

Graph-Based Machine Learning And Its Application On Multi-omics Data Analysis

By

Han Zhang

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
University of Birmingham
March 2022

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

ABSTRACT

With the development of omics technologies, there is a large amount of biological data available. The biological data is usually *complex*, *ill-sampled*, and *high dimensional*. As a result, gaining insightful knowledge from the biological data is still a challenging problem. Many of those biological data can be represented using graphs. Over the years, many graph-based machine learning methods have been developed to analyse graphs in many tasks, such as module detection, feature engineering, and link prediction. This thesis provides the applications of graph-based machine learning methods on three types of biological data with the following contributions:

- A novel method that allows overlapping between the peak modules in the metabolite annotation problem on liquid chromatography-mass spectrometry untargeted data.
- A novel method that allows high freedom of the shapes of the topological modules in the detection of network biomarkers for gene expression data.
- A novel method that incorporates the implicit networks constructed from the drug-target interaction network in the drug-target interaction prediction problem.

With these contributions, we can gain more understanding of the biological data.

Acknowledgements

First of all, I want to thank my supervisor Dr Shan He for his supervision. I could not present this thesis without his supervision. He provided necessary guidelines of the research direction, and gave me freedom in conducting research.

I would love to thank my research student monitor group members Dr Leandro Minku, Dr Rami Bahsoon, and the former members Dr John Bullinaria, Dr Kashif Rajpoot, and Prof Joshua Knowles. In the research student monitor group meetings, they have always provided helpful suggestions for my research.

I wish to thank Dr Haiying Wang from University of Ulster and Dr Jinming Duan from University of Birmingham for being the examiners of my thesis. The comments from both of them are extremely useful and insightful.

I would like to extend my thanks to my friends David McDonald and Chengbin Hou. I have enjoyed working with them. I would also like to thank all the Shan He Group members for their help.

Finally, I would like to thank my family, especially my wife Zhiyu Qin. They are always supportive of me. This thesis is dedicated to them.

Contents

Abstract **i**

Acknowledgements **ii**

1 Introduction **1**

1.1 Multi-omics data 1

1.1.1 Liquid chromatography–mass spectrometry data . . . 2

1.1.2 Gene expression data 4

1.1.3 Drug-target interaction data 5

1.2 Network representations of multi-omics data 6

1.2.1 Basic network concepts 7

1.2.2 Special networks 9

1.2.3 Network representation for LC-MS data 10

1.2.4 Network representation for gene expression data . . . 11

1.2.5 Network representation for drug-target interaction data 11

1.3 Graph-based machine learning methods 12

1.3.1 Module detection 13

1.3.2 Network embedding 15

1.4 Research questions 17

1.4.1 The annotation of metabolites for LC-MS untargeted data 17

1.4.2 Detection of network biomarker from gene expression data 19

1.4.3 Drug-target interaction prediction 22

1.5 Thesis contributions 23

1.6 List of publications 24
1.7 Thesis organisation 26

2 Literature review **28**
2.1 Metabolite annotation 28
2.1.1 The peak-pair based methods 29
2.1.2 The network based methods 31
2.1.3 The probability based methods 37
2.1.4 Metabolite annotation evaluation criteria 37
2.2 The detection of network biomarker 39
2.2.1 Clustering based methods 39
2.2.2 Clique based methods 43
2.2.3 Gene enrichment analysis 44
2.3 Drug-target interaction prediction 45
2.3.1 Target structure based methods 45
2.3.2 Ligand similarity based methods 46
2.3.3 Machine learning based methods 46
2.3.4 Drug-target interaction prediction evaluation criteria . 49
2.4 Chapter summary 50

3 Metabolite annotation via module detection **51**
3.1 Introduction 51
3.2 Method 53
3.2.1 Candidate ionised product list construction 54
3.2.2 Attributed peak similarity network construction 55
3.2.3 Peak module detection 56
3.2.4 Metabolite reaction matching 60
3.2.5 The proposed algorithm 61
3.2.6 Complexity analysis 62
3.3 Experiments 63

3.3.1 Comparison with state-of-the-art algorithms 63
3.3.2 Annotation score sensitivity 66
3.3.3 The effect of metabolite reaction matching 67
3.3.4 Parameter sensitivity 69
3.4 Chapter summary 70

4 Network biomarker detection using Gaussian mixture model clustering 73
4.1 Introduction 73
4.2 Method 75
4.2.1 Gene co-expression network construction 75
4.2.2 Network embedding for a gene co-expression network 76
4.2.3 Clustering 78
4.2.4 Measurement of module significance score 80
4.2.5 Overview of the proposed method 80
4.3 Experiments 82
4.3.1 Datasets 82
4.3.2 Hyper-parameter settings 83
4.3.3 Experiments on detecting network biomarkers 84
4.3.4 The effect of Gaussian mixture model clustering 86
4.3.5 Gene enrichment analysis of the network biomarkers . 88
4.3.6 Parameter sensitivity 93
4.4 Chapter summary 98

5 Network embedding based drug-target interaction reaction 99
5.1 Introduction 99
5.2 Method 101
5.2.1 Network construction 102
5.2.2 Network embedding 103
5.2.3 Feature vector construction for training samples 104

5.2.4 Drug-target interaction prediction 104
5.2.5 Overview of the proposed method 105
5.2.6 Complexity analysis 106
5.3 Experiments 107
5.3.1 Experiment settings 108
5.3.2 Comparison with other state-of-the-art methods 109
5.3.3 The effect of the implicit networks 110
5.3.4 Case study on novel drug-target interaction predictions 112
5.3.5 Parameter sensitivity 112
5.4 Chapter summary 116

6 Discussion **117**
6.1 Contributions & conclusions 117
6.1.1 Metabolite annotation 118
6.1.2 Network biomarker detection 118
6.1.3 Drug-target interaction prediction 119
6.2 Limitations 120
6.2.1 Limitation of the proposed metabolite annotation method 120
6.2.2 Limitation of the network biomarker detection method 120
6.2.3 Limitation of the drug-target interaction prediction method 121
6.3 Future directions 121
6.3.1 Future directions for metabolite annotation 121
6.3.2 Future directions for biomarker detection 122
6.3.3 Future directions for drug-target interaction prediction 122

Bibliography **124**

List of Figures

1.1 The profiling result of a sample using LC-MS. The metabolites are separated by LC, hence, different metabolites would be measured in different RT. Then, for each metabolite, the ionised products would be measured as multiple peaks due to the difference in m/z. 3

1.2 A demonstration of the proximity between nodes in a graph. The proximity is in regard to v_1, which is rendered in red, and all the other nodes are rendered in blue. The nodes in the yellow area are in first-order proximity with v_1, and those nodes include v_2, v_3, v_4. The nodes in the green area are in second-order proximity with v_1, and those nodes include v_5, v_6, v_7. The nodes in the grey area are in three-order proximity or fourth-order proximity with v_1, those nodes include v_8, v_9, v_{10}, v_{11} and v_{12}. 8

1.3 The main contents of this thesis. 26

2.1 An example of a peak shape with $m/z = 42.56$. The horizontal axis is the RT, and the vertical axis is the intensity of the peak 33

3.1 The framework of MAMD. MAMD takes a processed LC-MS peak table as the input. There are built-in candidate adduct list, metabolite list, and KEGG pathway reaction pairs MAMD representing the assumptions about the samples, and those assumptions can be customised for each specific datasets. The candidate metabolite list coupled with candidate adduct list is used to construct the candidate ionised product list which is described in Section 3.2.1. The peak table and the candidate ionised product list are used to construct an attributed PSN which is described in Section 3.2.2. The attributed PSN is used as the input for peak module detection which is described in Section 3.2.3. The annotated peak modules are used as the input for the metabolite pathway matching which is described in Section 3.2.4. Finally, the peak modules with both *score*1 and *score*2 are the output of MAMD. 53

3.2 The F1 score of MAMD and state-of-the-art algorithms under various numbers of decoy metabolites. In the six sub-figures, the horizontal-axis indicates the number of decoy metabolites and the vertical-axis indicates the F1 score of each algorithm. . 65

4.1 The proposed framework takes gene expression data and sample labels as input. The output is a list of the topological modules and the corresponding module significant scores. 76

4.2 The effect of using the GMM clustering on the module eigengene significance (MES) score of the detected network biomarkers. NBD-GMMC is the proposed method that employs network embedding coupled with GMM clustering to detect topological modules. NE-HC is the comparison method that employs network embedding coupled with hierarchical clustering (HC) to detect topological modules. From the bar chart, there is a performance gain in most of the datasets by employing GMM clustering over the distance based clustering method. Note that, the inputs for the clustering method, i.e., GMMC and HC, are the same. 87

4.3 The known interactions in the network biomarker detected by NBD-GMMC from BRCA. The network biomarker is plotted using StringDB (Szklarczyk et al., 2015). In this figure, a node represents a gene, and an edge represents a known pair-wise gene interaction. 89

4.4 The known interactions in the network biomarker detected by NBD-GMMC from KIRC. The network biomarker is plotted using StringDB (Szklarczyk et al., 2015). In this figure, a node represents a gene, and an edge represents a known pair-wise gene interaction. 90

4.5 The known interactions in the network biomarker detected by NBD-GMMC from LIHC. The network biomarker is plotted using StringDB (Szklarczyk et al., 2015). In this figure, a node represents a gene, and an edge represents a known pair-wise gene interaction. 91

5.1 An example of the implicit network construction. A circle represents a drug, and a diamond represents a target. From the DTI network, the DIN and TIN are constructed based on the common neighbours. 100

5.2 The AUPR scores of our method on the largest dataset DT-IN. The vertical axis indicates the AUPR scores and the horizontal axis indicates the experimental settings. It can be seen that, there is a performance gain by incorporating the implicit networks, and a large portion of the performance gain is from DIN. 111

5.3 The top 100 DTIs predicted by the proposed method on DT-IN dataset. Each circle represents a drug and each diamond represents a target. The edges between drugs and targets are the novel DTI predictions (i.e., new DTIs not recorded in the dataset). The top 20 DTI predictions are drawn in the solid lines, while the rest of DTI predictions are drawn in the dashed lines. 113

5.4 The parameter sensitivity analysis results of r that is the ratio between positive samples and negative samples in training the classifier. The vertical axis indicates the AUPR scores, and the horizontal axis indicates the experimental settings. The experiments are conducted on the dataset DT-IN. It can be seen that the performance of the proposed method keeps increasing until $r = 8$, after that the difference is marginal. 116

List of Tables

3.1 Candidate adducts in MAMD. 63
3.2 An example of two overlapping peak modules. 67
3.3 The impact of the annotation score under 1000 decoy metabolites. 68
3.4 Part 1 of the parameter sensitivity analysis of α and β on Std1.POS under 1000 decoy metabolites. 71
3.5 Part 2 of the parameter sensitivity analysis of α and β on Std1.POS under 1000 decoy metabolites. 72

4.1 The number of samples in the TCGA datasets. 83
4.2 Comparison of performance on detecting network biomarkers. 86
4.3 The gene enrichment analysis of the network biomarkers from BRCA,KIRC and LIHC. 92
4.4 The impact of hard threshold and soft threshold on BRCA. . . 95
4.5 The impact of hard threshold and soft threshold on KIRC. . . 96
4.6 The wall-clock time of the proposed method on BRCA and KIRC. 97
4.7 The impact of the node embedding dimension d on BRCA and KIRC. 97

5.1 The statistics of five benchmark DTI networks. 108
5.2 The AUPR scores on five benchmark datasets. 109
5.3 The parameter sensitivity analysis result of α on the DT-IN dataset. 114

List of Abbreviations

AUPR	**A**rea **U**nder **P**recision **R**ecall curve
AUC	**A**rea **U**nder operating characteristic **C**urve
BRCA	**BR**east invasive **CA**rcinoma
COAD	**CO**lon **AD**enocarcinoma
CPR	**C**ox **P**roportional-hazard **R**egression
DNA	**D**eoxyribo **N**ucleic **A**cid
DSSN	**D**rug **S**tructural **S**imilarity **N**etwork
DT-IN	**D**rug **T**arget **IN**hibition
DTI	**D**rug **T**arget **I**nteraction
EN	**EN**zyme
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
GCN	**G**ene **C**o-expression **N**etwork
GMM	**G**aussian **M**ixture **M**odel
GPCR	**G**-**P**rotein-**C**oupled **R**eceptors
HNSC	**H**ead-**N**eck **S**quamous cell **C**arcinoma
KIRC	**KI**dney **R**enal clear cell **C**arcinoma
KIRP	**KI**dney **R**enal **P**apillary cell carcinoma
LC	**L**iquid **C**hromotography
LC-MS	**L**iquid **C**hromotography-**M**ass **S**pectromy
LIHC	**LI**ver **H**epatocellular **C**arcinoma
LGM	**L**ogistic **R**egression **M**odel
LUAD	**LU**ng **AD**enocarcinoma

IC	**I**on **C**hannel
MAMD	**M**etabolite **A**nnotation via **M**odule **D**etection
MES	**M**odule **E**igengene **S**ignificance
MF	**M**olecular **F**ormula
MS	**M**ass **S**pectromy
m/z	**m**ass-to-**c**harge ratio
NE-DTIP	**N**etwork **E**mbedding based **D**rug **T**arget **I**nteraction **P**rediction
NBD-GMMC	**N**etwork **B**biomarker **D**etection via **G**aussian **M**ixture **M**odel
NR	**N**uclear **R**eceptor
PSN	**P**eak **S**imilarity **N**etwork
RNA	**R**ibo**n**ucleic **A**cid
RT	**R**etention **T**ime
PIAS	**P**eak **I**ntensity **A**cross **S**amples
PRAD	**PR**ostate **AD**enocarcinoma
SSN	**S**ample **S**imilarity **N**etwork
SD	**S**tandard **D**eviation
SVM	**S**upport **V**ector **M**achine
THCA	**TH**yroid **CA**ncer
TN	**T**rue **N**egative
TOM	**T**opological **O**verlapping **M**atrix
TP	**T**rue **P**ositive
TSSN	**T**arget **S**equence **S**imilarity **N**etwork
WCT	**W**all-**C**lock **T**ime

Chapter 1

Introduction

This chapter gives an introduction of the research topics of this thesis, and an overview of the whole thesis. The rest of this chapter is organised as follows. Section 1.1 introduces the multi-omics data. Section 1.2 introduces the concept of networks and the construction of biological networks from multi-omics data. Section 1.3 gives an overview of the network-based machine learning methods. Section 1.4 lists the research problems and research questions from LC-MS untargeted data, gene expression data, and DTI data. Section 1.5 lists the contributions. Section 1.6 lists the publications from this thesis. Section 1.7 lists the organisation of this thesis.

1.1 Multi-omics data

In recent years, many omics technologies have been developed. Omics technologies are defined as the methods that comprehensively and simultaneously measure the genetic or molecular profiles of a biological sample (Conesa and Beck, 2019). The omics data could reveal the functional status of genes and molecules. In practice, the experiment results from multiple samples are combined together as the biological data. It is worth mentioning that analysing the biological data is still a challenging problem, as the biological data is usually *complex*, *ill-sampled*, and *high dimensional*.

1.1.1 Liquid chromatography–mass spectrometry data

Liquid chromatography–mass spectrometry (LC–MS) is a primary analytical instrument for profiling the metabolites in the biological samples, e.g., blood, urine, saliva, etc. The metabolites are the small modules that have various functions in the organisms. There are roughly three steps in profiling of metabolites in LC-MS.

The first step is the separation of metabolites using liquid chromatography (LC). A biological sample is dissolved via mixing the sample with solvents that are called *mobile phase*. The types of the solvents are sample dependent. Then, the mixture of sample and solvents is injected in the top part of the LC column. The LC column contains fixed materials that are called *stationary phase*. The metabolites will elute from the LC column due to the effect of gravity as the LC column is vertically placed. Since the rates of elution are different between different metabolites, the metabolites are physically separated, i.e., the molecules from different metabolites are unlikely to be ejected from the LC column in the same time window.

The second step is the ionisation of metabolites. When the metabolites elute from the LC column, the metabolites are exposed to an abundant ion source to be ionised. There are two modes in LC-MS, the modes are positive mode and negative mode. For a metabolite, the ionised products usually take the forms of $[\text{MF+adduct}]^+$ and $[\text{MF-adduct}]^-$ in positive mode and negative mode respectively, where MF is the abbreviation for the molecular formula. It is worth mentioning that the number of adducts generated from a metabolite could vary from one to tens (Dunn et al., 2013).

The third step is the mass-to-charge ratio (m/z) measurement of ionised products. The ionised products of metabolites are injected into the mass spectrometry (MS), in which the ionised products are measured as peaks. A demonstration of the raw output of LC-MS is shown in Figure 1.1. For

each peak, the m/z, retention time (RT) and peak intensity are all measured by MS. The RT of a peak is calculated based on the time difference between the sample injection into the LC column and the measurement of m/z by the MS. The intensity of a peak indicates the abundance of this ionised product.

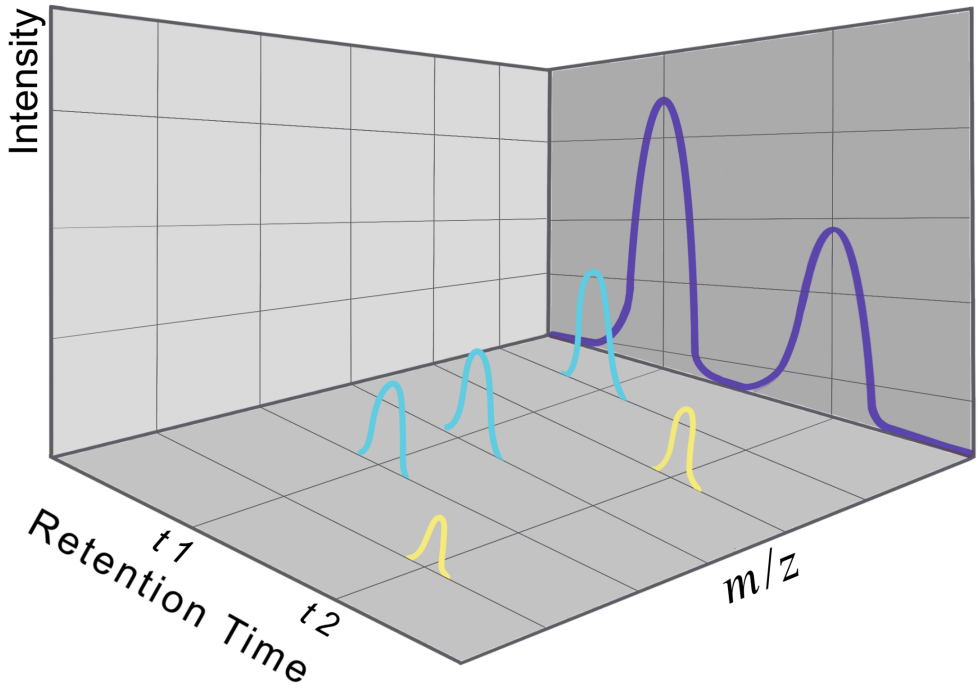


FIGURE 1.1: The profiling result of a sample using LC-MS. The metabolites are separated by LC, hence, different metabolites would be measured in different RT. Then, for each metabolite, the ionised products would be measured as multiple peaks due to the difference in m/z.

The raw output of LC-MS requires a lot of pre-processing. XCMS (Smith et al., 2006) is commonly used to process the raw LC-MS data. The major tasks in pre-processing include peak detection and alignment of peaks from multiple samples. In the peak detection task, peaks are identified, and the peak intensities are calculated based on the peak area. Usually, multiple samples will be profiled by LC-MS. Since there might be a subtle difference in elution rates between different samples, the RTs of the same peak from different samples could have a slight shift. As a result, peak alignment is required to detect the peak intensity across samples (PIAS). For a peak detected from

multiple samples, the median of RTs from all samples is regarded as the RT of the peak.

The output of XCMS includes three parts. First, a vector contains the m/z of the peaks. Second, a vector contains the RTs of the peaks. Third, a PIAS matrix that contains the intensity for all peaks across samples:

$$\begin{array}{c} Peaks \\ \\ \\ \\ \\ \end{array} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n^p} \\ x_{21} & x_{22} & \cdots & x_{2n^p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m^s1} & x_{m^s2} & \cdots & x_{m^sn^p} \end{bmatrix} \\ Samples \tag{1.1}$$

where each entry x_{ij} represents the peak intensity of peak i in sample j, m^s is the number of peaks and n^p is the number of samples. All the entries in the same row are from the same peak, and all the entries in the same column are from the same sample. The number of samples is based on the experimental design, and the samples could be grouped in different categories or there could be temporal information about the samples.

1.1.2 Gene expression data

Gene expression is the process that the information encoded in the genes is used to construct the end products, i.e., proteins. In recent years, many gene expression profiling technologies, e.g., DNA microarray and RNA sequencing, have been developed. Those technologies are capable of profiling the expression levels of genes (Mortazavi et al., 2008). In this thesis, we do not distinguish the difference between those two gene expression profiling technologies.

The gene expression data can be represented using a gene expression matrix:

$$\begin{array}{c} \\ Genes \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} \multicolumn{4}{c}{} \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n^s} \\ x_{21} & x_{22} & \cdots & x_{2n^s} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m^g1} & x_{m^g2} & \cdots & x_{m^gn^s} \end{bmatrix} \\ Samples \end{array} \tag{1.2}$$

where each entry x_{ij} represents the expression value of gene i in sample j, m^g is the number of genes and n^s is the number of samples. In a gene expression matrix, all the entries in the same row are from the same gene, and all the entries in the same column are from the same sample. Similar to Section 1.1.1, samples could be categorised in different groups, and the principle of designing experiments is to quantify the change of expression values under different conditions (Wang, Gerstein, and Snyder, 2009). There are many publicly available datasets for various types of cancers, e.g., the cancer genome atlas (TCGA) (Weinstein et al., 2013), in those datasets, there are many sample traits available such as gender, age, tumour/normal, tumour grade, etc.

1.1.3 Drug-target interaction data

The drug-target interaction (DTI) data describes the regulatory interactions between drugs and targets. The drugs are small molecules and the targets are usually proteins. The interactions between drugs and targets can be quantified via measuring the exhibition or inhibition relations between drugs and targets using high-throughput screening experiments (Gaulton et al., 2012).

The DTI data can be represented as a DTI matrix:

$$Drugs \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n^t} \\ x_{21} & x_{22} & \cdots & x_{2n^t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m^d1} & x_{m^d2} & \cdots & x_{m^dn^t} \end{bmatrix} \\ Targets \tag{1.3}$$

where each entry x_{ij} represents the regulatory information between drug i in target j, m^d is the number of drugs and n^t is the number of targets. In a DTI matrix, all the entries in the same row represent the interactions between a drug and all the targets, and all the entries in the same column represent the interactions between a target and all the drugs. Those entries are usually binary. $x_{ij} = 0$ indicates that drug i do not regulate target j, while $x_{ij} = 1$ indicates that drug i do regulate target j. Those binary values in a DTI matrix are computed from the affinity values of the DTIs. The affinity value describes the strength of the interactions. The binding affinity between a drug-target pair is the volume of a drug that is required to inhabit/exhibit the expression of the target by 50%. The binding affinity threshold of filtering the DTIs could be set as $10\mu mg$ (Luo et al., 2017).

1.2 Network representations of multi-omics data

Biological networks are often used to represent the relations between the biological entities observed from the multi-omics data. In this section, we will introduce the concept of network and the constructions of the biological networks from the aforementioned biological data.

1.2.1 Basic network concepts

Graph

A network is usually modelled as a graph which consists of a set of objects and their relations. A graph can be defined as:

$$G = (V, E) \tag{1.4}$$

where V denotes a set of nodes which are the entities of the networks and $E \subseteq V \times V$ denotes the set of edges i.e., relations between the entities. The numbers of nodes in a graph is denoted as $|V|$. The number of edges in a graph is denoted as $|E|$. A graph can also be represented using an adjacency matrix $A \subseteq V \times V$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{22} & \cdots & a_{nn} \end{bmatrix} \tag{1.5}$$

where each element $a_{ij} \in \{0, 1\}$ indicates whether an edge exists between the node pair (v_i, v_j). In this thesis, the edges are undirected, i.e., $a_{ij} = a_{ji}$.

The proximity between nodes

The proximity between nodes is a popular metric to measure the distance between nodes. The proximity between two nodes is defined as the minimum number of edges that is needed to travel to reach one node from the other node, and an example is shown in Figure 1.2.

The first-order proximity is the direct relation between two nodes. For each node pair (v_i, v_j), if $(v_i, v_j) \in E$, the proximity between them is a_{ij}. If $(v_i, v_j) \notin E$, the proximity between them is 0.The nodes that are directly connected v_i are regarded as the neighbours of the node v_i. The neighbours

of a node v_i is denoted as $\mathcal{N}(v_i)$, and the nodes without any neighbours are regarded as isolated nodes.

The second-order proximity is the two-step relation between two nodes. For each node pair (v_i, v_j), the second-order proximity is based on the common neighbours between them. The second-order proximity can capture the similarity of node pairs with similar structural contexts (Tang et al., 2015).

The high-order proximity is the relation between nodes that are at least three step away. The high-order proximity is assumed to be able to capture the global structural information of a network (Cao, Lu, and Xu, 2015).

