hancock, E. R. (2014). The mutual information between graphs. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 94–99. IEEE.

Estévez, P. A., Tesmer, M., Perez, C. A., and Zurada, J. M. (2009). Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201.

Fischetti, M., Hamacher, H. W., Jørnsten, K., and Maffioli, F. (1994). Weighted k-cardinality trees: Complexity and polyhedral structure. *Networks*, 24(1):11–21.

Flicek, P., Amode, M. R., Barrell, D., Beal, K., Billis, K., Brent, S., Carvalho-Silva, D., Clapham, P., Coates, G., Fitzgerald, S., et al. (2014). Ensembl 2014. *Nucleic acids research*, 42(D1):D749–D755.

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3):75–174.

Fresno, C. and Fernández, E. A. (2013). Rdavidwebservice: a versatile r interface to david. *Bioinformatics*, page btt487.

Gambardella, G., Moretti, M. N., de Cegli, R., Cardone, L., Peron, A., and di Bernardo, D. (2013). Differential network analysis for the identification of condition-specific pathway activity and regulation. *Bioinformatics*, 29(14):1776–1785.

Gao, C., McDowell, I. C., Zhao, S., Brown, C. D., and Engelhardt, B. E. (2016). Context specific and differential gene co-expression networks via bayesian biclustering. *PLoS Comput Biol*, 12(7):e1004791.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989:102.

Gong, P., Gai, K., and Zhang, C. (2011). Efficient euclidean projections via piecewise root finding and its application in gradient projection. *Neurocomputing*, 74(17):2754–2766.

Goyal, P. and Ferrara, E. (2017). Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*.

Granovetter, M. S. (1973). The strength of weak ties. *American journal of sociology*, 78(6):1360–1380.

Guo, Z., Li, Y., Gong, X., Yao, C., Ma, W., Wang, D., Li, Y., Zhu, J., Zhang, M., Yang, D., et al. (2007). Edge-based scoring and searching method for identifying condition-responsive protein–protein interaction sub-network. *Bioinformatics*, 23(16):2121–2128.

Ha, M. J., Baladandayuthapani, V., and Do, K.-A. (2015). Dingo: differential network analysis in genomics. *Bioinformatics*, 31(21):3413–3420.

Hau, C., Kanda, N., Tada, Y., Shibata, S., Sato, S., and Watanabe, S. (2014). Prolactin induces the production of th17 and th1 cytokines/chemokines in murine imiquimod-induced psoriasiform skin. *Journal of the European Academy of Dermatology and Venereology*, 28(10):1370–1379.

He, S., Zhu, Z., Jia, G., Tennant, D., Huang, Q., Tang, K., Heath, J., Musolesi, M., and Yao, X. (2016). Cooperative co-evolutionary module identification with application to cancer disease module discovery. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

Holme, P. and Saramäki, J. (2012). Temporal networks. *Physics reports*, 519(3):97–125.

Horvath, S., Zhang, B., Carlson, M., Lu, K., Zhu, S., Felciano, R., Laurance, M., Zhao, W., Qi, S., Chen, Z., et al. (2006). Analysis of oncogenic signaling networks in glioblastoma identifies aspm as a molecular target. *Proceedings of the National Academy of Sciences*, 103(46):17402–17407.

Hsu, C.-L., Juan, H.-F., and Huang, H.-C. (2015). Functional analysis and characterization of differential coexpression networks. *Scientific reports*, 5.

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2008). Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature protocols*, 4(1):44–57.

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13.

Huang, Q., White, T., Jia, G., Musolesi, M., Turan, N., Tang, K., He, S., Heath, J. K., and Yao, X. (2012). Community detection using cooperative co-evolutionary differential evolution. In *International Conference on Parallel Problem Solving from Nature*, pages 235–244. Springer.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Hulovatyy, Y., Chen, H., and Milenković, T. (2015). Exploring the structure and function of temporal networks with dynamic graphlets. *Bioinformatics*, 31(12):i171–i180.

Ideker, T. and Krogan, N. J. (2012). Differential network biology. *Molecular systems biology*, 8(1):565.

Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(suppl 1):S233–S240.