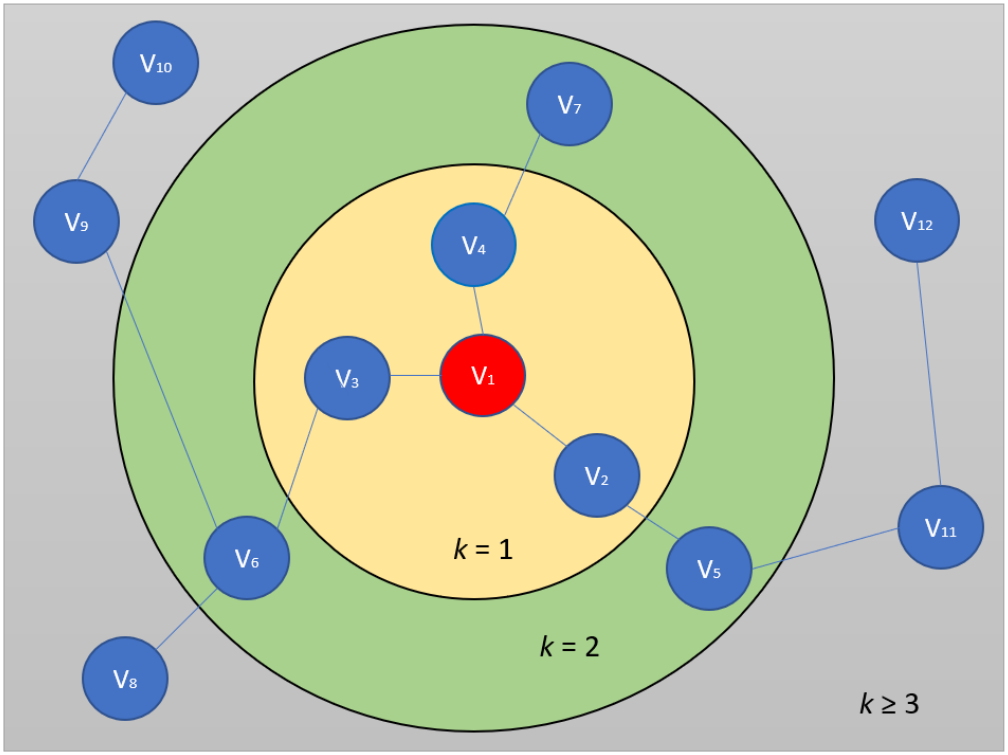


FIGURE 1.2: A demonstration of the proximity between nodes in a graph. The proximity is in regard to v_1, which is rendered in red, and all the other nodes are rendered in blue. The nodes in the yellow area are in first-order proximity with v_1, and those nodes include v_2, v_3, v_4. The nodes in the green area are in second-order proximity with v_1, and those nodes include v_5, v_6, v_7. The nodes in the grey area are in three-order proximity or fourth-order proximity with v_1, those nodes include v_8, v_9, v_{10}, v_{11} and v_{12}.

Sub-network

A sub-network $G^s = (V^s, E^s)$ is a subset of the network $G = (V, E)$. In G^s, the nodes V^s is a subset of V and the edges E^s is the subset of E.

Clique

A clique $G^c = (V^c, E^c)$ is a special kind of sub-network of the network $G = (V, E)$. In a clique, all the node pairs within this sub-network are connected, i.e.,there exists $(v_i^c, v_j^c) \in E^c$ for all node pairs $(v_i^c, v_j^c) \in V^c \times V^c, i \neq j$.

1.2.2 Special networks

There might be additional information about the entities in the biological networks, hence the aforementioned graph representation is not enough to describe the information. In this section, the special networks, which are relevant to our research, are introduced.

Attributed Network

In some networks, there could be additional attributes about the nodes. The additional attributes describe the properties of the nodes e.g., gender, age for social networks. For each node $v_i \in V$, there would be a feature vector $q \in \mathbb{R}^t$, where t is the dimension of the attributes.

Bipartite Network

A bipartite network can be treated as a special case of the attributed network. In a bipartite network, the nodes can be divided into two groups V^1 and V^2. In a bipartite network, all of the edges are from group V^1 to group V^2 and there are no edges between nodes within V^1 or V^2. A bipartite network can be represented as $G^{Bipartite} = (V^1, V^2, E)$, the edges are $V^1 \times V^2 \subseteq E^{Bipartite}$ that is different from a basic network.

Weighted Network

A network would be a weighted network if the edge weights are real value. Usually, the edge weights have been normalised to $[0, 1]$. As a result, for a node pair $(v_i, v_j) \in V \times V$, there is an edge weight $w_{ij} \in [0, 1]$. $w_{ij} = 0$ indicates that the edge does not exist between the node pair (v_i, v_j), while a positive edge weight indicates an edge does exist between them. For a positive edge weight, the higher the edge weight, the stronger the relation between the two corresponding nodes. All the edge weights can be represented in the weighted adjacency matrix $W = [w_{ij}]$.

1.2.3 Network representation for LC-MS data

An important task in analysing the LC-MS untargeted data is the annotation of peaks. In order to annotate peaks, the peaks are usually grouped together based on their features that are RT and PIAS. Peaks from the same metabolite source would have the following two properties. First, the peaks should have similar RTs, as molecules with the same chemical structures should coelute from the LC column. Second, peaks from the same metabolite source should have a similar pattern in PIAS, as the abundance of an ionised product follows the abundance of the metabolite source.

Based on two aforementioned properties, peak similarity network (PSN) is widely used to represent peaks and the similarities between the peaks in the network-based metabolite annotation methods (Kuhl et al., 2011; Broeckling et al., 2014; Uppal, Walker, and Jones, 2017; Senan et al., 2019). In a PSN, the nodes represent peaks and the edges represent the corresponding peaks that could be from the same metabolite source. The constructions of PSN will be discussed in Section 2.1.

1.2.4 Network representation for gene expression data

The gene co-expression network (GCN) is commonly used to represent the relations between the genes. In a GCN, each node represents a gene. The edge construction of a GCN will be discussed in Section 2.2. Finding biological meaningful gene co-expression modules has attracted a lot of research interests. The main reason is that the gene functional module, gene disease module and gene co-expression module are assumed to be overlapping (Barabási, Gulbahce, and Loscalzo, 2011).

1.2.5 Network representation for drug-target interaction data

An important task in analysing the DTI data is to make novel DTI predictions. Many DTI prediction methods are designed based on the "guild-by-association principle", i.e., drugs with similar structures are likely to bind with the same target, and targets with similar structure are likely to bind with the same drug. However, drugs are small molecules, while targets (i.e., proteins) are much larger. Hence, the structures of drugs and targets are represented using molecular fingerprints and protein sequence respectively. As a result, the similarity between drugs and targets need to be measured separately.

The prior knowledge about DTI data can be represented using networks. The known DTIs can be naturally represented using a network. In the drug similarity network, target similarity network and the DTI network, nodes would form densely connected sub-networks if they are highly connected in the similarity networks. Those densely connected sub-networks are useful to make novel predictions. Although, it is worth mentioning that the graph representation of networks could be sparse and high dimensional, as a result, they may not be compatible with many machine learning methods (Cui et al., 2018).

Drug-Target Interaction Network

The DTI data is bipartite by nature as there are no drug-drug and target-target interactions within. As a result, a DTI network is a bipartite network instead of a basic network. In a DTI network $G^{DTI} = (V^D, V^T, E^{DTI})$, V^D denotes a set of drugs, V^T denotes a set of targets, and E^{DTI} denotes the interactions between drugs and targets.

Drug Structure Similarity Network

To investigate the structural similarity between drugs, a drug structure similarity network (DSSN) $G^{DSSN} = (V^D, E^{DSSN})$ can be constructed. A drug is a small molecule. The chemical structures of drugs can be described by the molecular fingerprints. Given two molecular fingerprints, the pairwise similarity can be measured by many methods such as Tanimoto coefficient (Bajusz, Rácz, and Héberger, 2015) and Hattori et al., 2003.

Target Sequence Similarity Network

To investigate the sequence similarity between targets, a target structure similarity network (TSSN) $G^{TSSN} = (V^D, E^{DSSN})$ can be constructed. The targets are proteins, target structures can be described by the protein sequences. Given two protein sequences, the similarity score is usually calculated using Smith-Waterman algorithm (Smith, Waterman, et al., 1981).

1.3 Graph-based machine learning methods

As discussed in the Section 1.2, the biological datasets can be represented using graphs. Since graphs are usually *high-dimensional* and *sparse*, it is inapplicable to directly apply many machine learning methods on graphs (Cui

et al., 2018). Over the years, many machine learning methods have been developed to analyse the graphs, and those methods are regarded as graph-based machine learning methods (Chen and Ji, 2010; Akoglu, Tong, and Koutra, 2015). In this section, we give a brief introduction of the machine learning methods in module detection and network embedding, as those methods are relevant to the research questions in this thesis. The module detection methods are relevant to the research questions of the LC-MS untargeted data and the gene expression data. The network embedding methods are relevant to the research questions of the gene expression data and the DTI data.

1.3.1 Module detection

Detecting modules (sub-networks) from the networks has attracted a lot of research interest. The network module detection problem is an ill-defined problem (Fortunato and Hric, 2016), as different methods could have different definitions of the problem. In general, the module detection methods divide the graph into a series of modules, in which the nodes are densely connected. The network module detection methods have been applied on PSN and GCN to detect peak modules and gene co-expression modules respectively.

In most of the methods, the objective of module detection is to minimise the number of inter-module edges and maximise the number of intra-module edges. The inter-module edges are the edges that connect nodes from two different modules, and the intra-module edges are the edges that connect nodes in the same module.

Most of the module detection methods treat the problem as a clustering problem, i.e., divide the graph into non-overlapping modules (Fortunato and

Hric, 2016). Those clustering-based methods could be divided into the following categories: graph partition; optimisation; spectral clustering; hierarchical clustering.

Graph partition methods divide the graphs into a series of evenly sized non-overlapping modules (Kernighan and Lin, 1970; Karypis and Kumar, 1998). The number of the modules are user defined. The objective of those methods is to minimise the number of inter-module edges while keeping the module size almost equal.

The optimisation based methods detect modules based on the optimisation of an objective function (Newman and Girvan, 2004; Clauset, Newman, and Moore, 2004). The objective function is used to quantify the clustering result of the nodes in a graph. One of the popular objective function is modularity (Newman and Girvan, 2004) and it can be written as:

$$q = \frac{1}{2m^w} \sum_{v_i, v_j \in V} [w_{ij} - \frac{k_i k_j}{2m^w}] \delta(c_i, c_j) \tag{1.6}$$

where $m^w = \sum_{v_i, v_j \in V} w_{ij}$ is the total of edge weights in the network, w_{ij} is the edge weight between the node pair (v_i, v_j), $k_i = \sum_{v_j \in V} w_{ij}$ and $\delta(c_i, c_j)$ is the community label function. $\delta(c_i, c_j) = 1$ if $c_i = c_j$ and $\delta(c_i, c_j) = 0$ otherwise. Similar to graph partition methods, there is a penalty in the inter-module edges. Different from the graph partition methods, the sizes of the modules detected by those methods are data-driven.

Partitional clustering methods (Fortunato, 2010) learn the modules via projecting the nodes into a vector space. Those methods first learn the node embeddings which are vectors in a low dimension space. Then, a clustering method is applied on the node embeddings based on the distance of the node embeddings. K-mean clustering (MacQueen et al., 1967) is a commonly used method to detect modules.

The hierarchical clustering methods aim to identify the network modules

on different levels. In hierarchical clustering, the first step is to construct the dendrogram. The dendrogram is a tree-like structure to represent the graph partitions at different levels. There are two ways to construct a dendrogram:

- Agglomerative: this is a "bottom-up" approach. Each node is assumed to form a module. Then, the module pair is merged together based on a distance metric. The previous step will repeat until all nodes are merged into one module.

- Divisive: this is a "top-down" approach. All nodes are assumed to be in the same module. Then, this module is split based on a distance metric. The previous step will repeat until each node forms a module.

After the dendrogram is constructed, a tree cut method is applied on the dendrogram to divide the graph into modules. The major advantage of hierarchical clustering is that the number of the modules are decided by the resolution parameter that is the tree cut height, while the number of modules has to be directly defined by the user in other methods.

Apart from the aforementioned clustering-based methods, recent studies such as AMOUNTAIN (Li et al., 2020) aim to identify modules in an attributed network. The attributes of the nodes are the expression level of the genes. The detected module is a sub-network which is not only densely connected, but also highly expressed.

1.3.2 Network embedding

As discussed in previous sections, a network can be naturally represented using a graph $G = (V, E)$. However, a major limitation of the graph representation is the in-applicability to the machine learning methods (Cui et al., 2018). The reason for the in-applicability is that most of the machine learning methods take independent vectors in a vector space as input. Although

the columns/rows of the adjacency matrix can be used as the input of the downstream machine learning tasks, there are two major drawbacks: first, the adjacency matrix is high dimensional; second, the adjacency matrix only contains first-order proximity.

Many network embedding methods have been proposed to learn the node representations from a network. Network embedding can be defined as: given a network $G = (V, E)$, the task is to learn a mapping function $Z = f(G)$, where $Z \in \mathbb{R}^{|V| \times d}$, and usually $d << |V|$. The $i-$th row of Z is the embedding for node v_i. For any node pair (v_i, v_j), the distance between Z_i and Z_j should reflect the topological structural similarity between them in G.

The network embedding methods, which focus on the preserving of network structure, are introduced in this section. The reason is that many network embedding methods focus on other research problems such as dynamic network embedding and attributed network embedding, yet, those methods are irrelevant to our research questions.

The first group is matrix factorisation based methods (Ou et al., 2016; Wang et al., 2017b). Those methods linearly map the graph representation of the network to the low dimensional embeddings. Many of those methods apply singular value decomposition due to its success in approximating the low rank eigenvectors (Ou et al., 2016).

The second group is random walk based methods (Perozzi, Al-Rfou, and Skiena, 2014; Grover and Leskovec, 2016). Those methods include two stages: the node sequence construction; the node embedding learning. In the node sequence construction stage, truncated random walks are simulated on a network to get node sequences. Then, the node embeddings are learned based on the node pairs generated from the node sequences. It is worth mentioning that those methods achieve outstanding results in preserving the topological structures (Goyal and Ferrara, 2018).

The third group is deep neural network based methods (Wang, Cui, and Zhu, 2016; Cao, Lu, and Xu, 2016). Inspired from the success of deep neural networks in other research fields such as computer vision, those methods build deep neural network models to learn the node embeddings. Those methods are capable of non-linear mapping the graph representation of the network to node embeddings.

1.4 Research questions

The multi-omics data has been introduced in Section 1.1, this section focuses on the research questions in those data.

1.4.1 The annotation of metabolites for LC-MS untargeted data

Metabolite annotation is the foundation of LC-MS metabolomics research. The annotation of metabolite is putatively matching the observed m/z of a peak to the candidate ionised products (Sumner et al., 2007). The problem of metabolite annotation can be defined as:

Problem 1. *Given a list of peaks with features (m/z RT, PIAS), the task is to detect a series of peak modules. Peaks in the same module should have a similar pattern in PIAS, similar RT and be annotated with ionised products from the same metabolite source, i.e., m/z patterns within the peak module can be explained by the neutral mass of the metabolite.*

The problem of metabolite annotation remains a bottleneck (Dunn et al., 2013), mainly due to the following three reasons.

First, more than one candidate ionised products can be assigned to a peak. Peaks from different metabolite sources could have extremely similar or identical theoretical m/z (Kind and Fiehn, 2006). Moreover, even the

adduct pairs have overlapping m/z differences, e.g., ($[MF+H]^+$, $[MF+Na]^+$) and ($[MF+ACN+H]^+$, $[MF+ACN+Na]^+$) where Na, ACN denote Sodium and Acetonitrile respectively.

Second, the number of peaks in a peak module is uncertain. Since the number of adduct types generated from a metabolite could be one or more (Scheltema et al., 2009, Dunn et al., 2013), and each adduct could be measured as multiple peaks due to the existence of isotopes, the number of peaks varies greatly between peak groups.

Third, the amount of metabolites in a biological sample is usually unknown. Hence no prior knowledge can be used to determine the number of peak groups in clustering. The situation is worsened as the sample may be contaminated (Keller et al., 2008), and some peaks may be artificial or falsely detected (Brown et al., 2011).

Over the years, many metabolite annotation methods have been developed. Most of the existing methods group peaks into non-overlapping peak modules based on the peak features. The peaks in the same peak modules are assumed to be from the same metabolite source. For each peak module, the candidate metabolites that could explain the m/z patterns within the module are assigned to the peaks.

However, the grouping of peaks will be suboptimal as the peak features may not be enough to distinguish peaks from different metabolite sources. Most of the methods group peaks based on RT and PIAS. For RT, metabolites could coelute from LC, hence peaks from multiple metabolites could co-exist in a small RT window. For PIAS, peaks could have similar patterns in PIAS as long as the volumes of metabolites have similar patterns across samples. As a result, both RT and PIAS similarity between peaks are necessary but not sufficient conditions for two peaks to be from the same metabolite.

Dividing the peaks into non-overlapping peak modules could lower the performance of metabolite annotation. Peaks from the same metabolite

source could be divided into different peak modules due to the aforementioned sub-optimal peak grouping. This would lead to some metabolites not annotated to any peaks even though the peaks observed from them are being detected by LC-MS.

Allowing the overlapping between the peak modules could overcome this problem as the different combination of peaks can be annotated, hence, all the potential adduct/isotopic patterns can be captured, thus lead to our first research question:

- Can we develop a metabolite annotation method that allows the overlapping of peaks between the peak modules for LC-MS untargeted data?

A method for peak annotation which adopts a module detection strategy on the attributed PSN is introduced. The details of the approach are described in Section 3.2.

1.4.2 Detection of network biomarker from gene expression data

Gene co-expression network (GCN) is a popular way to infer the gene functional and gene disease relations from the gene expression data. In a GCN, a node represents a gene and an edge between two nodes represents a pairwise functional relation. The edges in a GCN are constructed based on the similarity of gene expression across samples, and each edge is associated with an edge weight corresponding to the strength of their relation. Note that edges with very small weights are usually removed in the GCN, hence, a GCN may not be a fully connected network. Popular methods of constructing edges in a GCN include Pearson correlation, Spearman correlation and mutual information (Song, Langfelder, and Horvath, 2012).

The topological structures of GCNs could reflect real-world gene functions. In a GCN, two nodes connected by an edge indicate that two genes are in the same biological process, as they are activated or inhibited simultaneously, hence, a densely connected topological module suggests all of those genes are in the same biological process. The gene functional modules, gene disease modules and topological modules are assumed to be overlapping in the analysis of the GCNs (Barabási, Gulbahce, and Loscalzo, 2011).

Finding network biomarkers in a GCN has attracted a lot of research interest (Langfelder and Horvath, 2008; Yuan et al., 2018; Tanvir and Mondal, 2019; Jin and Qin, 2020; Tanvir et al., 2019). The network biomarkers are topological modules, in which the gene expression values correlate the sample labels. There are two advantages in using network biomarkers over using single genes: first, network biomarkers are more robust in separating samples from different categories (Jin et al., 2008); second, network biomarkers are better in interpreting the molecular mechanism behind the disease (Wu, Chen, and Wang, 2014).

The network biomarker methods can be categorised as a wrapper feature selection method (Guyon and Elisseeff, 2003). There are two stages in the detection of network biomarkers. First, the gene co-expression module detection. The gene co-expression modules are detected based on the topological structure of GCN. Second, the module significance measurement. For each gene co-expression module, the module significance is measured to quantify the level of association between the sample traits and the expression levels of genes. Hence, the problem of network biomarker detection can be written as:

Problem 2. *Given a gene expression matrix $X \in \mathbb{R}^{|V| \times b}$, where b is the number of samples, and a sample label vector $\gamma \in \mathbb{R}^{b}$, the task is to detect topological modules and measure their corresponding module significance scores. There should be two steps in detecting network biomarkers. The first step is to detect topological modules*

from the GCN constructed from the gene expression data. The second step is to measure the module significance score using the gene expression data and sample labels γ.

The previous methods have strict constraints of the shapes of topological modules. Based on the constraints on the detected topological modules, the existing methods can be divided into two groups. The first group of methods assume the topological modules should be spherical in shape (Langfelder and Horvath, 2008; Tanvir and Mondal, 2019). Those methods employ distance based clustering methods on a GCN to detect topological modules. Since nodes are grouped into different modules via minimising the average distance of nodes in the modules, those methods assumed that the genes in the same biological process are spherical in shape. The second group of methods assume genes in the same biological process should form cliques in the GCN (Shi, Derow, and Zhang, 2010; Tanvir et al., 2019). A clique is a topological module in which the nodes are fully connected.

Since the shapes of gene functional modules and gene disease modules may not meet the aforementioned constraints, e.g., the shapes of those modules could be oval or long strips, the previous methods may break up the genes belonging to the same biological process into different topological modules due to those constraints. This leads to our second research question:

- Can we relax constraints on the shapes of the topological modules in the detection of network biomarkers?

A network biomarker detection method that employs Gaussian mixture model clustering has been introduced. The details of the approach are described in Section 4.2.

1.4.3 Drug-target interaction prediction

Finding novel DTI is a crucial step in drug development. However, experimental determination of DTI is very expensive, it is highly desirable to develop computational methods to provide candidate DTIs to accelerate the speed of drug development.

Many DTI prediction methods focus on the prediction of DTIs for existing drugs. This is because a drug might bind with multiple targets, and this phenomenon could lead to successful drug development (Xia et al., 2010). Many previous methods take the known DTIs, DSSN, and TSSN as input. The task of DTI prediction can be regarded as to learn a prediction matrix M to reflect the possibility of existence of the DTIs, i.e., a smaller value indicates there is no regulation between a drug-target pair, while a large value indicates the target is likely to be regulated by the drug. Hence, the problem of DTI prediction can be defined as:

Problem 3. *Given DTI network $G^B = (V^D, V^T, E^{DTI})$, DSSN and TSSN, the method should infer the DTI prediction matrix $M \subseteq V^D \times V^T$ for all drug-target pairs. M should reflect the possibility of existence of the DTIs, i.e., a smaller value indicates there is no regulation between a drug-target pair, while a large value indicates the target is likely to be regulated by the drug.*

The DTI network is a bipartite network that is introduced in Section 1.2.1, as there is no drug-drug interaction and target-target interaction in the DTI network. From the DTI network (a bipartite network), two implicit networks can be extracted. The implicit networks, i.e., drug implicit network (DIN) and target implicit network (TIN), are constructed using the second-order proximity in the DTI network. It is worth noticing that, an edge in the DIN indicates that two drugs would bind to at least one common target and an

edge in the TIN indicates that two targets would bind to at least one common drug, while DSSN (or TSSN) represents the structural similarity directly calculated between two drugs (or two targets).

Although many computational methods have been proposed to tackle the DTI prediction problem, the implicit networks constructed from the known DTI network are ignored in previous works. However, it has been shown the implicit networks constructed from a bipartite network can improve the performance of the link prediction task in the bipartite network (Wang et al., 2019). Besides, the implicit relations can also improve the performance of recommender systems (Yu et al., 2018). This leads to our final research question:

- Can we incorporate the implicit relations obtained from the DTI network to predict novel DTIs?

A network embedding based method for novel DTI prediction, which incorporates the implicit relations from the DTI network, is introduced. The details of the approach are described in Section 5.2.

1.5 Thesis contributions

The main purpose of this work is to apply graph-based machine learning on the research problems in multi-omics data. The contributions of this thesis can be summarised as:

- We propose a novel algorithm to annotate the peaks for LC-MS untargeted data with a module detection strategy that allows overlapping between peak modules. The proposed approach can out-perform the state-of-the-art methods on the most experimental settings. (Chapter 3)

- We propose a network biomarker detection method that allows high freedom of the shapes of the topological modules. From the experiments, the proposed framework shows competitive performance compared against the most popular approach across several datasets. The gene enrichment analysis suggests that network biomarkers detected by the proposed approach can be annotated with known gene functions that are related to cancer. (Chapter 4)
- We propose a network embedding based on the method to predict novel DTIs while incorporating the implicit relations from the DTI network. The proposed method shows competitive performance compared with state-of-the-art methods and the novel DTIs predicted by our approach can be supported by recent studies. (Chapter 5)
- Based on methods proposed in this thesis, we have developed the following open-source software:
  - MAMD (`https://github.com/BrisksHan/MAMD`), this is the implementation for the metabolite annotation method for LC-MS untargeted data. The details of this method are introduced in Chapter 3.
  - NBD-GMMC (`https://github.com/BrisksHan/NBD-GMMC`), this is the implementation for the network biomarker detection for gene expression data. The details of this method are introduced in Chapter 4.
  - NE-DTIP (`https://github.com/BrisksHan/NE-DTIP`), this is the implementation for the DTI prediction method. The details of this method are introduced in Chapter 5.

1.6 List of publications

This thesis summarises the contributions of the following papers:

- MAMD: Metabolite Annotation via Module Detection. Paper in preparation.

- Network Biomarker Detection from Gene Co-expression Network Using Gaussian Mixture Model Clustering. Paper submitted to IEEE/ACM transactions on computational biology and bioinformatics.

- Han Zhang et al. (2021). "A Network Embedding Based Approach to Drug-Target Interaction Prediction Using Additional Implicit Networks". In: *International Conference on Artificial Neural Networks*. Springer, pp. 491–503

I have been fortunate enough to be able to collaborate with many researchers in other projects. However, those works are beyond the scope of this thesis, the publications from those collaborations are:

- Chengbin Hou et al. (2020). "GloDyNE: Global Topology Preserving Dynamic Network Embedding". In: *IEEE Transactions on Knowledge and Data Engineering* (I was deeply involved in designing this method, second author)

- Chengbin Hou et al. (2022). "GloDyNE: Global Topology Preserving Dynamic Network Embedding (Extended Abstract)". In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)* (I was deeply involved in designing this method, second author)

- Waleed Fateen et al. (2021). "Characterisation of dysplastic liver nodules using low-pass sequencing and the proof of principle on liquid biopsies". In: *Journal of Pathology* (I conducted the bioinformatic analysis, fourth author)

1.7 Thesis organisation

This thesis consists of six chapters, and focuses on the application of graph-based machine learning on multi-omics data analysis. The main contents of this thesis is shown in Figure 1.3.