Ideker, T., Thorsson, V., Ranish, J. A., Christmas, R., Buhler, J., Eng, J. K., Bumgarner, R., Goodlett, D. R., Aebersold, R., and Hood, L. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292(5518):929–934.

Ishibuchi, H., Yoshida, T., and Murata, T. (2003). Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *Evolutionary Computation, IEEE Transactions on*, 7(2):204–223.

Jeong, H., Mason, S. P., Barabási, A.-L., and Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833):41–42.

Jia, G., Cai, Z., Musolesi, M., Wang, Y., Tennant, D. A., Weber, R. J., Heath, J. K., and He, S. (2012). Community detection in social and biological networks using differential evolution. In *Learning and Intelligent Optimization*, pages 71–85. Springer Berlin Heidelberg.

Jiang, P. and Singh, M. (2010). Spici: a fast clustering algorithm for large biological networks. *Bioinformatics*, 26(8):1105–1111.

Jiang, Q., Wang, Y., Hao, Y., Juan, L., Teng, M., Zhang, X., Li, M., Wang, G., and Liu, Y. (2009). mir2disease: a manually curated database for microrna deregulation in human disease. *Nucleic acids research*, 37(suppl 1):D98–D104.

Kanehisa, M. and Goto, S. (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30.

Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., and Tanabe, M. (2015). Kegg as a reference resource for gene and protein annotation. *Nucleic acids research*, page gkv1070.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM.

Karp, R. M. (1972). *Reducibility among combinatorial problems*. Springer.

Karwacz, K., Yosef, N., and Kuchroo, V. (2013). Irf-1 is a key transcriptional regulator of tr1 differentiation (p1135). *The Journal of Immunology*, 190(1 Supplement):50–10.

Khatri, P. and Drăghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., et al. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, 2(3):203–271.

Klammer, M., Godl, K., Tebbe, A., and Schaab, C. (2010). Identifying differentially regulated subnetworks from phosphoproteomic data. *BMC bioinformatics*, 11(1):351.

Kongsbak, M., Levring, T. B., Geisler, C., and Von Essen, M. R. (2015). The vitamin d receptor andt cell function. *Lipid Signaling in T Cell Development and Function*, page 119.

Koutra, D., Parikh, A., Ramdas, A., and Xiang, J. (2011). Algorithms for graph similarity and subgraph matching. In *Technical report*. Carnegie-Mellon-University.

Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., and Pržulj, N. (2010). Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, page rsif20100063.

Kuchaiev, O. and Pržulj, N. (2011). Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396.

Kudva, A., Scheller, E. V., Robinson, K. M., Crowe, C. R., Choi, S. M., Slight, S. R., Khader, S. A., Dubin, P. J., Enelow, R. I., Kolls, J. K., et al. (2011). Influenza a inhibits th17-mediated host defense against bacterial pneumonia in mice. *The Journal of Immunology*, 186(3):1666–1674.

Kuijjer, M. L., Tung, M., Yuan, G., Quackenbush, J., and Glass, K. (2015). Estimating sample-specific regulatory networks. *arXiv preprint arXiv:1505.06440*.

Kumari, S., Nie, J., Chen, H.-S., Ma, H., Stewart, R., Li, X., Lu, M.-Z., Taylor, W. M., and Wei, H. (2012). Evaluation of gene association methods for coexpression network construction and biological knowledge discovery. *PloS one*, 7(11).

Lancichinetti, A., Fortunato, S., and Kertész, J. (2009). Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015.

Langfelder, P. and Horvath, S. (2008). Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):559.

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

Lazarevic, V., Chen, X., Shim, J.-H., Hwang, E.-S., Jang, E., Bolm, A. N., Oukka, M., Kuchroo, V. K., and Glimcher, L. H. (2011). T-bet represses th17 differentiation by preventing runx1-mediated activation of the gene encoding ror [gamma] t. *Nature immunology*, 12(1):96–104.

Li, D., Brown, J., Orsini, L., Pan, Z., Hu, G., and He, S. (2016a). Moda: Module differential analysis for weighted gene co-expression network. *bioRxiv*, page 053496.

Li, D., He, S., Pan, Z., and Hu, G. (2016b). Active modules for multilayer weighted gene co-expression networks: a continuous optimization approach. *bioRxiv*, page 056952.