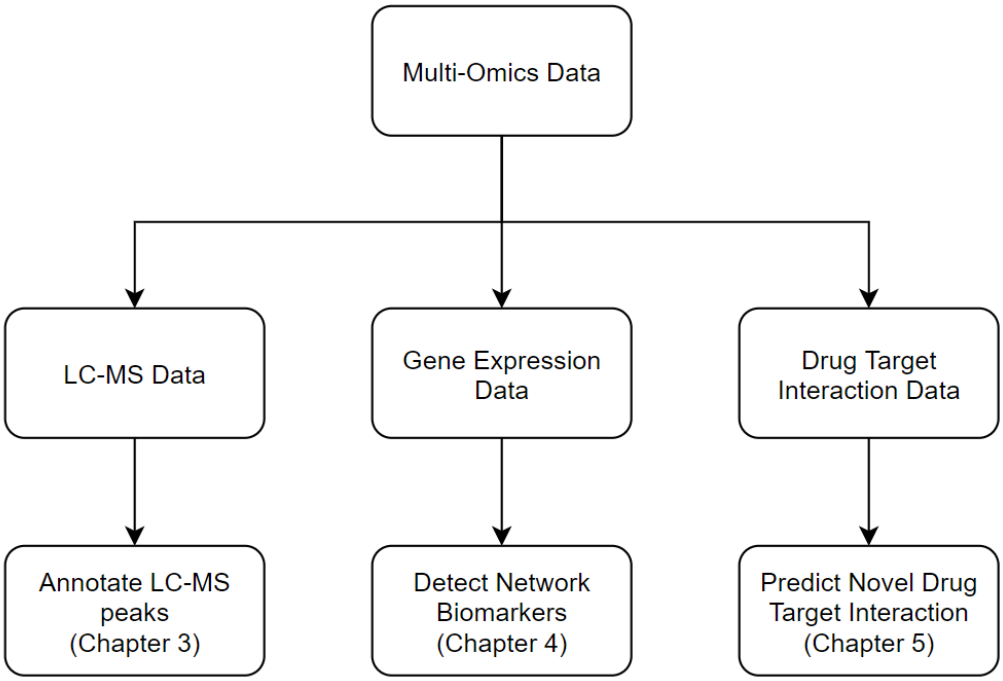


FIGURE 1.3: The main contents of this thesis.

In chapter 1, we introduce the background about multi-omics data and their corresponding network representations. Then, we discussed the graph-based machine learning methods. After that, the research questions and main contributions are discussed.

In Chapter 2, we review the important literature in related fields. In Section 2.1, the methods for metabolite annotation on LC-MS untargeted data are discussed. Then, Section 2.2 focuses on the methods in the detection of the network biomarkers. Finally, Section 2.3 focuses on the methods in predicting novel DTIs.

In Chapter 3, we study the problem of metabolite annotation in LC-MS untargeted data. Section 3.1 describes the intuition behind our approach.

Section 3.2 proposes a module detection based approach to annotate peaks. Our approach is validated on a number of real-world benchmark LC-MS datasets in Section 3.3.

In Chapter 4, we study the problem of network biomarker detection in gene expression data. Section 4.1 introduces the subject of network biomarker detection and the research gap. We propose a method to identify network biomarkers in Section 4.2. In Section 4.3, we first evaluate the module significance of the detected biomarkers. Then, the gene enrichment analysis is conducted to find whether there are known cancer related biological functions in the detected network biomarkers.

In Chapter 5, we propose a novel method for identifying the DTIs. Section 5.1 provides the motivation for this work. Section 5.2 describes the method. Section 5.3 compares the proposed method against the state-of-the-art methods. Moreover, we conduct literature reviews to find the support for the novel DTI predictions.

Chapter 6 concludes the thesis, discusses the limitations of the proposed methods, and proposes several future directions for those research problems defined in Section 1.4.

Chapter 2

Literature review

This chapter reviews the literature relevant to the three aforementioned research problems discussed in Section 1.4. Section 2.1 reviews literature in metabolite annotation on LC-MS untargeted data. Section 2.2 reviews the previous methods in network biomarker detection from gene expression data. Section 2.3 reviews the previous methods in DTI prediction. Finally, Section 2.4 summarises this chapter.

2.1 Metabolite annotation

The problem of metabolite annotation has been defined in Problem 1 in Section 1.4.1. Over the years, many metabolite annotation methods for LC-MS untargeted data have been developed. There are usually two steps in the annotation of peaks. The first step is the grouping of peaks. The second step is the annotation of peaks.

Based on the strategies for peaks grouping, the previous metabolite annotation methods can be divided into three categories: the peak-pair based methods; network based methods; probability based methods. This section gives a review on the methods in all three categories.

2.1.1 The peak-pair based methods

The peak-pair based methods annotate peaks based on a set of rules (Brown et al., 2011; Alonso et al., 2011). Those methods group peak pairs based on the assumptions about peaks from the same metabolite source. Then, for each peak pair, the theoretical ionised products are annotated to the peaks based on a set of built-in rules.

The grouping of peaks is based on the features of the peaks. As discussed in Section 1.1.1, peaks from the same metabolite source should have similar RT and similar patterns in PIAS. In those methods, for all peak pairs, the difference between RT and similarity between PIAS are calculated. The similarity between PIAS is measured using Pearson's/Spearman's correlation coefficient. A peak pair will be stored, if the two assumptions are met. First, the RT difference is below a user defined threshold. Second, the correlation coefficient between PIAS is above a user defined threshold. After the peak pairs are constructed, the peak pairs are annotated with the following steps.

The first step is isotopic pattern annotation. Isotopic patterns are highly reliable and easy to annotate. Those methods focus on detecting the isotopic patterns generated by the atom *carbon*. The Isotopic patterns between the atom *carbon* are frequently measured in LC-MS for two reasons. 1) Both $carbon - 13$ and $carbon - 14$ are stable in nature. 2) Atom *carbon* exists in all metabolites. For a peak pair, the following criteria must be met for them to be annotated as an isotopic pattern. First, the m/z difference is very close to $1.0033/z$ which is based on the mass difference between $carbon - 13$ and $carbon - 14$. Second, the median peak intensity of the peak assumed to be generated from $carbon - 13$ should be higher than that of the peak assumed to be generated from $carbon - 14$. The reason is that $carbon - 13$ is more frequently observed than $carbon - 14$.

The second step is the single-charged adduct annotation. The m/z difference between the single-charged ionised products can calculated without considering the mass of the metabolite e.g., the theoretical m/z of $[MF+H]^+$ is MF+1.007276 and that of $[MF+Na]^+$ is MF+22.989218, as $z = 1$ in those ionised products, the expected m/z difference between those two ionised is 21.981942. With a built-in rule of m/z difference between the ionised products, the single-charged adducts can be annotated to the peak pairs.

The third step is adduct annotation. The m/z difference between those ionised products can not be calculated based on the forms of adducts e.g., the theoretical m/z of $[2MF+H]^+$ is (2MF + 1.007276)/1 and that of $[MF+2Na]^{2+}$ is MF/2+22.989218, as a result, the mass of MF would affect the m/z difference between those ionised products. To annotate those ionised products, those methods calculate the neutral mass of the single-charged peaks. Then, for the node pairs with one peak being annotated as a single-charged ionised product, the complex adducts are annotated based on the neutral mass of the annotated single-charged ionised products and a set of built-in complex adduct rules.

Finally, for each annotated peak, the candidate MFs within tolerance will be annotated to it. It is worth mentioning that many metabolites could have the same MF this is because of the existence of isomers. In practice, all the corresponding metabolite names will be annotated to a peak alongside the MF.

The main advantage of those methods is that they are very fast in annotating the peaks. However, there are two drawbacks in those methods worth mentioning. The first drawback of those methods is that the uncertainty of peak annotation is ignored. In those methods, at most one ionised product is annotated to a peak, however, the annotation of ionised procuts is uncertain as discussed in Section 1.4.1. Ignoring the uncertainty of peak annotations, some metabolites may not be annotated at any peak even though the peaks

generated from them are observed by LC-MS. Although, a simple solution is to output all the matched ionised product pairs, the output will be hard to interpret due to the large quantity of annotations. The second drawback is that there could be RT drift in the annotation process. The RT difference means the RT difference in a module could be beyond the RT limit. If the peak pairs are annotated with adducts from the same metabolite source and there is overlapping of peaks between those peak pairs, those peaks are assumed to form a peak module. However, there is only a pairwise RT difference limit, the RT difference between two peaks that share a common neighbour could exceed the user defined threshold. This would violate the assumption that the peaks from the same metabolite source should have a small RT difference.

2.1.2 The network based methods

The second category of methods construct a PSN to represent the pair-wise relations between the peaks. As discussed in Section 1.4.1, in a PSN, a node represents the peak and an edge indicates that the two connected nodes could be from the same source (Kuhl et al., 2011; Uppal, Walker, and Jones, 2017; Senan et al., 2019). Those methods assume that peaks from the same metabolite should form a topological module that is densely connected in the PSN. In each peak module, the peaks are assumed to be from the same metabolite source. The MF(s) that can explain the m/z patterns in the module are annotated to the peaks. The general framework of those approaches is formulated in Algorithm 1.

CAMERA (Kuhl et al., 2011) is the first network based method in annotating metabolites. CAMERA takes the raw LC-MS output as input to annotate peaks. There are roughly four steps in the annotation process.

The first step of CAMERA is the partition of peaks into different compound spectrum to avoid RT drift. CAMERA is biased to the annotation of

Algorithm 1 The general framework of network based peak annotation

Input: peaks and their features
Output: peak annotation list.

1: construct PSN based on peak features.
2: detect peak modules from the PSN.
3: **for** each peak module **do**
4: annotate the isotopes in the peak modules.
5: annotate the adducts in the peak modules.
6: annotate the molecular formulas and metabolite names.
7: **end for**
8: **return** the peak annotations.

the most abundant peaks, as peaks with low PIAS could be artifact peaks that are peaks falsely detected in the pre-processing step (Brown et al., 2011). Based on this, CAMERA first identifies the peak with the highest peak intensity, and a RT window is constructed around the RT of it. Then, all the peaks within this RT window are assigned to a compound spectrum. After that, CAMERA searches for the next most abundant peak without being assigned to a compound spectrum and constructs a new RT window around it, the peaks in this new RT window that have not been assigned to a compound spectrum will be assigned to a new compound spectrum. The aforementioned steps are repeated until all the peaks are assigned to a compound spectrum.

The second step of CAMERA is to search isotope pairs in each compound spectrum. Specifically, CAMERA searches the peaks pairs with m/z difference is $1.0033/z$. Then construct a ISO matrix. In the ISO matrix, each element ISO_{ij} describes the isotopic pattern between them, i.e., $ISO_{ij} = 0$ indicates this node peak is an isotopic pattern and $ISO_{ij} = 0$ otherwise.

After that, CAMERA constructs a PSN $G = (V, E)$ in each compound spectrum. The edge weights in the CAMERA can be computed with following function:

$$w_{ij} = corr(X_i, X_j) + ISO_{ij} + 1/N \sum_{n=1}^{N} CPS_{nij} \tag{2.1}$$

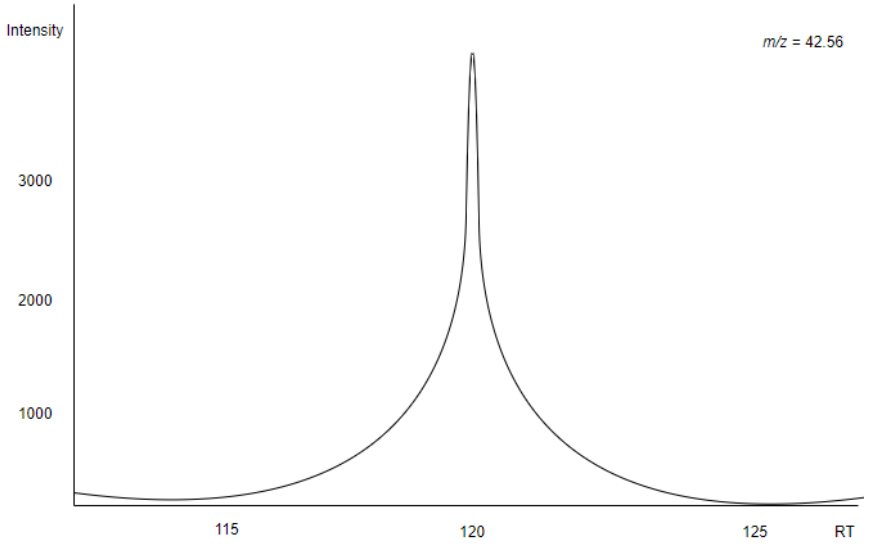


FIGURE 2.1: An example of a peak shape with $m/z = 42.56$. The horizontal axis is the RT, and the vertical axis is the intensity of the peak

where N is the number of samples, $X_i = \{\tau_1^i, ..., \tau_{|N|}^i\}, X_j = \{\tau_1^j, ..., \tau_{|N|}^j\}$ represents the PIAS for node v_i, v_j, $corr(X^i, X^j)$ represents the correlation coefficient between X_i, X_j, the CPS_{nij} represents peak intensity shape correlation coefficient for the $n-th$ sample between node pair (v_i, v_j), and example of peak shape is shown in Figure 2.1. The correlation measurement between vectors is usually Pearson's correlation, and for X_i and X_j, the Pearson's correlation coefficient can be calculated as:

$$corr(X_i, X_j) = \frac{\sum_{n=1}^{|N|}(\tau_n^i - \bar{\tau}^i)(\tau_n^j - \bar{\tau}^j)}{\sqrt{\sum_{n=1}^{|N|}(\tau_n^i - \bar{\tau}^i)}\sqrt{\sum_{n=1}^{|N|}(\tau_n^j - \bar{\tau}^j)}} \tag{2.2}$$

On each PSN, a community cut algorithm (Hartuv and Shamir, 2000) is applied to detect peak modules. In each peak module, all the peaks are assumed to be from the same metabolite. Similar to the peak-pair based methods, CAMERA applies a rule set to annotate the peaks. In a peak module, the MF that could explain most of the measured m/z is annotated to the peak module.

CAMERA is capable of providing highly precise annotation results. The reason is that the CAMERA focuses on the annotation of the most reliable

peaks. However, the recall of CAMERA is relatively low compared against other methods (Daly et al., 2014). There are mainly two reasons for this. First, the construction of the compound spectrum is based around the most abundant peaks, as a result, the peaks from the same metabolite could be divided into different compound spectrum. Second, for each peak, there is only one annotation. However, as discussed in Section 1.4.1, annotation of peaks is uncertain. As a result, annotating a single metabolite to a peak module would also lead to low recall.

xMSannotator (Uppal, Walker, and Jones, 2017) is also a popular method for annotating peaks. In xMSannotator, PSN $G = (V, E)$ contains all the measured peaks without having to divide peaks into a series of RT windows. For v_i and v_j, the edge weight can be computed with:

$$w_{ij} = \begin{cases} corr(X_i, X_j) & \text{if } corr(X_i, X_j) > \alpha \text{ and } |r_i - r_j| < \beta \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

where $X_i = \{\tau_1^i, ..., \tau_{|N|}^i\}, X_j = \{\tau_1^j, ..., \tau_{|N|}^j\}$ represents the PIAS for node v_i, v_j, $corr(X_i, X_j)$ represents the Pearson's correlation coefficient between X_i and X_j, r_i and r_j represents the RT for node v_i and v_j respectively, $\alpha \in [-1, 1)$ is the correlation coefficient threshold and $\beta \in (0, +\infty)$ is the RT threshold. Then, on the PSN, the module detection stage of WGCNA (Langfelder and Horvath, 2008) is applied to detect the peak modules. WGCNA is also a popular framework in detecting network biomarkers and it will be introduced in detail in Section 2.2.

Similar to CAMERA, peaks in the same module are assumed to be from the same metabolite source. From the candidate metabolites, the candidate ionised products are constructed. Then, in a peak module, the candidate ionised products are putatively annotated to peaks via matching the similarity between the m/z of the measured peaks and that of the candidate ionised

products. For an annotated peak module, an annotation score is introduced to measure the confidence of the annotation and it is defined as:

$$\begin{aligned} score &= (\text{isotope weight}) \times (\text{correlation}) \times (1/(\text{RTrange})^{\sigma}) \\ &= \times(\#\text{matchedAdduct} + \#\text{matchedIsotope} \times 10 \times \sum(\text{adduct Weight}) \end{aligned} \tag{2.4}$$

where # represents number, correlation represents the highest Pearson's correlation coefficient of PIAS in a peak module, σ = {1 if RT range of the peak module < β, 10 otherwise}, the adduct weights and isotope weights are based on the types of the adducts and the isotopes, i.e., the frequently observed adducts and isotopes will be assigned with high scores.

After the annotation scores are assigned to the peak modules. The confidence scores of peak module annotations are categorised into different levels. The confidence level is computed based on the following factors: the annotation score; the presence of isotopes; abundant ratio checks for the isotopes; the ratio checks of the atoms; the detection of other metabolites that are likely to co-existence.

The advantage of xMSannotator compared against CAMERA is that an annotation score is associated with each module. In real-world practice, the researchers can focus on the high confidence annotations, and ignore the low confidence annotations. Another advantage is that the metabolite pathway information is incorporated to further increase the annotation confidence of the peak modules.

The aforementioned two methods require PIAS to compute the edge weights. However in some experiments, only one sample is profiled by LC-MS, instead of multiple samples. As a result, those methods could not cope with the data. In order to address this issue, CliqueMS is developed.

In CliqueMS, the edge weights of PSN are computed using the cosine similarity of the chromatography shape of the peaks. An example of the

chromatography shape of a peak is shown in Figure 2.1. The peak shape in the chromatography is divided into K bins. Then, cliqueMS calculates the cosine similarities between the peaks based on the following function:

$$\cos_{ij} = \frac{\sum_{k=1}^{K} f_i(t_k) f_j(t_k)}{||f_i|| ||f_j||} \tag{2.5}$$

where $f_i(t_k)$ represents the peak intensity of peak i at time t_k, $||f_i|| = \sqrt{\sum_{k=1}^{K} (f_i(t_k))^2}$. For all peak pairs, the similarity scores are computed without dividing the peaks into different RT windows.

In CliqueMS, the peaks from the same metabolite source are assumed to be connected in the PSN. ClqiueMS adopts a heuristic strategy to search the cliques in the PSN. Each clique is regarded as a peak module.

In a peak module, there are two steps in the annotations of peaks. The first step is the identification of isotopic patterns. The CliqueMS searches all the m/z difference between the peaks to match with the built-in m/z difference of isotopes. The second step is to detect the most likely annotations. Unlike the other aforementioned methods, CliqueMS assumes the peaks in a module could be from multiple sources, the reason is that there could be a lack of PIAS to distinguish samples from different metabolite sources. A scoring function is employed to evaluate the confidence of the combinations of the annotations in a peak module. The annotations are ranked based on their scores and the top a annotations are regarded as reliable. Hence, for a peak, there is a ranked list of the potential peaks.

Apart from the aforementioned methods, there are other methods that adopt a network-based strategy to annotated peaks, e.g., ramcluter (Broeckling et al., 2014). However those methods focus on the annotations of peaks profiled by other instruments such as tandem MS, as a result, those methods are beyond the scope of this thesis.

2.1.3 The probability based methods

The third category of methods build a probability model to annotate peaks. (Rogers et al., 2009; Rogers et al., 2011; Silva et al., 2014; Daly et al., 2014; Del Carratore et al., 2019). In those methods, the marginal probability of assigning the candidate ionised products to the peaks are defined. Then, Gibbs sampling is employed to infer the peak annotations.

Rogers et al., 2009 pioneered the field using a probability model to annotate peaks, this method takes the measured m/z and the biochemical reactions to annotate peaks. Later, Rogers et al., 2011 extended the method in Rogers et al., 2009 by incorporating additional information such as RT. An implementation of those two methods is introduced in Probmetab (Silva et al., 2014). In the aforementioned methods, all the peaks are required to be annotated with MFs from the candidate MF list for the annotation process to work. However, the molecules in a biological sample are usually unknown. To address this issue, MetAssign assumes the peaks are generated from the Dirichlet process, then Gibbs sampling is used to infer the posterior distribution of the peaks. The posterior distribution of peaks includes the grouping and the annotation of peaks. In MetAssign, the biochemical connections are ignored, to address this issue, IPA (Del Carratore et al., 2019) is introduced. This method is similar to MetAssign, and the major advantage of IPA is the incorporation of biochemical connections.

2.1.4 Metabolite annotation evaluation criteria

The common way of evaluating the performance of the metabolite annotation methods is to annotate the peaks from datasets with known compounds. Unfortunately, the metabolites are unknown in the real world biological samples. To address this issue, some synthetic samples with known compounds are profiled by LC-MS as benchmark datasets.

The most popular benchmark datasets are standard mixture datasets (Daly et al., 2014). The standard mixture datasets are generated via profiling the samples with known mixture of compounds. There are three compound mixtures that are Std1, Std2 and Std3. The numbers of metabolites in the mixtures are 104, 96 and 40, respectively. Since these samples are measured in both positive mode and negative mode in LC-MS, there are six datasets in total, namely Std1.POS, Std1.NEG, Std2.POS, Std2.NEG, Std3.POS and Std3.NEG. The number of samples is three in all of the datasets.

An annotation algorithm should be able to identify the metabolites which are in the sample and reduce the false annotation. In order to evaluate the performance of algorithms, known compounds which are metabolites known to be in the sample and decoy metabolites which are not in the sample are added to the candidate metabolite list in the experiments. Following previous works (Daly et al., 2014; Uppal, Walker, and Jones, 2017), the annotation performance is evaluated as:

$$\text{Precision} = \frac{\text{number of annotated and in the sample}}{\text{total number of annotations}} \tag{2.6}$$

$$\text{Recall} = \frac{\text{number of annotated and in the sample}}{\text{total number of metabolites in the sample}} \tag{2.7}$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.8}$$

It is worth mentioning that the peak modules annotated with the same metabolite will be treated as a single annotation in calculating the precision, recall and F1 score. The reason is that for real world datasets, the labels of the peaks are not available.

2.2 The detection of network biomarker

The problem of network biomarker detection has been defined in Problem 2 in Section 1.4.2. Many methods have been developed to tackle the research problem. There are usually two stages in the detection of network biomarkers. Those stages are gene co-expression module detection and module significance measurement. The framework of those methods are summarised in Algorithm 2.

Based on the difference in gene co-expression module detection, the methods can be categorised into the following two categories: clustering based methods; clique detection based methods. The literature review will be conducted in both of the categories.

Apart from the aforementioned two stages, the gene enrichment analysis has also been used to detect the known gene interactions in the gene co-expression modules. Hence, the gene enrichment analysis will also be introduced in this section.

Algorithm 2 The general framework of network biomarker detection

Input: gene expression data, sample labels

Output: gene modules and the module significance scores.

1: construct a GCN based on gene expression data.
2: detect gene co-expression modules from the GCN.
3: **for** each gene co-expression module **do**
4: calculate the module significance score
5: **end for**
6: **return** the gene co-expression modules with module significance scores.

2.2.1 Clustering based methods

The node clustering based methods treat the problem of gene co-expression module detection as a clustering problem. Yuan et al., 2018 and Jin and

Qin, 2020 apply WGCNA (Langfelder and Horvath, 2008) to detect network biomarkers.

WGCNA is the most popular tool in analysing gene expression data. WGCNA takes the gene expression data and the sample traits as input. The first step is the construction of GCN $G = (V, E)$. In a GCN, each node indicates a gene and the weights between genes can be computed using:

$$w_{ij} = |corr(X_i, X_j)|^{\beta} \tag{2.9}$$

where X_i and X_j represent the expression of the genes i and j respectively i.e., the $i-th$ row and $j-th$ row in Matrix (1.2), $corr(X_i, X_j)$ represents the Pearson's correlation coefficient between X_i and X_j, $\beta \in (1, +\infty)$ is the soft threshold, where $corr(X_i, X_j)$ represents the Pearson's correlation between X_i and X_j. The function of the soft-threshold is to make the weight of the strong relations much larger than that of the weak relations, as the weak relations are not informative. In the tutorial of WGCNA, β can be selected based on the number of samples, and β can also be tuned to make the distribution of the edge weights scale-free (Langfelder and Horvath, 2008).

On the GCN, hierarchical clustering is applied on the GCN to detect gene co-expression modules. Since the hierarchical clustering requires the node distance to construct a dendrogram, the topology overlapping matrix (TOM) (Ravasz et al., 2002) is used to measure the similarity between genes in a GCN. The elements in TOM can be calculated as:

$$TOM_{ij} = \frac{l_{ij} + w_{ij}}{min\{k_i, k_j\} + 1 - w_{ij}} \tag{2.10}$$

where $l_{ij} = \sum_{v_u \in V} w_{iu} w_{ui}$ and $k_i = \sum_{v_u \in V} w_{iu}$. It can be seen that $Eq.$ (2.10) calculates the similarity between two nodes based on the first and second order proximity in the GCN.

Since TOM measures the similarity rather than distance between nodes, 1-TOM is used as the input for hierarchical clustering to construct a dendrogram. Finally, WGCNA applies dynamic tree cut (Langfelder, Zhang, and Horvath, 2008) or constant tree cut on dendrogram to detect gene co-expression modules.

WGCNA applies the eigengene significance to measure the module significance. As discussed in Section 1.4.2, there is a need to learn the low dimensional representation for the samples. For a detected gene co-expression module, WGCNA uses eigengene that is the first principle component of a gene expression matrix to represent the relations between the samples. The Pearson's correlation coefficient between the eigengene and sample labels is used to measure the module significance.

Tanvir and Mondal, 2019 aims to identify the conserved network biomarkers from multiple datasets. There are three steps in this method: the module detection; the module conservation evaluation; the module's significant measurement.

In this method, the edge weights in the GCN is computed as:

$$w_{ij} = \begin{cases} corr(X_i, X_j) & \text{if } corr(X_i, X_j) \geq \alpha \\ 0 & otherwise \end{cases} \tag{2.11}$$

where $\alpha \in [-1, 1)$ is the hard threshold, X_i and X_j represent the expression of the genes i and j respectively i.e., the $i-th$ row and $j-th$ row in Matrix (1.2), $corr(X_i, X_j)$ represents the Pearson's correlation coefficient between X_i and X_j. It is worth mentioning that the GCNs from multiple datasets are constructed in this method.