Li, D., Pan, Z., Hu, G., Zhu, Z., and He, S. (2017a). Active module identification in intracellular networks using a memetic algorithm with a new binary decoding scheme. *BMC Genomics*, 18(2):209.

Li, D., Pan, Z., Hu, G., Zhu, Z., and He, S. (2017b). Extracting active modules from multilayer ppi network: a continuous optimization approach. *Submitted*.

Li, W., Liu, C.-C., Zhang, T., Li, H., Waterman, M. S., and Zhou, X. J. (2011). Integrative

analysis of many weighted co-expression networks using tensor computation. *PLoS Comput Biol*, 7(6):e1001106.

Li, X., Wu, M., Kwoh, C.-K., and Ng, S.-K. (2010). Computational approaches for detecting protein complexes from protein interaction networks: a survey. *BMC genomics*, 11(1):S3.

Lin, C.-b. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779.

Lin, C.-C., Bradstreet, T. R., Schwarzkopf, E. A., Sim, J., Carrero, J. A., Chou, C., Cook, L. E., Egawa, T., Taneja, R., Murphy, T. L., et al. (2014). Bhlhe40 controls cytokine production by t cells and is essential for pathogenicity in autoimmune neuroinflammation. *Nature communications*, 5.

Liu, Y., Tennant, D. A., Zhu, Z., Heath, J. K., Yao, X., and He, S. (2014). Dime: a scalable disease module identification algorithm with application to glioma progression. *PloS one*, 9(2).

Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G. W., Mutzel, P., and Fischetti, M. (2006). An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Mathematical Programming*, 105(2-3):427–449.

Lui, T. W., Tsui, N. B., Chan, L. W., Wong, C. S., Siu, P. M., and Yung, B. Y. (2015). Decode: an integrated differential co-expression and differential expression analysis of gene expression data. *BMC bioinformatics*, 16(1):182.

Luo, F., Yang, Y., Chen, C.-F., Chang, R., Zhou, J., and Scheuermann, R. H. (2006). Modular organization of protein interaction networks. *Bioinformatics*, 23(2):207–214.

Lutkepohl, H. (1997). Handbook of matrices. *Computational Statistics and Data Analysis*, 2(25):243.

Ma, H., Schadt, E. E., Kaplan, L. M., and Zhao, H. (2011). Cosine: Condition-specific sub-network identification using a global optimization method. *Bioinformatics*, 27(9):1290–1298.

Maere, S., Heymans, K., and Kuiper, M. (2005). Bingo: a cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16):3448–3449.

Mewes, H.-W., Amid, C., Arnold, R., Frishman, D., Güldener, U., Mannhaupt, G., Münsterkötter, M., Pagel, P., Strack, N., Stümpflen, V., et al. (2004). Mips: analysis and annotation of proteins from whole genomes. *Nucleic acids research*, 32(suppl_1):D41–D44.

Miller, J. A., Horvath, S., and Geschwind, D. H. (2010). Divergence of human and mouse brain transcriptome highlights alzheimer disease pathways. *Proceedings of the National Academy of Sciences*, 107(28):12698–12703.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827.

Mitra, K., Carvunis, A.-R., Ramesh, S. K., and Ideker, T. (2013). Integrative approaches for finding modular structure in biological networks. *Nature Reviews Genetics*, 14(10):719–732.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628.

Moscato, P. et al. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878.

Nacu, Ş., Critchley-Thorne, R., Lee, P., and Holmes, S. (2007). Gene expression network analysis and applications to immunology. *Bioinformatics*, 23(7):850–858.

Nepusz, T., Yu, H., and Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nature methods*, 9(5):471–472.

Neri, F. and Cotta, C. (2012). Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2:1–14.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2):167–256.

Newman, M. E. (2004a). Analysis of weighted networks. *Physical Review E*, 70(5):056131.

Newman, M. E. (2004b). Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104.

Newman, M. E. (2013). Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822.

Organizers, D. C. (2016). Disease module identification dream challenge. `https://www.synapse.org/#!Synapse:syn6156761`. Accessed: 2017-07-15.

Orsini, L., Brown, J. B., Shams Solari, O., Li, D., He, S., Podicheti, R., Stoiber, M. H., Spanier, K. I., Gilbert, D., Jansen, M., et al. (2017). Early transcriptional response pathways in daphnia magna are coordinated in networks of crustacean specific genes. *Molecular Ecology*.

Orsini, L., Gilbert, D., Podicheti, R., Jansen, M., Brown, J. B., Solari, O. S., Spanier, K. I., Colbourne, J. K., Rush, D., Decaestecker, E., et al. (2016). Daphnia magna transcriptome by rna-seq across 12 environmental stressors. *Scientific data*, 3.

O'Shea, J. J., Steward-Tharp, S. M., Laurence, A., Watford, W. T., Wei, L., Adamson, A. S., and Fan, S. (2009). Signal transduction and th17 cell differentiation. *Microbes and Infection*, 11(5):599–611.

Ou-Yang, L., Dai, D.-Q., Li, X.-L., Wu, M., Zhang, X.-F., and Yang, P. (2014). Detecting temporal protein complexes from dynamic protein-protein interaction networks. *BMC bioinformatics*, 15(1):335.

Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *arXiv preprint physics/0506133*.

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

Poynton, H. C., Lazorchak, J. M., Impellitteri, C. A., Blalock, B. J., Rogers, K., Allen, H. J., Loguinov, A., Heckman, J. L., and Govindasmawy, S. (2012). Toxicogenomic responses of nanotoxicity in daphnia magna exposed to silver nitrate and coated silver nanoparticles. *Environmental science & technology*, 46(11):6288–6296.

Pramila, T., Miles, S., GuhaThakurta, D., Jemiolo, D., and Breeden, L. L. (2002). Conserved homeodomain proteins interact with mads box protein mcm1 to restrict ecb-dependent transcription to the m/g1 phase of the cell cycle. *Genes & development*, 16(23):3034–3045.

Prill, R. J., Iglesias, P. A., and Levchenko, A. (2005). Dynamic properties of network motifs contribute to biological network organization. *PLoS Biol*, 3(11):e343.

Pržulj, N. (2007). Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183.

Pu, S., Wong, J., Turner, B., Cho, E., and Wodak, S. J. (2008). Up-to-date catalogues of yeast protein complexes. *Nucleic acids research*, 37(3):825–831.

Qiu, Y.-Q., Zhang, S., and Zhang, X.-S. (2008). Uncovering differentially expressed pathways with protein interaction and gene expression data. In *The Second International Symposium on Optimization and Systems Biology*, pages 74–82.

Qiu, Y.-Q., Zhang, S., Zhang, X.-S., and Chen, L. (2010). Detecting disease associated modules and prioritizing active genes based on high throughput data. *BMC bioinformatics*, 11(1):26.

Rajagopalan, D. and Agarwal, P. (2005). Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics*, 21(6):788–793.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A.-L. (2002). Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555.

Rieckmann, J. C., Geiger, R., Hornburg, D., Wolf, T., Kveler, K., Jarrossay, D., Sallusto, F., Shen-Orr, S. S., Lanzavecchia, A., Mann, M., et al. (2017). Social network architecture of human immune cells unveiled by quantitative proteomics. *Nature Immunology*, 18(5):583–593.

Rodola, E., Torsello, A., Harada, T., Kuniyoshi, Y., and Cremers, D. (2013). Elastic net constraints for shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1169–1176.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltnane, J. M., et al.

(2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, 346(25):1937–1947.

Rowan, A. G., Fletcher, J. M., Ryan, E. J., Moran, B., Hegarty, J. E., O'Farrelly, C., and Mills, K. H. (2008). Hepatitis c virus-specific th17 cells are suppressed by virus-induced tgf-β. *The Journal of Immunology*, 181(7):4485–4494.

Ruan, J., Dean, A. K., and Zhang, W. (2010). A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC systems biology*, 4(1):8.

Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein–protein interactions in yeast. *Nature biotechnology*, 18(12):1257–1261.

Seif, F., Khoshmirsafa, M., Aazami, H., Mohsenzadegan, M., Sedighi, G., and Bahar, M. (2017). The role of jak-stat signaling pathway and its regulators in the fate of t helper cells. *Cell Communication and Signaling*, 15(1):23.

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504.