Then, six established module detection methods are applied on all the GCNs to detect gene co-expression modules. The module detection methods includes: Fastgreedy (Clauset, Newman, and Moore, 2004), Infomap (Rosvall

and Bergstrom, 2007), Label Propagation (Raghavan, Albert, and Kumara, 2007), Leading Eigenvector (Newman, 2006), Multilevel (Blondel et al., 2008) and Walktrap (Pons and Latapy, 2005).

Then, the module conservation evaluation is conducted on the detected gene co-expression modules. For a gene co-expression module $G^A = (V^A, E^A)$ detected from cancer A, there are five steps in calculating the conservation significance of this module. First, a GCN for cancer B is constructed, and it is denoted as $G^B = (V^B, E^B)$. Second, on G^B, a gene co-expression module using genes in V^A is constructed, and the number of edges in this module is denoted as e_1. Third, on G^B, $|V^A|$ genes are randomly selected to construct a gene co-expression module, and the number of edges in this module is denoted as e_2. Fourth, the third step is repeated for 1000 times, and then count the time of occurrence that e_2 is bigger than e_1. The count is denoted as T. Finally, the conservation significant value is calculated as:

$$P = \frac{T}{1000} \tag{2.12}$$

The gene co-expression modules with P-value smaller than 0.05 are ignored as they are not regarded as conserved across different GCNs.

In Tanvir and Mondal, 2019, the module significance of the conserved gene co-expression modules is measured using Cox proportional-hazard regression model (CPR) (Cox, 1972). This model aims to predict the survival rate of the patients. The decision function of CPR can be written as:

$$CPR = \lambda(t)exp(\beta S_i) \tag{2.13}$$

where $\lambda(t)$ is the baseline hazard function, S_i is the vector representing the gene expression for sample i, and β is the vector of the parameters.

In Tanvir and Mondal, 2019, the samples are divided into a training group and a testing group. The training of *Eq*. (2.13) is based on gene expression

from the training group. After that, the learned decision function is applied on the testing group to evaluate the prediction performance of the model. The gene co-expression modules with high hazard prediction performance will be regarded as network biomarkers.

2.2.2 Clique based methods

The clique based methods (Shi, Derow, and Zhang, 2010; Tanvir et al., 2019) detect the cliques from the GCN in contrast to the node clustering methods.

Shi, Derow, and Zhang, 2010 applies *Eq*. (2.11) to construct the edge weights in a GCN. After that, an iterative clique enumeration algorithm is applied to detect gene co-expression modules. Starting from the original GCN $G = (V, E)$, it detects the largest clique G^c in G and G^c is stored as a gene co-expression module. Then, the G^c is removed from the G. The aforementioned step is repeated until there is no nodes remaining in the GCN.

After the gene co-expression modules are detected, Shi, Derow, and Zhang, 2010 applies the Jonckheere-Terpstra test (Lehmann and D'Abrera, 1975) to evaluate the p-value of the null hypothesis. The null hypothesis of the test is that there is no significant difference between the gene expression among the categories. The gene co-expression modules that can reject the null assumption would be regarded as network biomarkers.

Tanvir et al., 2019 detect the clique-bipartite-like modules as the gene co-expression modules. The assumption behind this method is that the disease signal of genes from different cancer stages would form different gene co-expression modules. As a result, Tanvir and Mondal, 2019 explores a combination of cliques as network biomarkers.

In this method, the edge weights in the GCN are computed using *Eq*. (2.11). On the GCN, the single clique modules are detected using Bron and Kerbosch, 1973. Based on the inter-clique edges between the single clique

modules, the 2-clique-1-bipartite modules are detected. After that, based inter-module edges from the newly discovered 2-clique-1-bipartite modules, the 3-clique-2-bipartite modules are also discovered. The module significance measured is also conducted using CPR.

2.2.3 Gene enrichment analysis

Over the years, many gene enrichment analysis methods have been developed. The functional gene enrichment analysis of gene modules is to annotate them with the gene ontology (GO) terms (Botstein et al., 2000) and KEGG terms (Kanehisa and Goto, 2000). In this thesis, we do not examine the difference between the GO term and the KEGG term. Usually, the p-value of the annotation of GO terms and KEGG terms is used to quantify the confidence of the annotation. From Huang, Sherman, and Lempicki, 2009, the existing gene enrichment analysis tools could be categorised into three groups based on their gene function annotation algorithm. It is worth mentioning that many gene enrichment analysis tools could be categorised into multiple groups as there are multiple built-in gene function annotation algorithms.

The first group is singular enrichment analysis methods. Those methods take a selected gene list and the corresponding background gene list as input. For each term, the enrichment p-value is calculated based on the frequency of genes in the same function compared against that of the randomly selected genes in the background gene list. Popular methods in this group include: GoMinor (Zeeberg et al., 2003), David (Huang et al., 2007).

The second group is modular enrichment analysis methods. This strategy can be regarded as an extension of the singular enrichment analysis methods. Those methods are capable of incorporating the term–term and gene–gene relationship in calculating the p-values of the gene function terms. The popular methods include: David (Huang et al., 2007), StringDB (Szklarczyk et al.,

2015), Enrichr (Kuleshov et al., 2016), Metascape (Zhou et al., 2019).

The third group is gene set enrichment analysis methods. Unlike the methods in the first group, all the genes and the expression values are taken as input for the gene enrichment analysis. Those methods differ from the methods in the other two groups in the following aspects. First, genes are not divided into different groups. Second, the expression values of genes are also taken into consideration in calculating p-values of the gene function terms. Popular methods include GESA (Subramanian et al., 2007), PAGE (Kim and Volsky, 2005).

2.3 Drug-target interaction prediction

The problem of DTI prediction has been defined in Problem 3 in Section 1.4.3. Many methods have been developed for predicting novel DTIs. In this section, we will give a literature review on the related works in the problem of DTI prediction.

2.3.1 Target structure based methods

The target structure based methods (Keiser et al., 2009; Morris et al., 2009) simulate the docking process of drugs. Those methods take the 3D structures of the targets, i.e., proteins, and the drug structures as input. Based on those inputs, the probability of a drug docking with a protein is simulated, based on which, the novel DTIs are predicted. However, the 3D structures of the proteins are required, yet, the 3D structures of many proteins are unavailable due to its complexity. This greatly limits the applicability of those methods. And those methods are very time consuming.

2.3.2 Ligand similarity based methods

Ligand similarity based methods (Keiser et al., 2007; Keiser et al., 2009) use the structural similarity between ligands to predict interactions. However, the sequence similarities between targets are ignored in predicting DTIs.

2.3.3 Machine learning based methods

Machine learning based DTI prediction methods have attracted a lot of research interest in recent years. The advantage of those methods is that the prior knowledge from both drugs and targets is utilised in predicting DTIs. Those methods usually take the DTI network, drug structural similarity network (DSSN) and target sequence similarity network (TSSN) as inputs, and some methods may take the structural information about drugs and targets as inputs. Based on the difference in the machine learning models, the previous methods can be categorised as: embedding based methods; bipartite local model based methods; feature vector based methods.

Embedding based methods

The embedding based methods aim to map drugs and targets into a unified space to predict DTIs. Yamanishi et al., 2008 learns a mapping function for drugs and a mapping function for targets. Then, the drugs and targets are projected into a unified space. From which, the novel DTIs are predicted based on the drug-target distance. NRLMF (Liu et al., 2016) applies logistic matrix factorisation method to learn the embeddings of drugs and targets. Neighbour information is also incorporated to regularise the embeddings. Tripartite-DeepWalk (Zong et al., 2017) applies DeepWalk (Perozzi, Al-Rfou, and Skiena, 2014) on the a linked tripartite network constructed from drug-target, target-disease and drug-disease networks to learn the embeddings,

then uses the distance of embeddings between drugs and targets to make DTI predictions.

Bipartite local model based methods

There are also many methods that adopt a bipartite local prediction strategy. Those methods transform the DTI prediction problem to a DTI edge classification problem. Bipartite local models (Bleakley and Yamanishi, 2009) learns two classifiers. The first one predicts targets for a given drug and the second one predicts drugs for a given target. Then, the two classifiers are joined together to make a prediction. To address the issue that all known edges are treated as negative samples, NetLapRLS adopts a semi-supervised strategy to learn BLM. To handle unseen drugs/targets i.e., drugs/targets without known DTIs, BLM-NII (Mei et al., 2013) is proposed. BLM-NII integrates neighbour-based interaction-profile inferring to train the classifiers.

Feature vector based methods

Recent methods construct feature vectors for drugs and targets, then the feature vectors of drug-target pairs are usually concatenated to be training samples, finally, a classifier is trained to predict novel DTIs based on those training samples. In general, those methods can be categorised into two groups. The first group of methods construct feature vectors for drugs and targets based on the DSSN and the TSSN. The second group of methods construct feature vectors without constructing similarity networks.

The first group of feature vector based methods learn feature vectors from the similarity networks. Drugs/targets with similar structures are assumed to form clusters in the similarity networks, and feature vectors are constructed based on the proximity between drugs/targets in the similarity networks. DTINet (Luo et al., 2017) learns drug embeddings and target embeddings by simulating random walk with restart algorithm on the DSSN

and TSSN, then learns a projection matrix via minimising the distance between the DTI prediction matrix and the adjacency matrix of the DTI network. NeoDTI (Wan et al., 2019) is a deep learning based end-to-end method. In NeoDTI, the feature vectors are updated along with the weights in the projection matrix during the learning process. AOPEDF (Zeng et al., 2020) learns the drug embeddings and target embeddings with the freedom of choosing arbitrary connectivity of the DTI network, then applies deep forest (Zhou and Feng, 2017) as a classifier to predict DTIs. Dtigems+ (Thafar et al., 2020) learns feature vectors from two sources. First, the similarity in drug structure and target sequence Second, implicit relations in the DTI network. Then, a SVM is employed as a classifier based on the feature vectors of those sources.

The second group of feature vector based methods learn feature vectors without constructing similarity networks. It is worth mentioning that many methods in this group employed deep neural networks to learn the feature vectors or be the classifier, as deep neural networks are powerful in extracting features (LeCun, Bengio, and Hinton, 2015). Wang et al., 2017a proposed a method that employs stacked autoencoders to extract features from position-specific scoring matrix (Gribskov, McLachlan, and Eisenberg, 1987) of targets. The feature vectors of drugs are fingerprint representations. Finally, the rotation forest algorithm (Rodriguez, Kuncheva, and Alonso, 2006) is used as the final classifier. In CnnDTI (Hu et al., 2019), the feature vectors of drugs are constructed using PaDEL-descriptor (Yap, 2011) and the feature vectors of targets are constructed using Moran autocorrelation descriptors algorithm (Li et al., 2006), a convolutional neural network is trained to make DTI predictions. In FRnet-DTI (Rayhan et al., 2020), molecular fingerprints of drugs are used as feature vectors and evolutionary features (Mousavian et al., 2016) of proteins are the feature vectors of targets. Then, an autoencorder is employed to encode features from the feature vectors. Finally, a convolutional neural network is used as the classifier. In DNN-DITs (Chen

et al., 2021), a few feature vectors for drugs and targets are constructed using different methods. Then XGBoost (Chen and Guestrin, 2016) is employed to select powerful features from those feature vectors. Finally, a deep neural network is employed as the classifier.

2.3.4 Drug-target interaction prediction evaluation criteria

To evaluate the performance of DTI prediction, the previous methods divide known DTIs into two groups that are the training DTIs and testing DTIs. From the training DTIs, a DTI prediction model is trained. Based on the DTI prediction model, the prediction matrix M is learned. From M, the prediction scores between the testing DTI pairs and randomly sampled non-existence DTI from the original DTI network are compared. The entries in the prediction matrix M are real values that are between 0 and 1, as a result, it is inappropriate to use F1 score to measure the performance of prediction. Following previous works (Luo et al., 2017; Wan et al., 2019), the metric of evaluating the performance of predicting DTI is the area under precision and recall curve (AUPR). AUPR is assumed to be more suitable for imbalanced datasets, while other popular metrics such as the area under receiver operating characteristic curve (AUC) would give an optimistic evaluation of the prediction (Davis and Goadrich, 2006).

Since the AUPR from a single split of DTIs might not be reliable, recent studies suggest that k-fold cross validation should be conducted. In the k-fold cross validations, the known DTIs are splitted into k groups. In each experiment, one group is used as the testing DTIs, and all the other k-1 groups are used as training DTIs. In practice, the average AUPR of multiple independent k-fold cross validations is used to evaluate the prediction performance of the DTI prediction methods.

Another way of evaluating the DTI prediction is to conduct a literature

review of the newly identified DTIs. Some of the DTIs that are not in the datasets could be experimentally validated after the datasets are published.

2.4 Chapter summary

For metabolite annotation on LC-MS untargeted data, Problem 1 is defined in Section 1.4.1. In Section 2.1, the previous methods in annotating peaks of LC-MS untargeted data are reviewed. Then, the method for the performance evaluation of the annotation methods is introduced. For network biomarker detection, the Problem 2 has been defined in Section 1.4.2. In Section 2.2, the previous methods in detecting the network biomarkers from the gene expression data are reviewed. The gene enrichment analysis on gene topological modules is also introduced. For DTI prediction, Problem 3 has been defined in Section 1.4.3. In Section 2.3, the previous DTI prediction methods and the corresponding evaluating methods are reviewed.

Chapter 3

Metabolite annotation via module detection

In Section 1.4.1, the problem of metabolite annotation for LC-MS untargeted data is introduced. In Section 2.1, a number of metabolite annotation methods are reviewed. This chapter introduces our approach to the first research question. Section 3.1 introduces the intuition behind our approach. Section 3.2 introduces the details of the proposed approach. Section 3.3 validates our approach on several benchmark datasets against the popular metabolite annotation methods. Finally, Section 3.4 summarises the chapter.

3.1 Introduction

As discussed in Section 1.4.1, the annotation of metabolites is an important task for analysing LC-MS untargeted data. However, the annotation of metabolites still remains a bottleneck in metabolism (Dunn et al., 2013).

The main reason for the difficulty in the peak annotation is that more than one theoretical ionised product could be assigned to a peak. The reason is that the many theoretical ionised products could have overlapping theoretical m/z. Hence the main challenge of peak annotation is to distinguish the true metabolite source of the peaks from the false metabolite sources.

From the literature review on Section 2.1, the existing methods rely on the inter-peak relations to annotate peaks. Those methods first group peaks into non-overlapping peak modules. Then, the ionised products are annotated to peaks in each peak module based on the adduct pattern and isotopic pattern within. Finally, the MF of a metabolite is obtained through matching the neutral mass obtained from the previous step to a candidate neutral mass list.

However, the grouping peaks are uncertain due to the nature of LC-MS untargeted data as discussed in Section 1.4.1. The peaks from the same metabolite source might be divided into different peak modules. Thus, some metabolites may not be annotated to the corresponding peaks, even if those peaks are measured by LC-MS.

To handle the uncertainty of peak grouping, we propose a novel method named Metabolite Annotation via Module Detection (MAMD). MAMD adopts a module detection strategy for LC-MS metabolites annotation. Starting for each peak, MAMD searches various combinations of peaks as peak modules and each peak module explores the existence of a possible candidate metabolite. The peak module detection is conducted without confining the peaks in the predefined clusters. Moreover, peak modules are detected independently, and the overlapping of peaks between modules is allowed. Each peak module is assigned with an annotation score, as a result, the user could rank the annotations of a peak or remove all low confidence annotations. Since a peak could belong to more than one peak modules and each peak module is assigned with an annotation score, each peak is annotated with a list of possible ionised products with the corresponding annotation scores, MAMD can handle the uncertainty of ionised product assignment in addition to the uncertainty of peak grouping. Other novelties of MAMD are the module detection process is conducted on an attributed PSN and the

metabolite reaction information is also employed in MAMD to further determine the annotations scores of peak annotations.

3.2 Method

The proposed MAMD consists of four steps. First, a candidate ionised product list is generated for annotation searching. Then, an attributed PSN is constructed from the features of peaks to represent the peaks and their pairwise relationships. Third, peak modules are detected from the network using a novel strategy for putative annotation. Finally, the metabolite reaction information is used as the prior knowledge to adjust annotation scores for the pairs of annotated metabolites that co-exist in a pathway. The entire framework is shown in Figure 3.1.

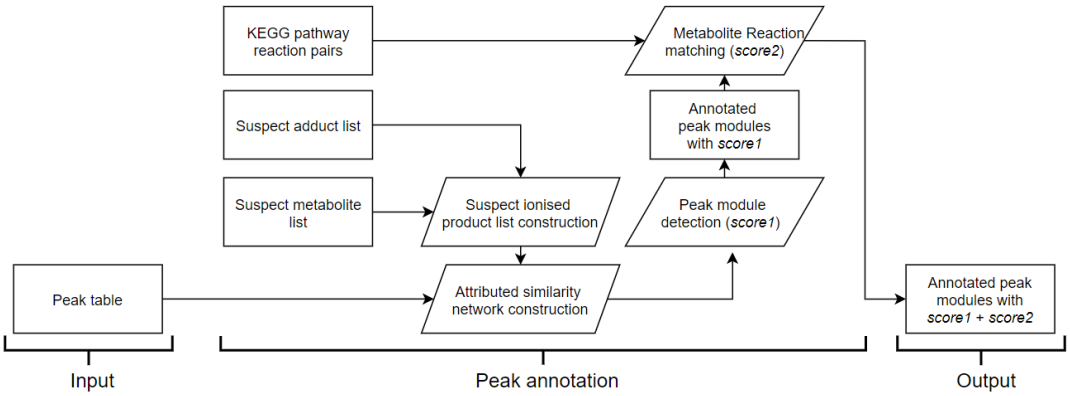


FIGURE 3.1: The framework of MAMD. MAMD takes a processed LC-MS peak table as the input. There are built-in candidate adduct list, metabolite list, and KEGG pathway reaction pairs MAMD representing the assumptions about the samples, and those assumptions can be customised for each specific datasets. The candidate metabolite list coupled with candidate adduct list is used to construct the candidate ionised product list which is described in Section 3.2.1. The peak table and the candidate ionised product list are used to construct an attributed PSN which is described in Section 3.2.2. The attributed PSN is used as the input for peak module detection which is described in Section 3.2.3. The annotated peak modules are used as the input for the metabolite pathway matching which is described in Section 3.2.4. Finally, the peak modules with both *score*1 and *score*2 are the output of MAMD.

3.2.1 Candidate ionised product list construction

A candidate ionised product list is constructed by combining a MF list M obtained from the candidate metabolite list and a candidate adduct list Φ. Both M and Φ can be user-defined. For each $m \in M$, the isotopic list $I(m)$ is constructed using BRAIN (Dittwald et al., 2013), which takes an MF as input and calculates the corresponding theoretical masses and occurrence probability of the isotopes. For each pair of $(m, \phi) \in M \times \Phi$, we can construct a candidate ionised product list $L(m, \phi)$ that consists of I_m isotopic indices, where $I(m)$ indicates the number of isotopes in $I(m)$. For each $l \in L(m, \phi)$, we use $y_{m\phi l}$, $z_{m\phi l}$ to denote theoretical m/z, the most abundant isotopic peak indicator which is 1 for true, 0 otherwise. The theoretical occurrence probability of an isotope derived from BRAIN is simplified to the most abundant indicator $z_{m\phi l}$ as there might be an error in the peak area detection, and the intensities of peaks might be log normalised.

For example, Guanidine (CH5N3) has two isotopes with occurrence probability larger than 0.01 based on BRAIN. Their exact masses are 59.048347 g/mol and 60.048678 g/mol, and the occurrence probabilities are 0.977973 and 0.021858, respectively. For adduct $[\text{MF+H}]^+$ that adds 1.007276 g/mol of the measured mass and it is charged once, we have $y_{m\phi 1} = 60.055623$, $y_{m\phi 2} = 61.055954$, $z_{m\phi 1} = 1$, $z_{m\phi 2} = 0$ for m =CH5N3 and $\phi = [\text{MF}+H]^+$.

An annotation reward term $t_{m\phi l}$ for each $l \in L(m, \phi)$ is defined to indicate the confidence of observing a candidate ionised product as:

$$t_{m\phi l} = \begin{cases} S_n + S_p[\phi \in \Phi_s] & \text{if } y_{m\phi l} = 1 \\ S_r & \text{otherwise} \end{cases} \tag{3.1}$$

where S_n is the minimum annotation reward for the most abundant ionised products, Φ_s denotes a set of primary adducts, S_p is the additional

reward for primary adducts, [ ] is the Iverson bracket that $[p] = 1$ if condition p is satisfied and $[p] = 0$ otherwise, S_r is the reward for less abundant isotopic ionised products.

The annotation reward varies between candidate ionised products for two reasons. First, some adducts are categorised as primary adducts as they are more likely to be observed than the others (Brown et al., 2011, Kuhl et al., 2011, Uppal, Walker, and Jones, 2017). Denoted as a subset $\Phi_s \subset \Phi$, the primary adducts are given an additional annotation reward because their confidences of existence are higher. Second, less abundant isotopic peaks are not given additional rewards even if they are generated by primary adducts. The reason is that those ionised products are regarded as validations of adducts.

3.2.2 Attributed peak similarity network construction

An attributed PSN $G = (V, E)$ is used to represent the pairwise relationship between peaks, in particular, each node in V denotes a peak, and each edge in E represents the two corresponding peaks that could be from the same metabolite source. Attributes R, P, Z, T are attached to the network to represent the peak features, R represents the RT, P represents the median value of PIAS, Z stores the most abundant indicators of the matched ionised products, and T stores the rewards of the matched ionised products.

To eliminate low confidence edges, given any pair of $(v_i, v_j) \in V \times V$, $i \neq j$, the edge a_{ij} is calculated as:

$$a_{ij} = \begin{cases} 1 & \text{if } corr(X_i, X_j) > \alpha \text{ and } |R_i - R_j| < \beta \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where X_i and X_j are the PIAS of node v_i, v_j respectively, $Corr(X_i, X_j)$ represents the Pearson's correlation coefficient between X_i and X_j, and R_i and R_j are the RT of the corresponding peaks, respectively. $\alpha \in (-1, 1)$ is the

user defined Pearson correlation coefficient cutoff and $\beta \in (0, +\infty)$ is the RT cutoff. For a pair of nodes (v_i, v_j), $a_{ij} = 1$ indicates an edge exists, while $a_{ij} = 0$ means the edge does not exist. Thus the low confidence edges are removed. It is worth noting that G may not be a connected network after the edge filtering.

For any node v_i, we define θ_i as the measured m/z, and attributes Z_i and T_i are calculated as:

$$Z_i = \{z_{m\phi l} \,|\, |y_{m\phi l} - \theta_i| < \Gamma\theta_i,\, m \in M, \phi \in \Phi, l \in L(m, \phi)\} \tag{3.3}$$

$$T_i = \{t_{m\phi l} \,|\, |y_{m\phi l} - \theta_i| < \Gamma\theta_i,\, m \in M, \phi \in \Phi, l \in L(m, \phi)\} \tag{3.4}$$

where Γ is the parts per million (ppm) tolerance. *Eq*. (3.3) and *Eq*. (3.4) bridges the measured peaks with the candidate ionised product list by defining the search space of possible annotation for each peak.

3.2.3 Peak module detection

The peak module detection is conducted on the attributed PSN and there are two components in it. First, an annotation score is introduced to measure the confidence of an annotated peak module. Second, a novel greedy search strategy guided by the annotation score is developed to detect and annotate peak modules.

Annotation score

In MAMD we propose an annotation score as the objective function for peak module detection. Here, an annotation score for a single annotated peak is designed. Then, it is used as a building block of the final annotation score that measures the confidence of an annotated peak module.

Given the simplest case, the m/z of a single peak $v_i \in V$ can be matched to the ionised products that are stored in T_i. For v_i, the annotation score regarding to m and ϕ is defined as:

$$f(m, \phi, T_i) = \sum_{t_{m_a \phi_b l_c} \in T_i} t_{m_a \phi_b l_c} [m_a = m, \phi_b = \phi] \tag{3.5}$$

where $t_{m_a \phi_b l_c}$ represents the annotation reward for the combination of m_a, ϕ_b, l_c. At most one candidate ionised product can be matched in *Eq.* (3.5), as the theoretical m/z of isotopic ionised products from the same adduct are larger than the ppm error.

Peak module can be regarded as a sub-network $G' = (V', E') \subset G = (V, E)$. If all the nodes V' in G' are from the same metabolite m, G' should have the following properties. First, the RT difference should be smaller than β for any node pairs in G', as molecules from the same metabolite source should be ejected from the LC with similar RT. Second, G' should be a connected sub-network, i.e., there exists a path for any node to other node in G', as the peaks in G' have a similar pattern in PIAS. Third, G' can be further divided into a series of sub-sub-networks G'_1,...,G'_N, wherein the nodes in $G'_n, n \in [1, N]$ are not only from the same metabolite but also from the same adduct. The annotated adducts of G'_n are denoted as $\Phi' = (\phi'_1, ..., \phi'_N)$.

The sub-sub-networks should satisfy the following constraints: First, the PIAS patterns of peaks within any sub-sub-network $G'_n, n \in [1, N]$ should be highly similar as they are isotopes. Hence, each sub-sub-network should be a clique. Second, the adduct without the most abundant isotopic peak being observed is considered as unreliable. Therefore for G'_n, there should exist a node v_i that has the following properties: $z_{m\phi l} = 1 \in Z_i$, $v_i \in V'_n$ and $z_{m\phi l} = 0 \in Z_j, \forall v_j \in V'_n \setminus v_i$. Particularly, if there is only one node in G'_n, this node should be the most abundant isotopic peak. Finally, to further validate the isotopic pattern, the peak assumed to be the most abundant isotopic peak

should have a higher median intensity than any other peaks in G'_n.

Combining the above criteria, the annotation score of a sub-network G' is defined as:

$$score1(G' = (G'_1, ..., G'_N), \Phi' = (\phi'_1, ..., \phi'_N), m) = \begin{cases} \sum_{n=1}^{N} \sum_{v_i \in G'_n} f(m, \phi'_n, T_i) & \text{if the following restriction are met} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

subject to restrictions:

$1: \ \forall n \in [1, ..., N], v_i \in G'_n, f(m, \phi'_n, T_i) \neq 0,$

$2: \ G'$ is connected,

$3: \ \forall (v_i, v_j) \in V' \times V', i \neq j, |R_i - R_j| < \beta,$

$4: \ \forall (v_i, v_j) \in V'_n \times V'_n, n \in (1, .., N), i \neq j \ , \quad a_{ij} = 1,$

$5: \ \forall V'_n, n \in (1, .., N),$

$$(\exists! z_{m\phi'_n l} = 1 \in Z_i) \wedge (P_i > P_j), v_i \in V'_n, v_j \in V'_n \setminus v_i.$$

The annotation score is the sum of all the annotation rewards of all nodes in a sub-network G' if the restrictions are fulfilled. If any of the restrictions listed above is violated, the annotation score is set to 0.

Peak module detection

A greedy search strategy is developed to detect peak modules effectively using *Eq*. (3.6) as the objective function. In particular, the search strategy selects a single peak as the initial annotated module, which is then expanded by adding its neighbours using a greedy search strategy.

For a PSN $G = (V, E)$ with attributes P, R, Z, T, the first step is to find all the initial annotated sub-networks. Since the fifth restriction in *Eq*. (3.6) ensures that most abundant isotope must be included to annotate an adduct,

we search $\forall v_i \in V$ to find every $z_{m\phi l} = 1 \in Z_i$ and save the triplets (v, m, ϕ) as a collection of start points that denoted as SP.

A start point $(v_i, m_1, \phi_1) \in SP$ can be seen as a sub-network G' in G with only one node and includes only one sub-sub-network $G' = (G'_1)$ with the adduct list $\Phi' = (\phi_1)$. Since $z_{m_1\phi_1 l} = 1 \in Z_i$, we have $f(m_1, \phi_1, T_i) > 0$. As a result, the first restriction in *Eq*. (3.6) is met. The remaining restrictions can be ignored, as they are the validations of peaks from the same metabolite source, yet, there is only one node in G'. Thus, we have $score1(G', m_1) > 0$. It is worth mentioning that (v_i, m_1, ϕ_1) would be removed from SP, as this combination has been explored.

The nodes in $\mathcal{N}(G')$ can be categorised into three groups based on the impact of adding them to G'. First, the isotopes of an annotated adduct. For a node v_j with $z_{m_1\phi_e l} = 0 \in Z_j$ and $\phi_e \in \Phi'$, it could be allocated to the sub-sub-network G'_n with the same annotated adduct ϕ_e as long as adding v_j does not violate any constraints in *Eq*. (3.6). Second, the most abundant isotope from an adduct not in Φ'. For a node v_j with $z_{m_1\phi_d l} = 1 \in Z_j$ and $\phi_d \notin \Phi'$, a new sub-network of G' could be created for v_j, and the ϕ_d could be added to the annotated adduct list Φ' as long as adding v_j does not violate any constraints in *Eq*. (3.6). Note that if $(v_j, \phi_d, m_1) \in SP$, the combination (v_j, ϕ_d, m_1) will be removed from SP to reduce the computational cost, if v_j has been added to G'. The reason is that this combination has been explored. Third, the nodes will violate any constraints in *Eq*. (3.6), if they are added to G'. Hence, $score1$ would be 0 if those nodes are added to G'.

In MAMD, the nodes in $\mathcal{N}(G')$ are added to G' using a greedy search strategy. From $\mathcal{N}(G')$, the node that will lead to the highest increase in $score1$ will be added to G'. Since there is G' has expanded, $\mathcal{N}(G')$ would change as well. From the new $\mathcal{N}(G')$, the node that will lead to the highest increase in $score1$ will be added to G'. The process of adding nodes will repeat, until

*score*1 can not be increased by adding any node in $\mathcal{N}(G')$ to G'. The final sub-network G', the annotated adduct list Φ', metabolite m, and *score*1 are stored as $(G', \Phi', m, score1(G', \Phi', m))$ in an annotated list (AL). A new searching process is conducted from the next starting point in SP, until all start points are removed from SP.

3.2.4 Metabolite reaction matching

MAMD utilises metabolite pathway information to increase the annotation scores for metabolite pairs that are more likely to co-exist. Inspired by MI-Pack (Weber and Viant, 2010), we use metabolite reaction pairs from KEGG (Kanehisa and Goto, 2000) as prior knowledge. The co-existence metabolite pairs in KEGG pathways are denoted as Ψ. For each annotated peak module, this module coupled with the other modules in the AL is used to construct annotated peak module pairs. The number of matching pairs between Ψ and annotated peak module pairs is used to calculate a pathway score. The difference between our approach and MI-Pack is that peak module pairs are matched to Ψ in our approach, while MI-Pack directly matches peak pairs to Ψ. For a peak module $(G^i, m_k) \in AL$ and the pathway score is defined as:

$$score2(m_k, \omega) = \min(\sum_{(m_l)\in\omega} \varphi[(m_k, m_l) \in \Psi], U) \tag{3.7}$$

where φ is the reward for each match of the reaction pairs, $U \in (0, +\infty)$ is the upper bound of the pathway reward, ω is a set of unique MF from AL, and min(a,b) denotes the smaller number between $a \in \mathbb{R}$ and $b \in \mathbb{R}$. The upper limit of U is introduced for two reasons. First, the mapped metabolites obtained from the MFs of annotated peak modules are uncertain, this is due to the existence of isomers. Second, the annotated peak modules from the previous step are uncertain especially for modules with small *score*1.

3.2.5 The proposed algorithm

The pseudocode of module detection and peak annotation in MAMD is summarised Algorithm 3. In Algorithm 3, lines 1-24 describe the peak module detection, and lines 25-30 describe the metabolite pathway matching.

Algorithm 3 The module detection and peak annotation in MAMD

Input: Network G with attributes R, P, Z, T.
Hyper-parameters: α; β.
Output: annotated peak modules Ω.

1: $SP = [\,]$.
2: $AL = [\,]$.
3: **for** each $v_i \in G$ **do**
4: **for** each $z_{m\phi l}$ in Z_i **do**
5: **if** $z_{m\phi l} = 1$ **then**
6: add (v_i, m, ϕ) into SP.
7: **end if**
8: **end for**
9: **end for**
10: **for** each $(v_i, m, \phi) \in SP$ **do**
11: create new $G' = [\,]$.
12: add v_i to G' and $G' = (G'_1)$, $\Phi' = (\phi)$.
13: remove (v_i, m, ϕ) from SP.
14: **if** a node $v_j \in \mathcal{N}(G')$ that can increase the annotation score $Eq.$ (3.6) via adding it to G' exist **then**
15: add the node $v_j \in \mathcal{N}(G')$ which could provide the most increase to G' and expand Φ' if necessary.
16: **if** Φ' is expanded (ϕ_n is used to denoted the added adduct) **then**
17: **if** $(v_j, \phi_n, m) \in SP$ **then**
18: remove (v_j, ϕ_n, m) from SP.
19: **end if**
20: **end if**
21: **end if**
22: repeat line 14-21 until no more nodes could be added to G'.
23: save $(G', \Phi', m, score1(G', \Phi', m))$ to AL.
24: **end for**
25: get a unique MF set ω from AL
26: **for** each $(G', \Phi', m, score1(G', \Phi', m)) \in AL$ **do**
27: calculate $score2(m,\omega)$ based on $Eq.$ (3.7).
28: save $(G', \Phi', m, score1(G', \Phi', m), score2(m, \omega))$ to Ω.
29: **end for**
30: **return** Ω.

3.2.6 Complexity analysis

Here, the complexity of peak module detection stage of Algorithm 3 is analysed. This stage includes line 3-24. The first part is line 3-9, the operation is to identify nodes with features $z_{m\phi l} = 1$. It will need to traverse around all the nodes and the all feature vectors Z. For the nodes, assuming the average number of candidate ionised products in the Z is z^a, the complexity can be written as $O(z^a|V|)$. Assuming the average time of a peak being in SP is z^b, there are $z^b|V|$ peak annotation combination in SP. In line 10-24, for a peak annotation combination in SP, a module detection is conducted, the module detection process requires to traverse all the neighbours of a module and the corresponding feature Z. Assuming the average number of neighbours of a peak module is n^a at each iteration, and the average number of peaks in the peak modules is k, the complexity of each module detection is $O(kn^a z^a)$. Since there are $z^b|V|$ combinations in SP, the complexity is $O(kn^a z^a z^b|V| + z^a|V|)$ without considering the peak annotation combinations being removed from SP. Assuming the proportion of peak annotation combinations in SP being used as start point of a peak module is c, the final complexity is $O(ckn^a z^a z^b|V| + z^a|V|)$. In practice, c, k, n^a, z^a, z^b can be seen as small constant values. Under the condition of 1000 decoy metabolites on Std1.POS, it has been recorded that $c = 0.91$, $k = 1.16$, $n^a = 24.93$, $z^a = 1.54$, $z^b = 0.29$. Under the condition of 6000 on Std1.POS, it has been recorded that $c = 0.95$, $k = 1.09$, $n^a = 25.34$, $z^a = 4.15$, $z^b = 1.20$.

The metabolite reaction matching stage of Algorithm 3 is described at line 25-29. The scores for all peak module pairs need to be computed. Since number of modules is $cz^b|V|$, assuming the corresponding metabolites in all the module are distinct and the number of metabolite pathway reaction pairs is p, the complexity can be written as $O(p(cz^b|V|)^2)$. Fortunately, there are usually a few thousands of peaks in a dataset, the running time is still bearable.

3.3 Experiments

3.3.1 Comparison with state-of-the-art algorithms

We conducted the experiments on standard mixture datasets that are introduced in Section 2.1. An annotation algorithm should be able to identify the metabolites in the sample and reduce the false detection. In order to evaluate the performance of algorithms, known compounds which are metabolites known to be in the sample and decoy metabolites which are not in the sample are added to the candidate metabolite list in the experiments.

For real world biological samples, the number of candidate metabolites can be huge. Therefore, we conduct experiments under 600, 1000, 2000, 4000 and 6000 decoy metabolites to test the performance of the metabolite annotation algorithms in different realistic scenarios. In contrast to Daly et al., 2014, the numbers of decoy metabolites are just 100, 600, and 1000. The decoy metabolites are the metabolites that are not in the samples, and they are randomly sampled from the KEGG metabolite library. Note that no isomers are included as none of existing algorithms can distinguish them including ours. The evaluation metrics have been introduced in Section 2.1.

TABLE 3.1: Candidate adducts in MAMD.

adducts in positive mode			
$[MF+H]^+$	$[MF+2H]^{2+}$	$[MF+3H]^{3+}$	$[MF+Na]^+$
$[MF+2Na]^{2+}$	$[MF+3Na]^{3+}$	$[MF+K]^+$	$[MF+NH4]^+$
$[MF+H+2Na]^{3+}$	$[MF+2H+Na]^{3+}$	$[MF+ACN+H]^+$	$[MF+ACN+2H]^{2+}$
$[MF+ACN+Na]^+$	$[MF+2ACN+H]^+$	$[MF+2ACN+2H]^{2+}$	$[MF+3ACN+2H]^{2+}$
$[MF+H+Na]^{2+}$	$[MF+H+NH4]^{2+}$	$[MF+H+K]^{2+}$	$[MF+2K-H]^+$
[MF+CH3OH+H	$[MF+IsoProp+H]^+$	$[MF+DMSO+H]^+$	$[MF+IsoProp+Na+H]^+$
$[MF+2Na-H]^+$	$[2MF+H]^+$	$[2MF+K]^+$	$[2MF+Na]^+$
$[2MF+NH4]^+$	$[2MF+ACN+H]^+$	$[2MF+ACN+Na]^+$	
adducts in negative mode			
$[MF-H]^-$	$[MF-2H]^{2-}$	$[MF-3H]^{3-}$	$[MF+FA-H]^-$
$[MF+Cl]^-$	$[MF+Br]^-$	$[MF-H2O-H]^-$	$[MF+TFA-H]^-$
$[MF+K-2H]^-$	$[MF+Hac-H]^-$	$[MF+Na-2H]^-$	$[2MF-H]^-$
$[2MF+Hac-H]^-$	$[3MF-H]^-$		

For MAMD, the candidate adducts in both positive mode and negative mode are listed in Table 3.1. Similar to the settings in existing algorithms, the primary adducts in positive mode are $[MF+H]^+$, $[NF+K]^+$, $[MF+Na]^+$, $[MF+NH4]^+$ and in negative mode are $[MF-H]^-$, $[MF+Cl]^-$, $[MF+Na-2H]^-$, $[MF+K-2H]^-$. A peak module will be ignored if the corresponding annotation score is below 4. In the experiment settings of MAMD, $\alpha = 0.7$, $\beta = 5$ seconds and $\Gamma = 3$ ppm unless otherwise specified, and the parameter sensitivity analysis is conducted in Section 3.3.4.

Five state-of-the-art algorithms are selected for comparison including CAMERA, mzMatch, MetAssign, xMSannotator, and CliqueMS. For these algorithms, the parameters from their own publications are used. For CAMERA and CliqueMS, neutral masses are annotated to peaks rather than metabolites, and neutral masses that do not belong to known or decoy metabolites are ignored. For MetAssign and CliqueMS, peaks are annotated with a ranked list of potential metabolites. The reported F1 scores are obtained from the top k annotations of each peak after fine tuning. For xMsannotator, a confidence score is assigned to each peak module and the F1 scores reported here are obtained from tuning the minimum confidence c. The F1 scores of all algorithms are summarised in Figure 3.2.

From Figure 3.2 we can find that the MAMD shows very promising performance compared with the state-of-the-art algorithms in all six datasets. MAMD receives the highest F1 scores when the largest number of decoy metabolites are added into the candidate metabolite list. While the number of decoy metabolites is relatively small, MAMD can still obtain the best or the second best F1 scores. The results also show that the MAMD is more robust on a larger number of decoy metabolites. While the F1 scores of the comparing algorithms decrease significantly, MAMD decreases gradually. The reason is that MAMD searches the peak modules for each candidate metabolite independently and overlapping between peak modules is allowed, as a

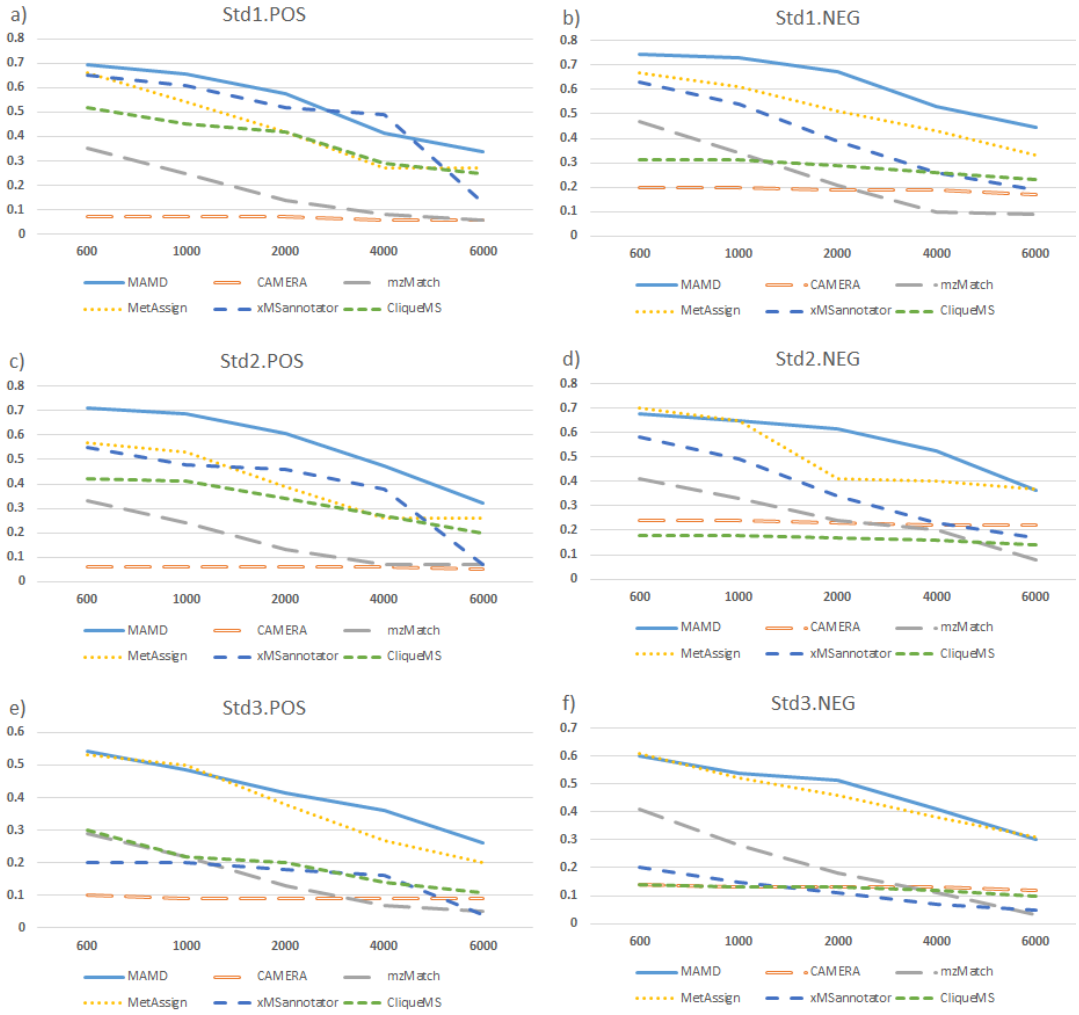


FIGURE 3.2: The F1 score of MAMD and state-of-the-art algorithms under various numbers of decoy metabolites. In the six sub-figures, the horizontal-axis indicates the number of decoy metabolites and the vertical-axis indicates the F1 score of each algorithm.

result, the annotation process will not be affected by the increasing of decoy metabolites as all the possible adduct and isotopic patterns will annotated to peaks. An example of the overlapping peak modules is shown in Table 3.2. From Table 3.2, it can be seen that there are overlapping peaks in those two modules, and they can be annotated with different patterns. In general, the metabolites which do exist in the samples will leave more evidence of the existence, however, it may not be true in the same cases as shown in the aforementioned example. The reason behind this is due to the uncertainty of peak annotation.

It is worth mentioning that the F1 scores from MetAssign, mzMatch and CAMERA are very similar to the reported results in Daly et al., 2014 for Std1 and Std2 but increases for Std3. The reason may be that the decoy metabolites are chosen in different methods. Finally, the F1 scores of xMSannotator for Std1.POS under 600 and 1000 decoy metabolites are slightly lower here than reported in Uppal, Walker, and Jones, 2017. The reason could be that only metabolites in HMDB (Wishart et al., 2009) are used to calculate F1 score in Uppal, Walker, and Jones, 2017 in addition to the differences of the decoy metabolites.

3.3.2 Annotation score sensitivity

In the existing algorithms, there are mainly two methods to represent the confidences of peak annotations, which are the confidence ranks and annotation scores. The second method is employed in MAMD, as all annotations for a peak can be ignored if their corresponding scores are too low. In contrast, the first method would still put the most likely annotation at the top even if the confidences of all the annotations are low.

Since we use the annotation score function to quantify the confidences of the annotations, there should be a trade-off between precision and recall with

TABLE 3.2: An example of two overlapping peak modules.

Peak module annotated with C6H12O6 that is known in Std1.POS.					
Peak ID	Measured m/z	RT	median PIAS	Annotation	Most Abundant
210	244.08	703.58	13342390	$[MF+ACN+Na]^+$	TRUE
221	222.10	705.03	12530315	$[MF+ACN+H]^+$	TRUE
1385	245.08	703.58	893586	$[MF+ACN+Na]^+$	FALSE
1407	223.10	704.70	842411	$[MF+ACN+H]^+$	FALSE
3152	203.05	704.32	43265	$[MF+Na]^+$	TRUE
Peak module annotated with C8H15NO6 that is known to be not in Std1.POS.					
Peak ID	Measured m/z	RT	median PIAS	Annotation	Most Abundant
210	244.08	703.58	13342390	$[MF+Na]^+$	TRUE
221	222.10	705.03	12530315	$[MF+H]^+$	TRUE
912	465.17	703.58	1114844	$[2MF+Na]+$	TRUE
1385	245.08	703.58	893586	$[MF+Na]^+$	FALSE
1407	223.10	704.70	842411	$[MF+H]+$	FALSE
1681	285.11	704.70	322726	$[MF+ACN+Na]^+$	TRUE
3658	260.05	701.10	62575	$[MF+K]^+$	TRUE

the annotation score. Here we study the impact of the annotation score on precision and recall. The minimum annotation score cutoff is set to 3, 4 and 5, and a candidate metabolite list with 1000 decoy metabolites is used for evaluation. The performance of MAMD is shown in Table 3.3, from which there are three observations. First, a lower annotation score cutoff can yield a higher recall rate. Second, a high annotation score cutoff can yield a high precision rate. Third, by tuning the annotation score cutoff, it is possible to switch the trade-off between precision and recall.

3.3.3 The effect of metabolite reaction matching

To evaluate the performance gain of incorporating metabolite reaction matching, the performance of annotation only using *score*1 is also shown in Table 3.3. Comparing *score*1 against *score*1 + *score*2, there is a reasonable increase of performance. It can be seen that both recall and F1 scores increase with the incorporating of *score*2, and the difference in precision is marginal in most datasets. To find whether the difference in precision, recall and F1 score

TABLE 3.3: The impact of the annotation score under 1000 decoy metabolites.

		$score \geq 3$	$score \geq 4$	$score \geq 5$
			$score1$	
Std1.POS	Precision	0.3585	0.7206	0.6800
	Recall	0.7308	0.4712	0.3269
	F1 score	0.4810	0.5698	0.4416
Std1.NEG	Precision	0.4239	0.9167	0.9429
	Recall	0.7500	0.4231	0.3173
	F1 score	0.5417	0.5789	0.4748
Std2.POS	Precision	0.3966	0.7547	0.7568
	Recall	0.7320	0.4124	0.2887
	F1 score	0.5145	0.5333	0.4179
Std2.NEG	Precision	0.4422	0.8605	0.8788
	Recall	0.6701	0.3814	0.2990
	F1 score	0.5328	0.5286	0.4462
Std3.POS	Precision	0.3333	0.6000	0.6471
	Recall	0.5000	0.3000	0.2750
	F1 score	0.4000	0.4000	0.3860
Std3.NEG	Precision	0.3393	0.6667	0.6000
	Recall	0.4750	0.3000	0.2250
	F1 score	0.3958	0.4138	0.3273
			$score1 + score2$ (MAMD)	
Std1.POS	Precision	0.3680	0.6804	0.7375
	Recall	0.8173	0.6346	0.5673
	F1 score	0.5075	0.6567	0.6413
Std1.NEG	Precision	0.4293	0.7802	0.8841
	Recall	0.7885	0.6827	0.5865
	F1 score	0.5559	0.7282	0.7052
Std2.POS	Precision	0.4108	0.7867	0.8182
	Recall	0.7835	0.6082	0.4639
	F1 score	0.5390	0.6860	0.5921
Std2.NEG	Precision	0.4533	0.8500	0.8750
	Recall	0.7010	0.5258	0.4330
	F1 score	0.5506	0.6497	0.5793
Std3.POS	Precision	0.3443	0.6154	0.7000
	Recall	0.5250	0.4000	0.3500
	F1 score	0.4158	0.4848	0.4667
Std3.NEG	Precision	0.3333	0.6667	0.6818
	Recall	0.4750	0.4500	0.3750
	F1 score	0.3918	0.5373	0.4839

is caused by randomness, Wilcoxon signed-rank test (Wilcoxon, 1992) is applied to two vectors that is the size of 18 (3 score settings on 6 datasets) from $score1$ and $score1 + score2$ respectively. In this test, the null hypothesis is that the mean ranks between the vectors are the same. The p-value of precision is 0.62, the p-value of recall is 0.00, and the p-value of F1 score is 0.00. Using 0.05 as the cutoff of p-value. We can not reject the null hypothesis in precision, while the null hypothesis in recall and F1 score can be rejected. Hence, by incorporating $score2$, the recall would increase and precision remains at a similar level. The reason is that the metabolite reaction matching mainly adjusts the annotation scores of peak module pairs with the corresponding molecules that are known to co-exist. In real-world applications, researchers may focus on specific peaks with interesting patterns such as spikes in peak intensities. The researchers can manually check the annotations and their corresponding annotation scores to design further experiments.

3.3.4 Parameter sensitivity

In the previous experiments, the experiments are conducted with the hyper-parameters set to be $\alpha = 0.7$, $\beta = 5$ seconds, as those settings have been selected in the previous method (Uppal, Walker, and Jones, 2017). To evaluate the impact of α and β, we have conducted additional experiments on Std1.POS under the condition of 1000 decoy metabolites. A grid search is conducted on $\alpha \in \{-1, -0.6, -0.3, 0, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and $\beta \in \{0.1, 1, 3, 5, 7, 9, 15, 30, 50, 100\}$. The result is split into two tables, which are Table 3.4 and Table 3.5, as the results would not fit in a single table.

From Table 3.4 and Table 3.5, there are four observations. First, precision will increase with the increase of α across almost all most β settings with the exception of $\beta = 0.1$ and $\beta = 1$. The reason is that many unreliable edges in the GCN can be removed by increasing α in most cases. Yet, when

$\beta = 0.1$ and $\beta = 1$, the edges only connect peaks with very small RT difference, hence, they are very likely to be from the same metabolite source and correlations of PIAS are very strong between those peaks. Second, recall will decrease with the increase of α especially at $\alpha = 0.8$. This suggests many peak features from the same metabolite source may not have very high correlation in PIAS, although this could be due to the fact that there are only three samples in this dataset. Third, precision will decrease with the increasing of β. Fourth, recall will increase with the increasing of β. Combining the third and fourth observations, it is interesting to find that the margin of increasing recall is smaller than the margin of decreasing precision in comparison. Hence, most of the additional annotations in $\beta = 50$ and $\beta = 100$ could be false annotations, as a result, peaks from the same metabolite source would be in a small RT window.