Sharan, R. and Ideker, T. (2006). Modeling cellular machinery through biological network comparison. *Nature biotechnology*, 24(4):427–433.

Smedley, D., Haider, S., Ballester, B., Holland, R., London, D., Thorisson, G., and Kasprzyk, A. (2009). Biomart–biological queries made easy. *BMC genomics*, 10(1):22.

Smyth, G. K. (2005). Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer.

Spirin, V. and Mirny, L. A. (2003). Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences*, 100(21):12123–12128.

Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *science*, 302(5643):249–255.

Suddason, T. and Gallagher, E. (2016). Genetic insights into map3k-dependent proliferative expansion of t cells. *Cell Cycle*, 15(15):1956–1960.

Szklarczyk, D., Franceschini, A., Wyder, S., Forslund, K., Heller, D., Huerta-Cepas, J., Simonovic, M., Roth, A., Santos, A., Tsafou, K. P., et al. (2014). String v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research*, page gku1003.

Tang, K., Mei, Y., and Yao, X. (2009). Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *Evolutionary Computation, IEEE Transactions on*, 13(5):1151–1166.

Tang, X., Wang, J., Liu, B., Li, M., Chen, G., and Pan, Y. (2011). A comparison of the functional modules identified from time course and static ppi network data. *BMC bioinformatics*, 12(1):339.

Tanigaki, K. and Honjo, T. (2010). Chapter seven-two opposing roles of rbp-j in notch signaling. *Current topics in developmental biology*, 92:231–252.

Taylor, I. W., Linding, R., Warde-Farley, D., Liu, Y., Pesquita, C., Faria, D., Bull, S., Pawson, T., Morris, Q., and Wrana, J. L. (2009). Dynamic modularity in protein interaction networks predicts breast cancer outcome. *Nature biotechnology*, 27(2):199–204.

Thiele, I. and Palsson, B. Ø. (2010). A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nature protocols*, 5(1):93.

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

Tripathi, S., Pohl, M. O., Zhou, Y., Rodriguez-Frandsen, A., Wang, G., Stein, D. A., Moulton, H. M., DeJesus, P., Che, J., Mulder, L. C., et al. (2015). Meta-and orthogonal integration of influenza omics data defines a role for ubr4 in virus budding. *Cell host & microbe*, 18(6):723–735.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525.

Tuomela, S., Salo, V., Tripathi, S. K., Chen, Z., Laurila, K., Gupta, B., Äijö, T., Oikari, L., Stockinger, B., Lähdesmäki, H., et al. (2012). Identification of early gene expression changes during human th17 cell differentiation. *Blood*, 119(23):e151–e160.

Ulitsky, I., Laurent, L. C., and Shamir, R. (2010). Towards computational prediction of microrna function and activity. *Nucleic acids research*, 38(15):e160–e160.

Ulitsky, I. and Shamir, R. (2007). Identification of functional modules using network topology and high-throughput data. *BMC systems biology*, 1(1):8.

Ulitsky, I. and Shamir, R. (2009). Identifying functional modules using expression profiles and confidence-scored protein interactions. *Bioinformatics*, 25(9):1158–1164.

van Dam, S., Võsa, U., van der Graaf, A., Franke, L., and de Magalhães, J. P. (2017). Gene co-expression analysis for functional classification and gene–disease predictions. *Briefings in Bioinformatics*, page bbw139.

Villarino, A. V., Gallo, E., and Abbas, A. K. (2010). Stat1-activating cytokines limit th17 responses through both t-bet–dependent and–independent mechanisms. *The Journal of Immunology*, 185(11):6461–6471.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Wang, J., Peng, X., Li, M., and Pan, Y. (2013). Construction and application of dynamic protein interaction network based on time course gene expression data. *Proteomics*, 13(2):301–312.

Wang, J., Peng, X., Peng, W., and Wu, F.-X. (2014). Dynamic protein interaction network construction and applications. *Proteomics*, 14(4-5):338–352.

Wang, Y. and Xia, Y. (2008). Condition specific subnetwork identification using an optimization model. *Proc Optim Syst Biol*, 9:333–340.

Wang, Y.-C. and Chen, B.-S. (2010). Integrated cellular network of transcription regulations and protein-protein interactions. *BMC Systems Biology*, 4(1):20.