3.4 Chapter summary

In order to tackle the challenge of peak grouping, we proposed MAMD. A novel annotation score that could quantify the annotation confidence of each peak module is first introduced. Then, from each peak, we use the annotation score as the objective function to detect peak modules via greedy search. Furthermore, the metabolite pathway information is also used to increase the annotation confidences of peaks modules which could co-exist. From the empirical experiments, MAMD not only provides a competitive performance against other state-of-the-art algorithms, but also demonstrates a flexible trade-off between precision and recall, moreover it is more robust against the candidate metabolite lists with a large number of decoy metabolites.

TABLE 3.4: Part 1 of the parameter sensitivity analysis of α and β on Std1.POS under 1000 decoy metabolites.

		α =-1	α =-0.6	α =-0.3	α =0	α =0.3
β =0.1	Precision	0.7083	0.7183	0.7183	0.7143	0.7143
	Recall	0.4904	0.4904	0.4904	0.4808	0.4808
	F1 score	0.5795	0.5829	0.5829	0.5747	0.5747
β =1	Precision	0.7033	0.7033	0.7033	0.7033	0.7033
	Recall	0.6154	0.6154	0.6154	0.6154	0.6154
	F1 score	0.6564	0.6564	0.6564	0.6564	0.6564
β =3	Precision	0.6634	0.6768	0.6768	0.6837	0.6907
	Recall	0.6442	0.6442	0.6442	0.6442	0.6442
	F1 score	0.6537	0.6601	0.6601	0.6634	0.6667
β =5	Precision	0.6442	0.6569	0.6569	0.6700	0.6768
	Recall	0.6442	0.6442	0.6442	0.6442	0.6442
	F1 score	0.6442	0.6505	0.6505	0.6569	0.6601
β =7	Precision	0.6321	0.6442	0.6442	0.6569	0.6634
	Recall	0.6442	0.6442	0.6442	0.6442	0.6442
	F1 score	0.6381	0.6442	0.6442	0.6505	0.6537
β =9	Precision	0.6262	0.6381	0.6381	0.6505	0.6569
	Recall	0.6442	0.6442	0.6442	0.6442	0.6442
	F1 score	0.6351	0.6411	0.6411	0.6473	0.6505
β =15	Precision	0.6239	0.6321	0.6321	0.6442	0.6442
	Recall	0.6538	0.6442	0.6442	0.6442	0.6442
	F1 score	0.6385	0.6381	0.6381	0.6442	0.6442
β =30	Precision	0.5667	0.5877	0.5877	0.5982	0.5982
	Recall	0.6538	0.6442	0.6442	0.6442	0.6442
	F1 score	0.6071	0.6147	0.6147	0.6204	0.6204
β =50	Precision	0.5426	0.5610	0.5610	0.5702	0.5702
	Recall	0.6731	0.6635	0.6635	0.6635	0.6635
	F1 score	0.6009	0.6079	0.6079	0.6133	0.6133
β =100	Precision	0.4931	0.5109	0.5109	0.5185	0.5224
	Recall	0.6827	0.6731	0.6731	0.6731	0.6731
	F1 score	0.5726	0.5809	0.5809	0.5858	0.5882

TABLE 3.5: Part 2 of the parameter sensitivity analysis of α and β on Std1.POS under 1000 decoy metabolites.

		α =0.5	α =0.6	α =0.7	α =0.8	α =0.9
β =0.1	Precision	0.7143	0.7206	0.7206	0.7121	0.7031
	Recall	0.4808	0.4712	0.4712	0.4519	0.4327
	F1 score	0.5747	0.5698	0.5698	0.5529	0.5357
β = 1	Precision	0.7000	0.7079	0.7126	0.7209	0.7176
	Recall	0.6058	0.5962	0.5962	0.5962	0.5865
	F1 score	0.6495	0.6528	0.6492	0.6526	0.6455
β =3	Precision	0.6875	0.6875	0.6989	0.7000	0.7045
	Recall	0.6346	0.6250	0.6250	0.6058	0.5962
	F1 score	0.6600	0.6600	0.6599	0.6495	0.6458
β =5	Precision	0.6804	0.6804	0.6804	0.6809	0.6882
	Recall	0.6346	0.6346	0.6346	0.6154	0.6154
	F1 score	0.6567	0.6567	0.6567	0.6465	0.6497
β =7	Precision	0.6667	0.6667	0.6667	0.6701	0.6771
	Recall	0.6346	0.6346	0.6346	0.6250	0.6250
	F1 score	0.6502	0.6502	0.6502	0.6468	0.6500
β =9	Precision	0.6600	0.6600	0.6600	0.6701	0.6701
	Recall	0.6346	0.6346	0.6346	0.6250	0.6250
	F1 score	0.6471	0.6471	0.6471	0.6468	0.6468
β =15	Precision	0.6471	0.6471	0.6471	0.6566	0.6633
	Recall	0.6346	0.6346	0.6346	0.6250	0.6250
	F1 score	0.6408	0.6408	0.6408	0.6404	0.6436
β =30	Precision	0.6000	0.6000	0.6055	0.6132	0.6250
	Recall	0.6346	0.6346	0.6346	0.6250	0.6250
	F1 score	0.6168	0.6168	0.6197	0.6190	0.6250
β =50	Precision	0.5667	0.5667	0.5726	0.5789	0.6000
	Recall	0.6538	0.6442	0.6442	0.6346	0.6346
	F1 score	0.6071	0.6071	0.6063	0.6055	0.6168
β =100	Precision	0.5267	0.5267	0.5313	0.5403	0.5537
	Recall	0.6635	0.6538	0.6538	0.6442	0.6442
	F1 score	0.5872	0.5872	0.5862	0.5877	0.5956

Chapter 4

Network biomarker detection using Gaussian mixture model clustering

This chapter focuses on answering the second research question, which is introduced Section 1.4.2. Section 4.1 provides the rationale behind our approach. Section 4.2 introduces the details of our method. Section 4.3 validates our approach via comparing the proposed method against the most popular method in a number of datasets, and conducting gene enrichment analysis on the detected network biomarkers. Finally, Section 4.4 summarises this chapter.

4.1 Introduction

As discussed in Section 1.4.2, the detection of network biomarkers is an important task in analysing gene expression data. From the literature review in Section 2.2, the previous methods have strict constraints on the shapes of topological modules. Based on the constraints in detecting topological modules, the previous methods can be divided into two groups. The first group of methods assumes the topological modules should be spherical in shape (Langfelder and Horvath, 2008; Tanvir and Mondal, 2019). Those methods

employ distance based clustering methods on a GCN to detect topological modules. Since nodes are grouped into different modules via minimising the average distance between nodes in the modules, those methods assume that the genes in the same biological process are spherical in shape. The second group of methods assumes genes in the same biological process should form cliques in the GCN (Shi, Derow, and Zhang, 2010; Tanvir et al., 2019). A clique is a topological module in which the nodes are fully connected.

The topological structures of the gene functional modules and gene disease modules may not meet the constraints in the previous methods. The shapes of those modules could be oval or long strips. With the constraints in the previous methods, the genes in the same biological process could be divided into different modules. This would deteriorate the discrimination ability of the network biomarkers, moreover, the molecular mechanism behind the disease could be difficult to identify in the further analysis of the detected network biomarkers.

In order to address the aforementioned issue, we propose a novel method named Network Biomarker Detection using Gaussian Mixture Model Clustering (NBD-GMMC). The main idea behind our method is to apply Gaussian mixture model (GMM) clustering to allow high freedom of the shapes of the topological modules. However, the adjacency matrix of a GCN is high dimensional, and GMM clustering usually does not perform well on high dimensional data (Zhao, Shrivastava, and Tsui, 2018). As a result, a network embedding method is applied to a GCN to get low dimensional node embeddings. Then, GMM clustering is applied to the node embeddings to detect topological modules. On each topological module, module eigengene significance (MES) score (Langfelder and Horvath, 2008) is employed to measure the module significance score. The topological modules with high MES scores are regarded as network biomarkers.

To evaluate the performance of the proposed method, we conduct empirical studies on a set of eight cancer datasets. The results show that the MES scores of the network biomarkers detected by the proposed method are higher than that of the network biomarkers detected by the most popular method in most datasets. To find whether the performance gain is from network embedding or GMM clustering, we have compared GMM clustering against a distance based method, and the inputs of both methods are node embeddings. The results suggest that there is a performance gain in employing GMM clustering than employing a distance based clustering method in most of the datasets. Furthermore, we have conducted the gene enrichment analysis on the network biomarkers detected by the proposed method, and the network biomarkers can be annotated with gene functions that are related to cancer.

4.2 Method

There are four steps in the proposed method as shown in Figure 4.1. The first step is to construct GCN from gene expression data. The second step is to apply network embedding to GCN to obtain low dimensional node embeddings. The third step is to apply GMM clustering to node embeddings to detect topological modules. The final step is to measure the module significance score of each topological module.

4.2.1 Gene co-expression network construction

A GCN is constructed from the gene expression matrix X to represent the real world gene interactions. In NBD-GMMC, a combination of soft threshold $st \in (1, +\infty)$ and hard threshold $ht \in (0, 1)$ is used to construct edges in the

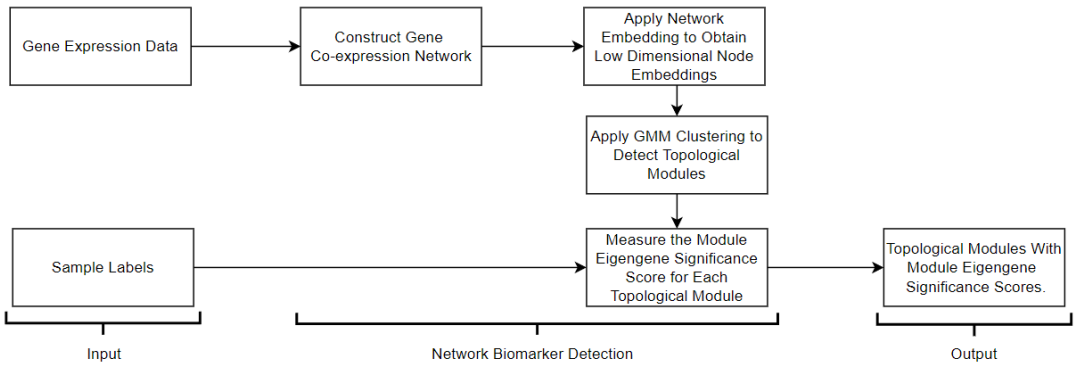


FIGURE 4.1: The proposed framework takes gene expression data and sample labels as input. The output is a list of the topological modules and the corresponding module significant scores.

GCN. For two node $v_i, v_j \in V$, the weight $w_{i,j}$ is calculated as:

$$w_{i,j} = \begin{cases} |corr(X_i, X_j)|^{st} & \text{if } |corr(X_i, X_j)|^{st} > ht \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

Where X_i, X_j represent the gene expression across samples for v_i and v_j respectively, $corr(X_i, X_j)$ represents the Pearson's correlation coefficient between X_i and X_j.

There are three reasons for using *Eq*. (4.1) to construct edges. First, the absolute value of Pearson's correlation coefficient would allow the network embedding method to handle negative regulation relations. Second, the soft threshold st is used to increase the gap between strong correlations and weak/median correlations. Third, the hard threshold ht is used to decrease the computational cost of the network embedding method.

4.2.2 Network embedding for a gene co-expression network

The modified DeepWalk (Perozzi, Al-Rfou, and Skiena, 2014) is applied to GCNs to obtain low dimensional node embeddings, as GMM clustering does not work well on the high dimensional data (Zhao, Shrivastava, and Tsui, 2018). The reason of the choice of the modified DeepWalk is that it could

achieve the state of the art performance in the task of network reconstruction (Goyal and Ferrara, 2018), as a result, the node embeddings learned from the modified DeepWalk can well preserve the similarity between nodes in a network.

The modified DeepWalk employs random walks and a sliding window to capture the topological structure of a network. From each node in $v_i \in V$, r random walks with the length l are simulated. For a node sequence s starting with node v_i, the first node s_0 is v_i. The following node in s is randomly selected from the neighbours $\mathcal{N}(v_i)$ based on the edges weights. The probability of nodes from $\mathcal{N}(v_i)$ been selected is:

$$P(v_j|v_i) = \begin{cases} \frac{w_{i,j}}{\sum_{v_{j'} \in \mathcal{N}(v_i)} w_{i,j'}} & \text{if } v_j \in \mathcal{N}(v_i) \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

The reason for using *Eq*. (4.2) to define the probability of a neighbour being selected is to ensure the sum of all the nodes being selected to 1. The above process is repeated until the length of the node sequence s reaches l. It is worth mentioning that, without ht, the GCN would be a fully connected network. As a result, *Eq*. (4.2) would be very expensive to compute. With the introduction of ht, the number of $\mathcal{N}(v_i)$ would be smaller than $|V| - 1$, hence, the running cost of the network embedding method can be reduced.

To capture the topological structures of the GCN, a window with the size of $2 * w + 1$ slides across the node sequence s, and node pairs are constructed between the node in the centre and other nodes in the sliding window. Node pairs from all the node sequences are saved in the training node pairs $\mathcal{D}$.

Skip-gram negative sampling (SGNS) (Mikolov et al., 2013) is applied to decode the node proximity information in $\mathcal{D}$, as the computational cost of SGNS is lower than that of the original method in DeepWalk. The intuition behind SGNS is to distinguish node pairs in $\mathcal{D}$ from random node pairs. For

example, given a pair $(v_i, v_j) \in \mathcal{D}$, the task is to distinguish the node v_i from other nodes in V. In SGNS, the objective function for a node pair $(v_i, v_j) \in \mathcal{D}$ is defined as:

$$\max \sim \log \sigma(Z_i \cdot Z_j) + \sum_{j'=1}^{q} \mathbf{E}_{v_{j'} \sim P_{\mathcal{D}}}[log\sigma(-Z_i \cdot Z_{j'})] \tag{4.3}$$

where $P_{\mathcal{D}}$ is the unigram distribution raised to the power of $3/4$, q is the number of negative samples. *Eq*. (4.3) is optimised using stochastic gradient descent over $\mathcal{D}$ to learn the node embeddings Z. In general, the nodes that frequently appear together would be close in the node embeddings.

4.2.3 Clustering

GMM clustering (Blei, Jordan, et al., 2006) is applied to node embeddings Z to detect topological modules. The reason is that each Gaussian distribution has a mean vector μ and a co-variance matrix Σ to describe its shape. Hence, there is a large freedom of the shapes of the topological modules.

The parameters of the GMM is inferred using variational inference (Blei, Jordan, et al., 2006). The advantage of using it instead of expectation maximisation (Reynolds, 2009) is that there is no need to specify the number of modules. To conduct variational inference, the objects (node embeddings) are to assumed to be are drawn from a Dirichlet process. A Dirichlet process is a generative model and it consists the scaling parameter $\alpha \in (0, +\infty)$ and base distribution H. The scaling parameter α decides concentration of objects. The objects are assumed to be drawn from different atoms, that are assumed to be independently drawn from the same base distribution H.

The proportions of objects in the atoms are assumed to be drawn from breaking a unit length stick that is also known as the stick breaking process. The proportion of object in the first atom is drawn from with the following

function:

$$\pi_0 = a_0 \sim \text{Beta}(1, \alpha) \tag{4.4}$$

where π_0 is the proportion of objects in the first atom and a_0 is the randomly drawn number from the Beta(1,α). After a chunk of the unit length stick has been broken by the first atom, the following atom is assumed to break from the remainder of the stick using the same strategy. Hence, the proportion of the k-th atoms, where $k > 0$, is:

$$\pi_i = a_k \prod_{i=1}^{k-1} (1 - a_i) \tag{4.5}$$

where α_k is also randomly drawn from Beta(1,α). By combining *Eq*. (4.4) and *Eq*. (4.5), the number of objects that belong to a sample is not equal in size.

From the node embeddings, the parameters of GMM are learned using variational inference with the aforementioned assumptions. The inferred parameters of GMM are $M = \{\mu_1, \mu_2, ..., \mu_\beta\}$ and $N = \{\Sigma_1, \Sigma_2, ..., \Sigma_\beta\}$, where μ_i is the mean vector of $i-$th Gaussian distribution, Σ is the co-variance matrix of the $i-$th Gaussian distribution, and β is the atom number cutoff. β is introduced, as it is impossible to infer an infinite number of parameters. Note that, even the atom number cutoff β been introduced, the proportion a of some atoms will be set to close to zero, as a result, the number of modules is still largely decided by the node embeddings and the scaling parameter α.

Using the parameters M and N of GMM, the nodes can be grouped into different modules. For any Gaussian distributions containing less than t nodes, where t is the minimum module size, are removed. Then, the corresponding nodes would be grouped to the new most likely Gaussian distributions based on the parameters of the remaining Gaussian distributions.

4.2.4 Measurement of module significance score

The MES score (Langfelder and Horvath, 2008) is applied to the topological modules to measure the module significance scores. The task of measuring module significance score is to measure the strength of correlation between the gene expression values and the sample labels for each topological module.

For a topological module, a gene expression matrix is constructed with all the samples and only the genes in this module. Then, the first principle component (Bishop, 2006) of the transpose of the gene expression matrix is regarded as the module eigengene p. Then, for each topological module, the MES score is measured with the following function:

$$\Lambda = |corr(p, \gamma)| \tag{4.6}$$

where Λ represents the MES score, and the topological modules with high module significance scores are regarded as network biomarkers.

4.2.5 Overview of the proposed method

The framework of NE-NBD is summarised in Algorithm 4. Line 1-3 is the gene co-expression module detection stage. Line 4 is the module significance measurement stage. Finally, in line 5, the gene co-expression modules and the MES scores are the output.

Algorithm 4 NBD-GMMC: network biomarker detection from gene expression network using Gaussian mixture model clustering

Input: gene expression matrix X; sample label γ.

Hyper-parameters: soft threshold st; hard threshold ht; number of walks per node r; walk length l; window size w; number of negative samples q; embedding dimension d; scaling parameter α; atom number cutoff β; minimum module size t.

Output: the gene co-expression modules and the corresponding MES scores.

1: construct a GCN using Eq. (4.1) with the hyper-parameters settings ht, st.
2: apply the modified DeepWalk (Algorithm 5) on the GCN to learn the node embeddings with hyper-parameters settings r, w, q and d.
3: detect topological modules from node embeddings using GMM clustering with the hyper-parameters settings α, β, and t.
4: measure the MES scores for all topological modules.
5: **return** a set of gene modules and the corresponding MES scores.

Algorithm 5 Modified DeepWalk for network embedding

Input: Network $G = (V, E)$.

Hyper-parameters: number of walks per node r; walk length l; window size w; number of negative samples q; embedding dimension d.

Output: node embedding $Z^{|V| \times d}$.

1: **for** each in v_i in V **do**

2: simulate random walks with length l starting from node v_i for r times to generate node sequences.

3: **end for**

4: build node-pair training set $\mathcal{D}$ based on the node pairs generated by the sliding window with size w over each node sequence.

5: train $SGNS$ model with node-pair set $\mathcal{D}$ to learn node embedding Z for the network.

6: **return** node embedding Z.

4.3 Experiments

4.3.1 Datasets

The experiments are conducted on eight datasets from The cancer genome atlas (TCGA) (Weinstein et al., 2013). The datasets are breast invasive carcinoma (BRCA), colon adenocarcinoma (COAD), head-neck squamous cell carcinoma (HNSC), kidney renal cell carcinoma (KIRC), liver hepatocellular carcinoma (LIHC), lung adenocarcinoma (LUAD), prostate adenocarcinoma (PRAD), and thyroid cancer (THCA). Those datasets are all RNA sequencing data. In each dataset, the samples could be divided into two categories: tumour tissue and normal tissue. The numbers of tumour tissues and normal tissues in all of those datasets are summarised in Table 4.1.

TABLE 4.1: The number of samples in the TCGA datasets.

	BRCA	COAD	HNSC	KIRC	LIHC	LUAD	PRAD	THCA
Tumour	1100	287	522	534	373	517	498	509
Normal	112	41	44	72	50	59	52	59

4.3.2 Hyper-parameter settings

The performance of network biomarker detection is evaluated for NBD-GMMC and TOM-HC. TOM-HC is the module detection method from WGCNA, and the reason for its choice is that WGCNA is the most popular method in detecting network biomarkers (Dam et al., 2018).

There are two hyper-parameters in TOM-HC. The first hyper-parameter is the soft st in constructing edges in a GCN. It is set to 6 for all datasets for two reasons. The first reason is that this is the recommended setting for datasets with more than 40 samples in the tutorial of WGCNA. The second reason is that it is infeasible to manually find the optimal setting to fit the scale-free criteria (Langfelder and Horvath, 2008) for each GCN (there are 1200 GCNs in our experiments from 30 independent 5-fold validation on 8 datasets). The second hyper-parameter is the minimum module size t. It is set to be 4 via conducting a parameter sensitivity analysis on BRCA, as there are more samples in BRCA than the other 7 datasets, as shown in Table 4.1. It is worth mentioning that this method is insensitive to the parameter settings (Saelens, Cannoodt, and Saeys, 2018).

There are three subsets of the hyper-parameters in NBD-GMMC. The first subset is from the GCN construction. The hard threshold ht is set to be 0.01 to reduce the running time. The soft threshold st is also set to be 6 as in the comparison method. The effect of ht and st on MES score is discussed in Section 4.3.6. The second subset is from the modified DeepWalk. The embedding dimension d is set to be 5, the number of walk per node r is set to be 10, the walk length l is set to be 80, the number of negative samples q

is set to be 5. All the hyper-parameters except node embedding dimension are following previous works (Perozzi, Al-Rfou, and Skiena, 2014; Grover and Leskovec, 2016), and the effect of node embedding dimension d on the MES score is discussed in Section 4.3.6. The third subset is from the GMM clustering. The atom number cutoff β is set to be $2 \times \sqrt[2]{|V|}$. The scaling parameter α is set to $0.1/\beta$ as we are interested in finding large modules. The minimum module size t is also set to be 4 for fairness. In this paper, all the hyper-parameters are in the aforementioned settings unless otherwise specified.

4.3.3 Experiments on detecting network biomarkers

Following previous works (Langfelder and Horvath, 2008; Tanvir and Mondal, 2019), only genes with high variance are kept to construct GCNs in the aforementioned datasets. The reason is that genes with similar expression values across samples from different categories are not informative. In each dataset, the genes are ranked by the variance, and the top 2000 genes are kept for network biomarker detection, and this setting is similar to Tanvir and Mondal, 2019.

In this section, we investigate the performance in finding network biomarkers that have different expression values between tumour tissues and normal tissues. To evaluate the generalisation performance of the detected network biomarkers, 30 independent 5-fold validation experiments are conducted on each dataset. In each experiment, the samples are categorised as training samples or testing samples. There are four steps in measuring the performance of network biomarker detection for both methods. First, the GCN is constructed using gene expression of training samples. Second, the module detection method is applied to the GCN to detect topological modules. Third, the measurement of training MES score is conducted on

each topological module using the gene expression values and the labels of training samples. The topological module with the highest MES score is regarded as the network biomarker. Finally, measurement of testing MES score is conducted on the network biomarker using the gene expression values and the labels of the testing samples.

The baseline of measuring the performance is the average of MES scores from the 30 independent 5-fold validations, i.e., 150 experiments. However, the average of MES scores can be easily affected by a few outliers. To find whether there is a difference in the mean ranks of the module significance scores from the TOM-HC and NBD-GMMC, Wilcoxon signed-rank test (Wilcoxon, 1992) is applied to two vectors that contain the 150 module significance scores from TOM-HC and NBD-GMMC respectively. In this test, the null hypothesis is that the mean ranks of the module significance scores are the same. In this paper, 0.05 is used as the p-value threshold of rejecting the null hypothesis.

The results of all experiments are summarised in Table 4.2. From Table 4.2, we have the following observations. First, the proposed method could outperform the comparison method in six out of eight datasets. Moreover, there is a large performance gain in COAD and HNSC. This shows the effectiveness of NBD-GMMC in detecting network biomarkers with high discriminate ability. Second, the performance gain of the proposed method is unlikely to be caused by a few outliers from the results of the Wilcoxon signed-rank test. In the datasets that the proposed method can outperform the comparison method, the null hypothesis can be rejected in COAD, HNSC, KIRC, and LIHC. Third, the standard deviation (SD) of the network biomarkers from 30 independent 5-fold validations is largely dataset based. In general, the SDs of MES scores of network biomarkers are smaller in our method than that of the comparison method.

TABLE 4.2: Comparison of performance on detecting network biomarkers.