Wang, Z., Gerstein, M., and Snyder, M. (2009). Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63.

Warde-Farley, D., Donaldson, S. L., Comes, O., Zuberi, K., Badrawi, R., Chao, P., Franz, M., Grouios, C., Kazi, F., Lopes, C. T., et al. (2010). The genemania prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic acids research*, 38(suppl 2):W214–W220.

Wei, L., Laurence, A., Elias, K. M., and O'Shea, J. J. (2007). Il-21 is produced by th17 cells and drives il-17 production in a stat3-dependent manner. *Journal of Biological Chemistry*, 282(48):34605–34610.

Wei, Y.-C. and Cheng, C.-K. (1989). Towards efficient hierarchical designs by ratio cut partitioning. In *Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers., 1989 IEEE International Conference on*, pages 298–301. IEEE.

Wikipedia. Biological pathway. [Online; accessed 2015-01-05].

Wikipedia. Integer programming. [Online; accessed 2017-12-11].

Wikipedia. Linear programming. [Online; accessed 2017-12-11].

Wikipedia. Simulated annealing. [Online; accessed 2017-11-22].

Wilkinson, D. M. and Huberman, B. A. (2004). A method for finding communities of related genes. *proceedings of the national Academy of sciences*, 101(suppl 1):5241–5248.

Woeginger, G. J. (1992). Computing maximum valued regions. *Acta Cybern.*, 10(4):303–315.

Wu, Z., Zhao, X., and Chen, L. (2009). Identifying responsive functional modules from protein-protein interaction network. *Molecules and cells*, 27(3):271–277.

Xu, Y., Salapaka, S. M., and Beck, C. L. (2013). A distance metric between directed weighted graphs. In *52nd IEEE Conference on Decision and Control*, pages 6359–6364. IEEE.

Xue, Z., Huang, K., Cai, C., Cai, L., Jiang, C.-y., Feng, Y., Liu, Z., Zeng, Q., Cheng, L., Sun, Y. E., et al. (2013). Genetic programs in human and mouse early embryos revealed by single-cell rna [thinsp] sequencing. *Nature*, 500(7464):593–597.

Yang, H., Lee, S.-M., Gao, B., Zhang, J., and Fang, D. (2013). Histone deacetylase sirtuin 1 deacetylates irf1 protein and programs dendritic cells to control th17 protein differentiation during autoimmune inflammation. *Journal of Biological Chemistry*, 288(52):37256–37266.

Yates, A., Akanni, W., Amode, M. R., Barrell, D., Billis, K., Carvalho-Silva, D., Cummins, C., Clapham, P., Fitzgerald, S., Gil, L., et al. (2015). Ensembl 2016. *Nucleic acids research*, 44(D1):D710–D716.

Yosef, N., Shalek, A. K., Gaublomme, J. T., Jin, H., Lee, Y., Awasthi, A., Wu, C., Karwacz, K., Xiao, S., Jorgolli, M., et al. (2013). Dynamic regulatory network controlling th17 cell differentiation. *Nature*, 496(7446):461–468.

Yu, G., Wang, L.-G., Yan, G.-R., and He, Q.-Y. (2015). Dose: an r/bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics*, 31(4):608–609.

Zhang, B. and Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1).

Zhang, T. (2010). Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research*, 11:1081–1107.

Zhao, X.-M., Wang, R.-S., Chen, L., and Aihara, K. (2008). Uncovering signal transduction networks from high-throughput data by integer linear programming. *Nucleic acids research*, 36(9):e48–e48.

Zhao, Y., Levina, E., and Zhu, J. (2011). Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326.

Zhu, J., Yamane, H., and Paul, W. E. (2010). Differentiation of effector cd4 t cell populations. *Annual review of immunology*, 28:445.

Zhu, Z., Ong, Y.-S., and Dash, M. (2007). Wrapper–filter feature selection algorithm using a memetic framework. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(1):70–76.

Zinman, G. E., Naiman, S., O'Dee, D. M., Kumar, N., Nau, G. J., Cohen, H. Y., and Bar-Joseph, Z. (2015). Moduleblast: identifying activated sub-networks within and across species. *Nucleic acids research*, 43(3):e20–e20.

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