		TOM-HC		NBD-GMMC		Wilcoxon
		MES score	SD	MES score	SD	p-value
BRCA	Training	0.6603	0.0672	**0.6846**	0.0266	0.1237
	Testing	0.6557	0.0884	**0.6775**	0.0589	0.2273
COAD	Training	0.7112	0.0290	**0.7718**	0.0220	0.0000
	Testing	0.7156	0.0971	**0.7665**	0.0745	0.0000
HNSC	Training	0.5295	0.1163	**0.6671**	0.0256	0.0000
	Testing	0.5025	0.1749	**0.6663**	0.1005	0.0000
KIRC	Training	0.7906	0.0506	**0.8074**	0.0126	0.0229
	Testing	0.7919	0.0866	**0.8202**	0.0453	0.0250
LIHC	Training	**0.6712**	0.0214	0.6665	0.0184	0.0000
	Testing	0.6508	0.0866	**0.6585**	0.0755	0.0003
LUAD	Training	0.8030	0.1177	**0.8163**	0.0549	0.5108
	Testing	0.8026	0.1484	**0.8069**	0.0883	0.4799
PRAD	Training	**0.5134**	0.0225	0.4836	0.0270	0.0000
	Testing	**0.5100**	0.0976	0.4762	0.0978	0.0000
THCA	Training	0.4930	0.0263	**0.5043**	0.0350	0.0000
	Testing	**0.4837**	0.1018	0.4772	0.1123	0.7137

SD is the abbreviation for standard deviation

4.3.4 The effect of Gaussian mixture model clustering

To evaluate the performance gain of using GMM clustering, we conducted additional experiments. In the module detection stage of the proposed method, there are two steps that are network embedding and GMM clustering. Although, we have noticed that there is a performance gain in our method compared with a distance based method, i.e., TOM-HC, it is not clear that the performance gain is from GMM clustering or from network embedding. In order to answer this question, we introduce a new method named network embedding-hierarchical clustering (NE-HC). In the new method, network embedding is applied to GCN to calculate node embeddings. Then, node embeddings are used to construct the dendrogram, in which the distance metric is average distance. Then, dynamic tree cut is applied to the dendrogram to detect topological modules. Finally, MES is applied to each topological module to measure the module significance score. The shared

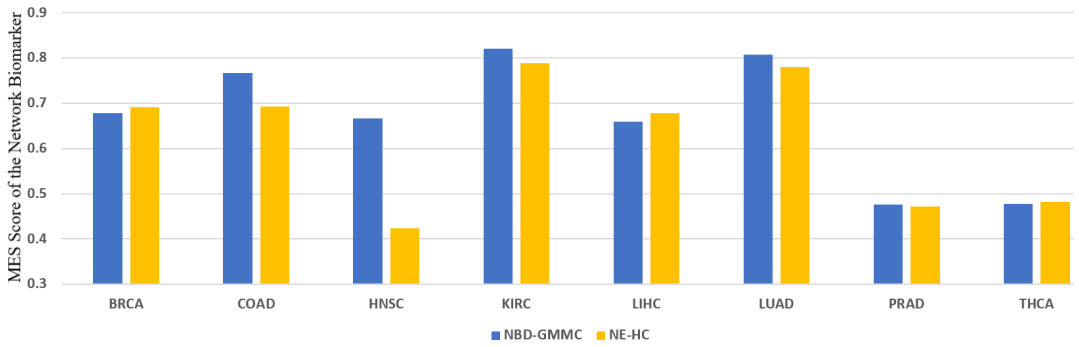


FIGURE 4.2: The effect of using the GMM clustering on the module eigengene significance (MES) score of the detected network biomarkers. NBD-GMMC is the proposed method that employs network embedding coupled with GMM clustering to detect topological modules. NE-HC is the comparison method that employs network embedding coupled with hierarchical clustering (HC) to detect topological modules. From the bar chart, there is a performance gain in most of the datasets by employing GMM clustering over the distance based clustering method. Note that, the inputs for the clustering method, i.e., GMMC and HC, are the same.

hyper-parameters of NE-HC and NE-GMMC are in the same settings.

We have conducted 30 independent 5-fold validations on all datasets for NE-HC. The experiment result of the network biomarkers on the testing datasets is shown in Figure 4.2. From which, we can conclude that the proposed method can outperform NE-HC in most datasets. Although the input of GMM clustering and HC is the same, there is a performance gain in using GMM clustering. The reason behind this is that a distance based method may not capture the topological structure of the gene functional modules and gene disease modules. Moreover, by combing Table 4.2 and Figure 4.2, NE-HC and TOM-HC shows similar result in datasets such as COAD, HNSC, KIRC, LUAD and the proposed method can outperform them in those datasets, this is further evidence that most of the performance gain in Table 4.2 is from the employment of GMM clustering.

4.3.5 Gene enrichment analysis of the network biomarkers

From the previous experiment, the proposed framework can detect network biomarkers with high module significance. In here, we conduct gene enrichment analysis on the network biomarker detected from the proposed framework to find the whether they include known gene interactions, we used the Stringdb (Szklarczyk et al., 2015) to conduct functional gene ontology (GO) analysis and KEGG pathway analysis (Kanehisa and Goto, 2000). In the gene enrichment analysis, false discovery rate cutoff is set as 0.05. Note that, the network biomarkers reported in here are from the first run of the 150 experiments for both BRCA and KIRC.

BRCA

The first network biomarker is from BRCA. The known interactions between genes are plotted in Figure 4.3 and gene enrichment analysis is shown in Table 4.3. In this network biomarker, the number of genes in the network biomarker is 106, and the number of gene interactions between them is 745. In gene GO and KEGG terms, two of them can be supported by recent studies to be related to cancer. First, the PPAR signaling would regulate tumour growth (Tachibana, Yamasaki, Ishimoto, et al., 2008), and PPAR pathway could be an important predictor of breast cancer (Chen et al., 2012), moreover, the dataset BRCA is constructed from patients with breast cancer. Second, adipocytes lipolysis has been shown to be involved in cancer (Yang and Mottillo, 2020), and the regulation of lipolysis in adipocytes has been annotated to the network biomarker.

KIRC

The second network biomarker is from KIRC. The known interactions between genes are plotted in Figure 4.4 and the gene enrichment analysis is

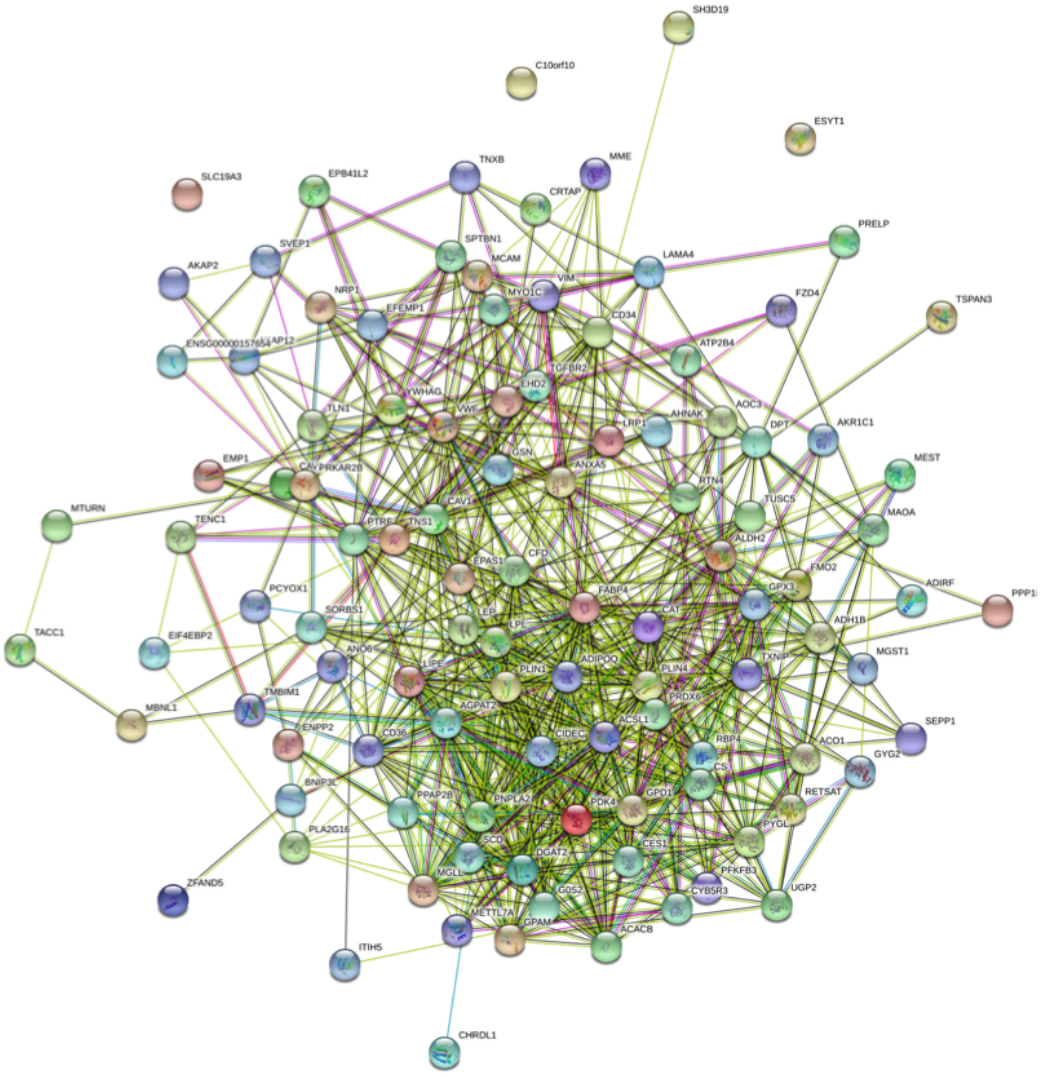


FIGURE 4.3: The known interactions in the network biomarker detected by NBD-GMMC from BRCA. The network biomarker is plotted using StringDB (Szklarczyk et al., 2015). In this figure, a node represents a gene, and an edge represents a known pair-wise gene interaction.

also shown in Table 4.3. In this network biomarker, the number of genes in the network biomarker is 88, and the number of gene interactions between them is 327. In this network biomarker, many of the functions are ion transporter activity, and the ion transporter could be a biomarker for various types of cancer (Lastraioli, Iorio, and Arcangeli, 2015).

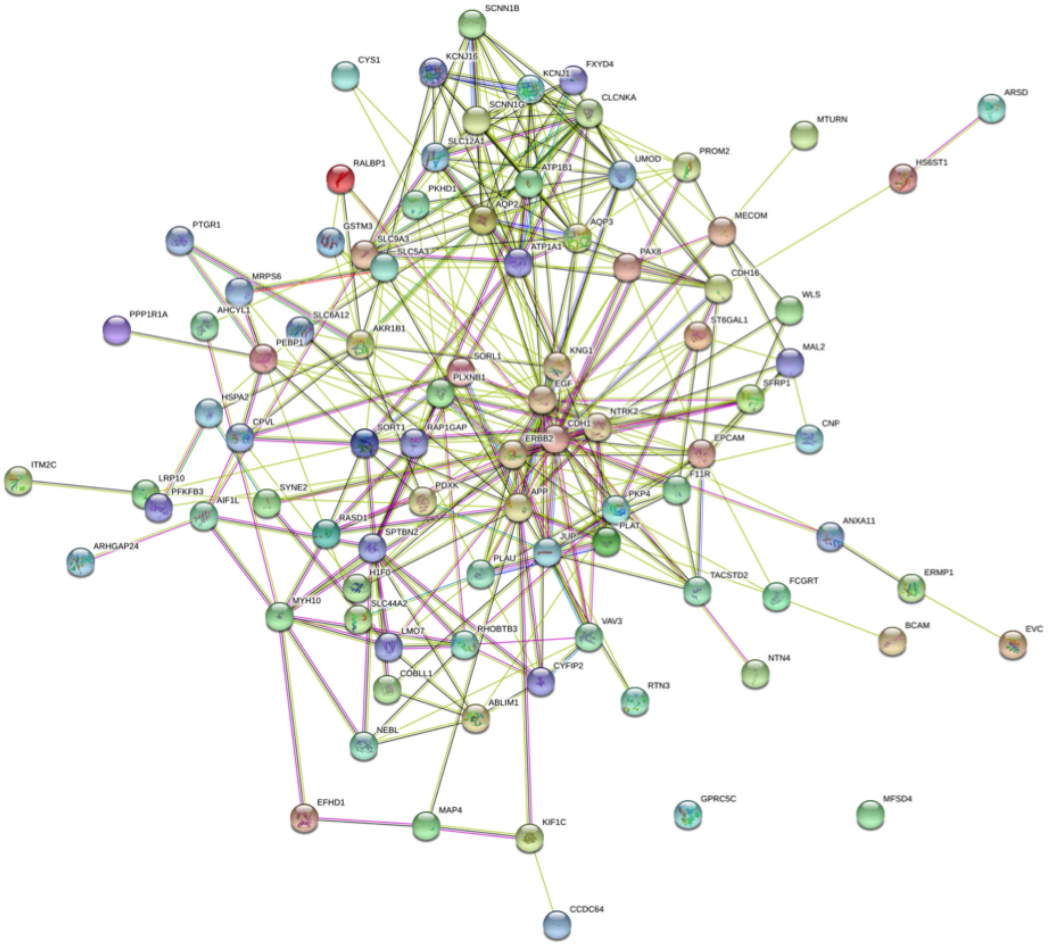


FIGURE 4.4: The known interactions in the network biomarker detected by NBD-GMMC from KIRC. The network biomarker is plotted using StringDB (Szklarczyk et al., 2015). In this figure, a node represents a gene, and an edge represents a known pair-wise gene interaction.

LIHC

The third network biomarker is from LIHC. The known interactions between genes are plotted in Figure 4.5 and the gene enrichment analysis is also shown in Table 4.3. In this network biomarker, the number of genes in

the network biomarker is 190, and the number of gene interactions between them is 2707. In gene GO and KEGG terms, several of them can be supported by recent studies to be related to cancer. First, the PPAR signalling would regulate tumour growth (Tachibana, Yamasaki, Ishimoto, et al., 2008) as discussed in the analysis of the network biomarker in BRCA. Second, fatty acid metabolism would alternate in the tumour (Currie et al., 2013). Third, Tryptophan metabolism is a treatment of cancer (Platten et al., 2019). Finally, Heme binding is annotated to the network biomarker and, Heme is a prominent player in multiple aspects of tumours (Fiorito et al., 2020).

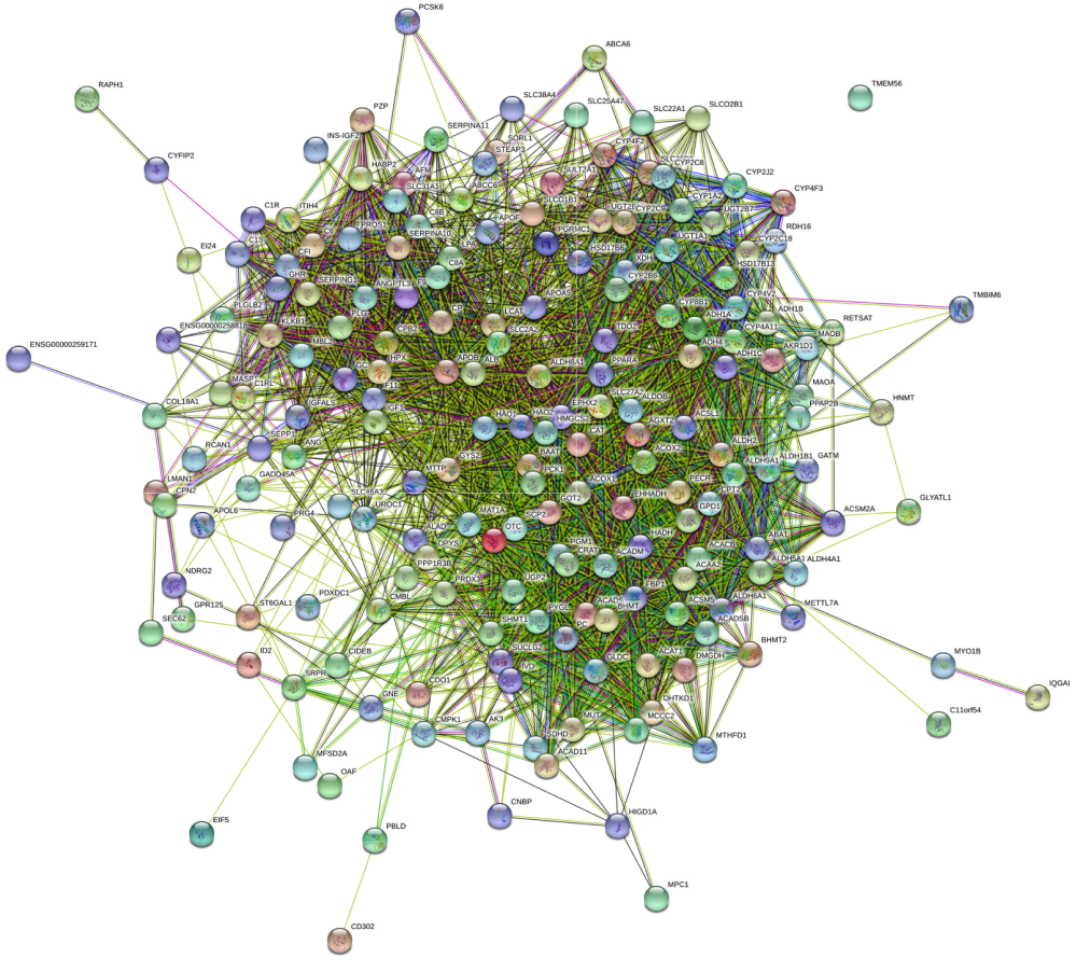


FIGURE 4.5: The known interactions in the network biomarker detected by NBD-GMMC from LIHC. The network biomarker is plotted using StringDB (Szklarczyk et al., 2015). In this figure, a node represents a gene, and an edge represents a known pair-wise gene interaction.

TABLE 4.3: The gene enrichment analysis of the network biomarkers from BRCA,KIRC and LIHC.

BRCA				
Category	Term	P-value	FDR	Description
Function	GO.0052689	0.0000	0.0272	Carboxylic ester hydrolase activity
Function	GO.0016298	0.0000	0.0272	Lipase activity
KEGG	map03320	0.0000	0.0020	PPAR signaling pathway
KEGG	map00561	0.0000	0.0020	Glycerolipid metabolism
KEGG	map04923	0.0007	0.0380	Regulation of lipolysis in adipocytes
KIRC				
Category	Term	P-value	FDR	Description
Function	GO.0046873	0.0001	0.0427	Metal ion transmembrane transporter activity
Function	GO.0015081	0.0001	0.0427	Sodium ion transmembrane transporter activity
Function	GO.0015079	0.0003	0.0427	Potassium ion transmembrane transporter activity
KEGG	map04960	0.0003	0.0301	Aldosterone-regulated sodium reabsorption
LIHC				
Category	Term	P-value	FDR	Description
Function	GO.0003824	0.0000	0.0000	Catalytic activity
Function	GO.0048037	0.0000	0.0000	Cofactor binding
Function	GO.0050662	0.0000	0.0000	Coenzyme binding
Function	GO.0016491	0.0000	0.0000	Oxidoreductase activity
Function	GO.0036094	0.0000	0.0012	Small molecule binding
Function	GO.0046906	0.0000	0.0039	Tetrapyrrole binding
Function	GO.0043167	0.0000	0.0039	Ion binding
Function	GO.0020037	0.0000	0.0062	Heme binding
Function	GO.0016903	0.0001	0.0078	Oxidoreductase activity, acting on the aldehyde or oxo group of donors
Function	GO.0019842	0.0003	0.0199	Vitamin binding
Function	GO.0008392	0.0003	0.0199	Arachidonic acid epoxygenase activity
Function	GO.0003995	0.0003	0.0199	Acyl-CoA dehydrogenase activity
Function	GO.0097159	0.0005	0.0236	Organic cyclic compound binding
Function	GO.0016627	0.0005	0.0236	Oxidoreductase activity, acting on the CH-CH group of donors
Function	GO.0043168	0.0009	0.0339	Anion binding
Function	GO.0046914	0.0011	0.0409	Transition metal ion binding
Function	GO.0004497	0.0012	0.0431	Monooxygenase activity
Function	GO.0016725	0.0014	0.0484	Oxidoreductase activity, acting on CH or CH2 groups
Function	GO.0016620	0.0015	0.0485	Oxidoreductase activity, acting on the aldehyde or oxo group of donors, NAD or NADP as acceptor
KEGG	map01100	0.0000	0.0000	Metabolic pathways
KEGG	map00280	0.0000	0.0000	Valine, leucine and isoleucine degradation
KEGG	map00071	0.0000	0.0000	Fatty acid degradation
KEGG	map00830	0.0000	0.0002	Retinol metabolism
KEGG	map04146	0.0000	0.0023	Peroxisome
KEGG	map00380	0.0001	0.0031	Tryptophan metabolism
KEGG	map03320	0.0002	0.0044	PPAR signaling pathway
KEGG	map00650	0.0002	0.0044	Butanoate metabolism
KEGG	map01212	0.0003	0.0060	Fatty acid metabolism
KEGG	map00982	0.0003	0.0060	Drug metabolism - cytochrome P450
KEGG	map00410	0.0005	0.0070	beta-Alanine metabolism
KEGG	map01200	0.0008	0.0107	Carbon metabolism
KEGG	map00640	0.0009	0.0112	Propanoate metabolism
KEGG	map00590	0.0012	0.0140	Arachidonic acid metabolism
KEGG	map00010	0.0012	0.0140	Glycolysis / Gluconeogenesis
KEGG	map00340	0.0014	0.0148	Histidine metabolism
KEGG	map04610	0.0017	0.0166	Complement and coagulation cascades
KEGG	map05204	0.0018	0.0169	Chemical carcinogenesis
KEGG	map00980	0.0042	0.0374	Metabolism of xenobiotics by cytochrome P450
KEGG	map00630	0.0043	0.0374	Glyoxylate and dicarboxylate metabolism
KEGG	map00330	0.0049	0.0396	Arginine and proline metabolism
KEGG	map00350	0.0055	0.0419	Tyrosine metabolism

4.3.6 Parameter sensitivity

There are many hyper-parameters in the proposed method. The hyper-parameters are from three parts: the construction of the GCN, the modified DeepWalk, and GMM clustering. Most of those hyper-parameters have been studied in the previous works. In here, we evaluate three hyper-parameters that are the hard threshold ht, the soft threshold st, and the embedding dimension d. In this section, the experiments are conducted on BRCA and KIRC. 30 independent 5-fold validations are conducted to measure the impact of those hyper-parameters on the MES scores.

The effect of hard threshold of soft threshold

The hard threshold ht and soft threshold st are used together in the construction of the GCN. In this section, the effect of ht and st on the MES score and wall-clock time (WCT) of the proposed method is discussed, We have conducted a grid search for the combinations of $ht \in \{0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001\}$ and $st \in \{4, 5, 6, 7, 8, 9, 10\}$. The WCT experiment is conducted on a computer, and the CPU is Intel 7700HQ, and parallel computing is utilised for random walks and SGNS.

The result of the effect on detecting network biomarkers has been shown in Table 4.4 and Table 4.5. From Table 4.4 and Table 4.5, there are three observations. First, the proposed method is not very sensitive to the selections of ht and st. It can be seen that of the 35 experimental settings on two datasets, the MES scores are very similar. Second, the performance gain of the proposed method shown in Table 4.2 is unlikely to be caused by settings of hard threshold. It can be seen that even with a very small ht, e.g., $ht = 0.000001$, the performance of the proposed method is still higher than that of the comparison method in both of the datasets. Third, the effect of ht and st is data dependent. It can be seen that NBD-GMMC performs slightly better with

$ht = 0.1$ and $ht = 0.01$ for all st settings in BRCA. However, such a trend can not be found in KIRC. The reason is that the structures of gene topological modules are different between the two datasets.

The result of the effect on WCT has been shown in Table 4.6. From Table 4.6, there are two observations. First, given any ht, with the increasing of st, the WCT would decrease. Second, given any st, with the increasing of ht, the WCT would decrease. The reason behind those observations is that GCN would be sparse with a large ht and st, as shown in *Eq*. (4.1). As a result, the computational cost of the random walk process, which is shown in *Eq*. (4.2), would be reduced. Hence, the WCT would decrease.

The effect of node embedding dimension

The hyper-parameter d decides the dimension of the embeddings of the nodes in a GCN. It could have an impact on both the number of detected modules and the MES scores. The results of the parameter sensitivity of $d \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ have been shown in Table 4.7. From Table 4.7, we have three observations. First, the number of modules keeps increasing with the increasing embedding dimension. The reason behind this is that the node embeddings would be more informative with the increasing embedding dimension. Second, MES scores of the network biomarkers would be relatively low with a small d. The proposed method shows poor performance at BRCA with $d = 1$. The proposed method shows poor performance at KIRC with $d = 1$ and $d = 2$. The reason is that node embedding is not informative enough with a very small d, hence, genes from different functional processes could be grouped together. Third, the MES scores of the network biomarkers would not keep increasing with the increase of d. The reason is that node embedding would be too informative, hence, large gene functional modules could break up into several smaller topological modules.

TABLE 4.4: The impact of hard threshold and soft threshold on BRCA.

		ht=0.1		*ht*= 0.01		*ht*=0.001	
		MES score	SD	MES score	SD	MES score	SD
st=4	Training	0.6983	0.0120	0.6749	0.0321	0.6746	0.0269
	Testing	0.6968	0.0516	0.6644	0.0573	0.6642	0.0588
st=5	Training	0.6958	0.0158	0.6733	0.0320	0.6755	0.0316
	Testing	0.7085	0.0398	0.6643	0.0612	0.6623	0.0772
st=6	Training	0.6989	0.0206	0.6846	0.0266	0.6806	0.0282
	Testing	0.6947	0.0518	0.6775	0.0589	0.6750	0.0537
st=7	Training	0.6803	0.0261	0.6937	0.0129	0.6827	0.0253
	Testing	0.6717	0.0599	0.6897	0.0487	0.6742	0.0607
st=8	Training	0.6980	0.0185	0.6938	0.0135	0.6702	0.0319
	Testing	0.6965	0.0608	0.6964	0.0436	0.6631	0.0638
st=9	Training	0.6901	0.0209	0.6922	0.0152	0.6744	0.0303
	Testing	0.6810	0.0621	0.6991	0.0498	0.6650	0.0612
st=10	Training	0.6779	0.0220	0.6927	0.0176	0.6883	0.0311
	Testing	0.6737	0.0598	0.7023	0.0417	0.6922	0.0610
		ht=0.0001		*ht*=0.00001		*ht*=0.000001	
		MES score	SD	MES score	SD	MES score	SD
st=4	Training	0.6711	0.0303	0.6732	0.0276	0.6749	0.0274
	Testing	0.6614	0.0780	0.6651	0.0716	0.6641	0.0759
st=5	Training	0.6824	0.0244	0.6825	0.0238	0.6826	0.0263
	testing	0.6706	0.0824	0.6701	0.0817	0.6752	0.0490
st=6	Training	0.6835	0.0249	0.6888	0.0191	0.6839	0.0249
	Testing	0.6773	0.0531	0.6854	0.0484	0.6801	0.0513
st=7	Training	0.6803	0.0261	0.6854	0.0210	0.6857	0.0219
	Testing	0.6717	0.0599	0.6806	0.0606	0.6830	0.0474
st=8	Training	0.6820	0.0258	0.6838	0.0246	0.6823	0.0237
	Testing	0.6700	0.0762	0.6785	0.0534	0.6776	0.0522
st=9	Training	0.6759	0.0305	0.6798	0.0243	0.6790	0.0248
	Testing	0.6669	0.0684	0.6638	0.0785	0.6667	0.0690
st=10	Training	0.6659	0.0311	0.6702	0.0301	0.6732	0.0312
	Testing	0.6559	0.0610	0.6457	0.0912	0.6635	0.0740

TABLE 4.5: The impact of hard threshold and soft threshold on KIRC.

		ht=0.1		*ht*= 0.01		*ht*=0.001	
		MES score	SD	MES score	SD	MES score	SD
st=4	Training	0.8074	0.0125	0.8038	0.0290	0.8059	0.0278
	Testing	0.8200	0.0454	0.8140	0.0648	0.8130	0.0667
st=5	Training	0.8061	0.0124	0.8075	0.0135	0.8065	0.0129
	Testing	0.8191	0.0456	0.8153	0.0753	0.8191	0.0466
st=6	Training	0.8047	0.0130	0.8074	0.0126	0.8048	0.0188
	Testing	0.8162	0.0451	0.8202	0.0453	0.8174	0.0494
st=7	Training	0.8017	0.0135	0.8072	0.0124	0.8066	0.0127
	Testing	0.8153	0.0458	0.8200	0.0452	0.8192	0.0453
st=8	Training	0.7995	0.0136	0.8069	0.0124	0.8068	0.0126
	Testing	0.8162	0.0486	0.8198	0.0453	0.8197	0.0454
st=9	Training	0.7980	0.0132	0.8064	0.0124	0.8070	0.0124
	Testing	0.8138	0.0478	0.8194	0.0454	0.8198	0.0454
st=10	Training	0.7978	0.0140	0.8056	0.0125	0.8067	0.0123
	Testing	0.8145	0.0490	0.8187	0.0457	0.8196	0.0456
		ht=0.0001		*ht*=0.00001		*ht*=0.000001	
		MES score	SD	MES score	SD	MES score	SD
st=4	Training	0.8030	0.0285	0.8030	0.0296	0.8034	0.0358
	Testing	0.8085	0.0926	0.8108	0.0753	0.8141	0.0642
st=5	Training	0.8064	0.0135	0.8055	0.0136	0.8052	0.0138
	Testing	0.8150	0.0512	0.8175	0.0458	0.8153	0.0575
st=6	Training	0.8062	0.0135	0.8051	0.0137	0.8056	0.0127
	Testing	0.8147	0.0680	0.8187	0.0453	0.8185	0.0459
st=7	Training	0.8043	0.0275	0.8058	0.0127	0.8059	0.0136
	Testing	0.8174	0.0529	0.8189	0.0457	0.8174	0.0474
st=8	Training	0.8061	0.0131	0.8061	0.0128	0.8060	0.0126
	Testing	0.8141	0.0777	0.8192	0.0455	0.8190	0.0458
st=9	Training	0.8066	0.0129	0.8065	0.0124	0.8061	0.0126
	Testing	0.8205	0.0449	0.8196	0.0459	0.8192	0.0457
st=10	Training	0.8064	0.0127	0.8064	0.0126	0.8061	0.0127
	Testing	0.8193	0.0453	0.8193	0.0456	0.8192	0.0456

TABLE 4.6: The wall-clock time of the proposed method on BRCA and KIRC.

BRCA						
	ht =0.1	ht =0.01	ht =0.001	ht =0.0001	ht =0.00001	ht =0.000001
st =4	193	353	857	1290	1827	2474
st =5	180	287	535	840	1269	2063
st =6	173	241	394	575	903	1402
st =7	157	213	312	425	732	965
st =8	146	200	251	350	552	719
st =9	150	186	212	289	451	558
st =10	145	185	194	259	384	448
KIRC						
	ht=0.1	ht =0.01	ht =0.001	ht =0.0001	ht =0.00001	ht =0.000001
st =4	198	511	1220	2076	2669	3101
st =5	141	335	741	1398	2033	2548
st =6	115	259	498	914	1483	2020
st =7	106	215	376	652	1080	1531
st =8	97	179	304	496	780	1178
st =9	93	157	258	393	596	885
st =10	90	137	224	324	477	694

The wall-clock time is measured in seconds

TABLE 4.7: The impact of the node embedding dimension d on BRCA and KIRC.

BRCA									
Embedding dimension	d =1	d =2	d =3	d =4	d =5	d =6	d =7	d =8	d =9
Training MES score	0.2383	0.6670	0.6884	0.6840	0.6846	0.6780	0.6664	0.6586	0.6511
Testing MES score	0.2450	0.6610	0.6811	0.6767	0.6775	0.6632	0.6521	0.6328	0.6358
Number of Modules	3.91	10.67	20.75	34.30	40.67	43.79	46.71	48.42	48.84
KIRC									
Embedding dimension	d =1	d =2	d =3	d =4	d =5	d =6	d =7	d =8	d =9
Training MES score	0.5305	0.5366	0.7850	0.8058	0.8074	0.8072	0.8070	0.8072	0.8067
Testing MES score	0.5266	0.4952	0.7728	0.8179	0.8202	0.8200	0.8194	0.8168	0.8193
Number of Modules	3.21	11.04	20.76	32.12	41.18	52.08	61.05	67.92	72.25

4.4 Chapter summary

To allow high freedom of the shapes of the topological modules, we propose NBD-GMMC. From the empirical study, NBD-GMMC can outperform the most popular method in most of the datasets. Further experiments suggest that some of the performance gain is from the employment of GMM clustering. Finally, the gene enrichment analysis on the network biomarkers detected from BRCA, KIRC, and LIHC suggest that the network biomarker can be annotated with cancer related gene functions.

Chapter 5

Network embedding based drug-target interaction reaction

This chapter focuses on answering the third research question that is introduced Section 1.4.3. Section 5.1 provides the intuition behind our approach. Section 5.2 provides the details of our method. Section 5.3 validates our approach on a number of benchmark DTI datasets. Finally, Section 5.4 summarises the chapter.

5.1 Introduction

Identifying novel drug-target interactions (DTIs) is a crucial step in drug development. Since experimentally determining DTIs is expensive and time-consuming (Haggarty et al., 2003), it is desirable to develop computational methods to identify promising candidate DTIs to accelerate the speed of drug development.

Machine learning based methods have attracted a lot of research interests in recent years. Most of them focus on identifying novel DTIs from known drugs and targets, as a drug might bind with more than one target (Xie et al., 2012), which could lead to successful drug re-positioning. Many methods transform the DTI prediction problem into a binary classification problem. The known edges in a given DTI network are assumed to be positive

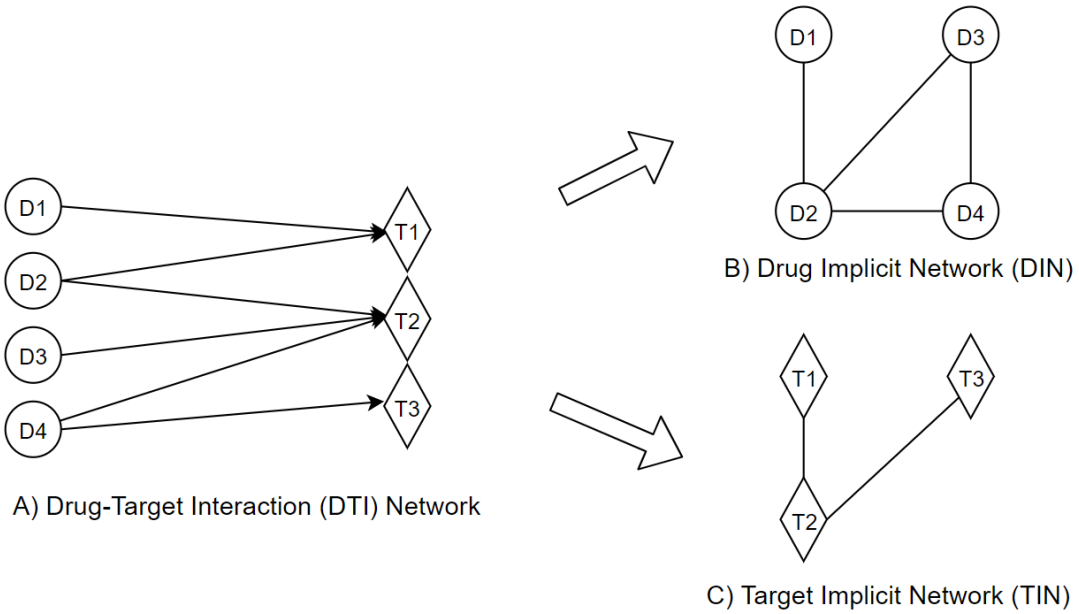


FIGURE 5.1: An example of the implicit network construction. A circle represents a drug, and a diamond represents a target. From the DTI network, the DIN and TIN are constructed based on the common neighbours.

samples and the unknown edges are assumed to be negative samples. The machine learning based methods aim to distinguish positive samples and negative samples based on prior knowledge about drugs and targets, e.g., known DTIs, drug chemical structures, target sequences, etc.

Although many computational methods have been proposed to tackle the DTI prediction problem, to the best of our knowledge, the implicit networks constructed from the known DTI network have been ignored in the existing works. However, it has been shown that the implicit networks constructed from the bipartite network can improve the performance of the link prediction task in bipartite network (Wang et al., 2019). Besides, the implicit relations can also improve the performance of recommender systems (Yu et al., 2018).

Motivated by the success of using implicit networks or relations in other problems, we suggest also considering the implicit networks extracted from the DTI network (a bipartite network) in the DTI prediction problem. The implicit networks, i.e., drug implicit network (DIN) and target implicit network

(TIN) as shown in Figure 5.1, are constructed using the second-order proximity of the DTI network. It is worth noticing that, an edge in the DIN indicates that two drugs would bind with at least one common targets and an edge in the TIN indicates that two targets would bind with at least one common drugs, while the DSSN and TSSN represent the structural similarity directly calculated between either two drugs or two targets. The topological structures of DIN and TIN would be different from the topological structures of TSSN and DSSN, as they are computed from different perspectives.

To incorporate the implicit networks, we propose a method termed as Network Embedding based Drug-Target Interaction Prediction (NE-DTIP). The experiments suggest that NE-DTIP can out-perform other methods in the datasets with dense implicit networks, and there is a performance gain in the incorporation of the implicit networks. We also conduct a case study to verify the top twenty DTI predictions by the proposed method on the latest dataset, and it is interesting to find that six out of twenty novel DTIs (i.e., new DTIs not recorded in the dataset) are supported by recent studies.

5.2 Method

NE-DTIP is a machine learning based method, and it includes two stages: the feature vector construction stage and the DTI classification stage.

During the feature vector construction stage, DIN and TIN are constructed from the DTI network. After that, drug embeddings and target embeddings are learned from DIN, TIN, DSSN, and TSSN using a network embedding method.

During the DTI classification stage, the edges in the DTI network are regarded as positive samples, while randomly sampled unknown drug-target

pairs in the DTI network are regarded as negative samples. The feature vector of each training sample (for a drug-target pair) is constructed via concatenating the corresponding drug embeddings and target embeddings. Finally, all the training samples are employed to learn the parameters of a binary classifier for DTI predictions.

5.2.1 Network construction

The construction of the implicit networks

The implicit networks $G^{DIN} = (V^D, E^{DIN})$ and $G^{TIN} = (V^T, E^{TIN})$ are constructed from the DTI network $B = (V^D, V^T, E^{DTI})$. Both of the implicit networks are homogeneous networks, each node represents a drug in G^{DIN} and a target in G^{TIN}, and an edge between two nodes indicates the corresponding nodes share at least one common neighbour(s) in the DTI network. Following the definition in Deng, Lyu, and King, 2009, for any node pair in DIN and TIN, the edge weight can be defined as:

$$w_{ij}^{DIN} = \sum_{k \in V^T} w_{ik}^{DTI} w_{jk}^{DTI} \tag{5.1}$$

$$w_{ij}^{TIN} = \sum_{k \in V^D} w_{ki}^{DTI} w_{kj}^{DTI} \tag{5.2}$$

where, w_{ij}^{DIN} denotes the edge weight for node pair (v_i^D, v_j^D) in DIN and w_{ij}^{TIN} denotes the edge weight for node pair (v_i^T, v_j^T) in TIN.

Sparse operation for DSSN and TSSN

The proposed method takes pre-computed DSSN and TSSN as input. The edge weight in DSSN can be constructed with Tanimoto coefficient (Willett, Barnard, and Downs, 1998) and the edge weight in TSSN can be constructed with Smith-Waterman Algorithm (Smith, Waterman, et al., 1981). The DSSN and TSSN are densely connected, as each node pair would have a positive

similarity score as weight. However, the similarity scores of many edges are very small. Hence, those edges are not informative. By removing edges with small similarity scores, the time complexity of the network embedding algorithm can be significantly improved. Since it is difficult to set cut-off scores for both drug structural similarity score and target sequence similarity score, we introduce an edge density parameter α to remove the non-informative edges, as it has been shown the DSSN shows a strong community structures via only keeping the edges with the high similarity scores (Vogt et al., 2016). In both DSSN and TSSN, we rank edges by the weights, and use the density parameter α to denote the proportion of edges to keep, i.e., $\alpha = 0.1$ means only the top 10 percent of edges are kept. The effect of α on the performance of DTI prediction is discussed in Section 5.3.5.

5.2.2 Network embedding

A network embedding method is employed to learn low dimensional node embeddings for constructing feature vectors. The reasons are that the rows and columns of the weight matrix of a network only contain first-order proximity information, and they are sparse and high dimensional. Those factors made the weight matrix sub-optimal for the downstream machine learning task. To construct feature vectors of the DTIs, the modified DeepWalk is applied to learn the embeddings of drugs and targets. The reason for the choice of the modified DeepWalk is the same as discussed in Section 4.2.2.

The modified DeepWalk, that is introduced in Section 4.2.2, is applied on TIN, DIN, TSSN and DSSN to learn drug and target embeddings and the embeddings from the four networks can be denoted as Z^{TIN}, Z^{DIN}, Z^{TSSN} and Z^{DSSN} respectively.

5.2.3 Feature vector construction for training samples

The feature vectors of training samples are constructed based on the learned embeddings: Z^{TIN}, Z^{DIN}, Z^{TSSN} and Z^{DSSN}. The positive samples are the edges from the DTI networks. For the $m-th$ drug-target pair $(v_i^D, v_j^T) \in E^{DTI}$, the feature vector of positive samples $Z_m^{Positive}$ is obtained by:

$$Z_m^{Positive} = Z_i^{DIN} \oplus Z_i^{DSSN} \oplus Z_j^{TIN} \oplus Z_j^{TSSN} \tag{5.3}$$

where $\oplus$ represents the concatenation of two vectors.

For $m-th$ positive sample, there is a corresponding label $y_m^{positive} = 1$. The negative samples are constructed from the unknown pairs in the DTI network as those pairs have not been experimentally validated, hence, they are unlikely to have interactions. For the $h-th$ randomly sampled drug-target pair $(v_i^D, v_j^T) \notin E^{DTI}$, the feature vector $Z_h^{Negative}$ can be obtained by:

$$Z_h^{Negative} = Z_i^{DIN} \oplus Z_i^{DSSN} \oplus Z_j^{TIN} \oplus Z_j^{TSSN} \tag{5.4}$$

The corresponding label of the negative sample is $y_h^{negative} = -1$. The ratio r of positive sample and negative sample is 1:10, and the effect of r on the performance of DTI prediction is discussed in Section 5.3.5.

5.2.4 Drug-target interaction prediction

A support vector machine (SVM) with *Platt scaling* (Platt et al., 1999) classifier is trained to separate positive and negative samples as SVM has been shown to be accurate and robust (Archana and Elangovan, 2014). For a drug-target pair, the probability of classifying a feature vector as a DTI edge can

be written as:

$$p(y=1|x) = \frac{1}{1+\exp(-(\sum_{k=1}^{p} y_k a_k K(x, x_k) + b))} \tag{5.5}$$

where p is the number of training samples, K indicates the kernel function, $a = (a_1, ..., a_p)$ and b are learnable parameters. The kernel function provides the non-linear ability to classify samples without having to project the features of samples into higher dimensional space.In this work, we use Radial basis function (RBF) kernel (Broomhead and Lowe, 1988) to calculate the distance between samples:

$$K(x, x') = exp(-\gamma||x - x'||^2) \tag{5.6}$$

where γ is a pre-defined parameter. The parameters in *Eq*. (5.5) are *learned* using matrix decomposition via transforming it to a quadratic programming problem (Chang and Lin, 2011), in which a tolerance parameter c is introduced.

After all the parameters in the decision function are learned, we start to infer the DTI prediction matrix M. For all drug-target pairs, the feature vectors are constructed as described in the previous section. After that, all feature vectors for all drug-target pairs are fed into the trained decision function to infer the DTI prediction matrix M.

5.2.5 Overview of the proposed method

The pseudocode is summarised in Algorithm 6. The proposed method takes the DTI network, DSSN and TSSN as input. The output is the DTI prediction matrix M. In line 1, the DIN and TIN are constructed. In line 2, the top α proportion of edges in DSSN and TSSN are kept while removing other edges. In

line 3-5, the drug embeddings and target embeddings are obtained by applying a network embedding method on the DIN, TIN, DSSN, and TSSN. In line 6, feature vectors for positive samples and negative samples are constructed. In line 7, a SVM classifier is trained using the feature vectors of training samples. In line 8, Feature vectors are generated for all drug-target pairs. In line 9, all the feature vectors are fed into the decision function of SVM to infer the DTI prediction matrix M. Finally, in line 10, output the DTI prediction matrix M.

5.2.6 Complexity analysis

The first part of NE-DTIP is the construction of DIN and TIN. For each drug-drug pair and target-target pair, we need to calculate the edge weight. Since traversing over common neighbours in both *Eq* (5.1) and *Eq* (5.2) is required, this complexity of this part can be written as $O(C_1|V^D|^2 + C_2|V^T|^2)$ where C_1 is the assumption of average number of common neighbours between drugs and C_2 is the assumption of average numbers between targets. The second part is sparse operation in DSSN and TSSN. The complexity is $O(|V^D|)$ and $O(|V^T|)$ for the spareness operation of DSSN and TSSN respectively. The third part is the SGNS version of DeepWalk. Since it is applied on all of DSSN, TSSN, DIN and TIN, the complexity of this part is $O(2rlw|V^D| + 2rlw|V^T|)$. The number of training samples for the SVM is $11|E^{DTI}|$ as the ratio between the positive samples and negative samples is 1:10, hence, the classification complexity is $O((11|E^{DTI}|)^3)$ (Abdiansah and Wardoyo, 2015). Finally, we need to predict the classification result for all drug-Target pairs and complexity is $O(|V^D||V^T|)$. The overall complexity is $O(C_1|V^D|^2 + C_2|V^T|^2 + |V^D| + |V^T| + 2rlw|V^D| + 2rlw|V^T| + (11|E^{DTI}|)^3 + |V^D||V^T|)$ or $O(|V^D|^2 + |V^T|^2 + |V^D||V^T| + |E^{DTI}|^3)$ without constants.

Algorithm 6 NE-DTIP

Input: DTI network; DSSN; TSSN; **Hyper-parameters**: density parameter α; kernel parameter for a support vector machine (SVM) classifier γ. the ratio r between positive samples and negative samples.

Output: DTI prediction matrix $M \in \mathbb{R}^{|V^D| \times |V^T|}$.

1: construct DIN and TIN (two implicit networks) from the DTI network. The edge weights in DIN and TIN are constructed by *Eq*. (5.1) and *Eq*. (5.2) respectively.
2: keep the top α proportion of edges in DSSN and TSSN.
3: **for** DIN, TIN, DSSN and TSSN **do**
4: apply Algorithm 5 to learn node embedding Z from a network.
5: **end for**
6: construct feature vectors for positive samples (from known DTIs) and negative samples (from unknown DTIs) and ratio between positive samples and negative samples is r.
7: train a SVM classifier with all training samples, which yields a trained SVM classifier.
8: construct feature vectors for all drug-target pairs.
9: feed all feature vectors (from the last step) into the trained SVM classifier to learn the DTI prediction matrix M.
10: **return** the DTI prediction matrix M.

5.3 Experiments

The experiments are conducted on five benchmark datasets. Four of those datasets are published in Yamanishi et al., 2008 and they are nuclear receptor (NR), G-protein-coupled receptors (GPCR), ion channel (IC) and enzyme (EN). All of those four datasets are obtained from DrugBank (Wishart et al., 2006). The fifth dataset, drug-target inhibition (DT-IN), is published in Luo

et al., 2017, in which, the DTIs are obtained from multiple sources, and only the DTIs with the binding threshold below 10μM are kept.

For all the benchmark datasets, DSSNs and TSSNs are pre-computed. There is a difference in constructing the DSSNs between those datasets. For NR, GPCR, IC, and EN, the drug structural similarity scores are constructed using Hattori et al., 2003, while for DT-IN, the drug structural similarity score is constructed using Tanimoto coefficient (Willett, Barnard, and Downs, 1998). The target sequence similarity scores in all of the five datasets are constructed using Smith-Waterman algorithm (Smith, Waterman, et al., 1981).

TABLE 5.1: The statistics of five benchmark DTI networks.

	NR	GPCR	IC	EN	DT-IN
DTI edge	90	635	1476	2926	4978
Drug number	54	223	210	445	732
Targets number	26	95	204	664	1915
Implicit drug edge	218	2748	2546	5137	25628
Implicit target edge	54	668	8843	15497	46843

5.3.1 Experiment settings

The DTI prediction task can be viewed as classifying an imbalanced datasets. The datasets are imbalanced as there are far more unknown edges than the known edges in the DTI network. Following previous work (Wan et al., 2019), the metric of evaluating the performance of predicting DTI is the area under precision and recall (AUPR) curve, as it is more suitable for imbalanced datasets. Other popular metrics such as receiver operator characteristic curve would give an optimistic evaluation of the prediction (Davis and Goadrich, 2006).

Five independent ten-fold cross validations are conducted to evaluate the performance of the DTI prediction methods on each dataset. To calculate

TABLE 5.2: The AUPR scores on five benchmark datasets.

		Netlaprls	BLM-NII	NRLMF	NeoDTI	NE-DTIP
NR	AUPR	0.2288	**0.3617**	0.3336	0.2308	0.2906
	SD	0.0486	0.0984	0.0629	0.0835	0.0992
GPCR	AUPR	0.4149	0.4578	**0.4979**	0.4966	0.4398
	SD	0.0358	0.0434	0.0392	0.0624	0.0628
IC	AUPR	0.4704	0.4763	0.5201	0.5841	**0.6077**
	SD	0.0291	0.0276	0.0278	0.0356	0.0397
EN	AUPR	0.6930	0.7299	0.7352	0.7844	**0.7963**
	SD	0.0265	0.0285	0.0294	0.0267	0.0248
DT-IN	AUPR	0.7816	0.8024	0.8484	0.8560	**0.8610**
	SD	0.0230	0.0231	0.0186	0.0161	0.0145

SD is the abbreviation of standard deviation, the best results are in bold

the AUPR of the testing edges, the randomly sampled unknown edges are treated as negative samples. Following the experimental setting in NeoDTI (Wan et al., 2019), The ratio between the testing edges and randomly sampled unknown edges is 1:10. To eliminate the randomness caused by the difference in training edges/samples and testing edges/samples, the splitting of the edges in cross validations and the negative samples in calculating AUPR are the same across all the experiments for all the datasets.

5.3.2 Comparison with other state-of-the-art methods

The methods are Netlaprls, BLM-NII, NRLMF and NeoDTI, which have been introduced in related works. Although there are newer methods, those methods take additional networks such as drug-drug interaction and target-target interaction as inputs, yet those additional networks may not be available in some datasets. For fairness, the inputs are DTI network, DSSN and TSSN for all methods. The hyper-parameters of the compared methods are obtained via conducting a grid search on DT-IN with a ten-fold cross validation. The set of hyper-parameters with the highest performance in AUPR is selected to conduct all experiments on all five datasets.

The hyper-parameters of our method consists the density hyper-parameter α, hyper-parameters in DeepWalk and hyper-parameters in SVM. The density hyper-parameter α for DSSN and TSSN is set as 0.1. In the network embedding method, the hyper-parameters for embedding dimension, number of walks, walk length l, window size w, and negative sample number q are set to 128, 10, 80, 10, and 5 respectively, as these parameters have been shown to have good performance in most cases according to Perozzi, Al-Rfou, and Skiena, 2014; Grover and Leskovec, 2016; Hou et al., 2020. The hyper-parameter of RBF kernel in SVM is $\gamma = 1/(11|E^{DTI}|)$ and the tolerance hyper-parameter in training SVM is set to 1, both of which are according to Chang and Lin, 2011. In this chapter, the hyper-parameters are in the aforementioned settings, unless otherwise specified.

The result is summarised in Table 5.2, from which, we have the following observations. First, the proposed method shows very competitive performance compared with the previous method, as the proposed method outperforms state-of-the-art methods in three out of five benchmarks and those three datasets are bigger than the other two datasets. Second, the performance of our method is affected by the implicit relations between drugs and targets. It can be seen from Table 5.1, the TINs from NR and GPCR are far more sparse than those from IC, NE and DT-IN. Due to the sparseness of TINs in NR and GPCR, many nodes are isolated in the corresponding implicit networks. As a result, the embeddings learned from the implicit networks (in NR and GPCR datasets) are less informative, which degrades the performance of our method.

5.3.3 The effect of the implicit networks

To evaluate the performance gain of adding the implicit networks in predicting DTIs, three new methods for constructing feature vectors of drugs and

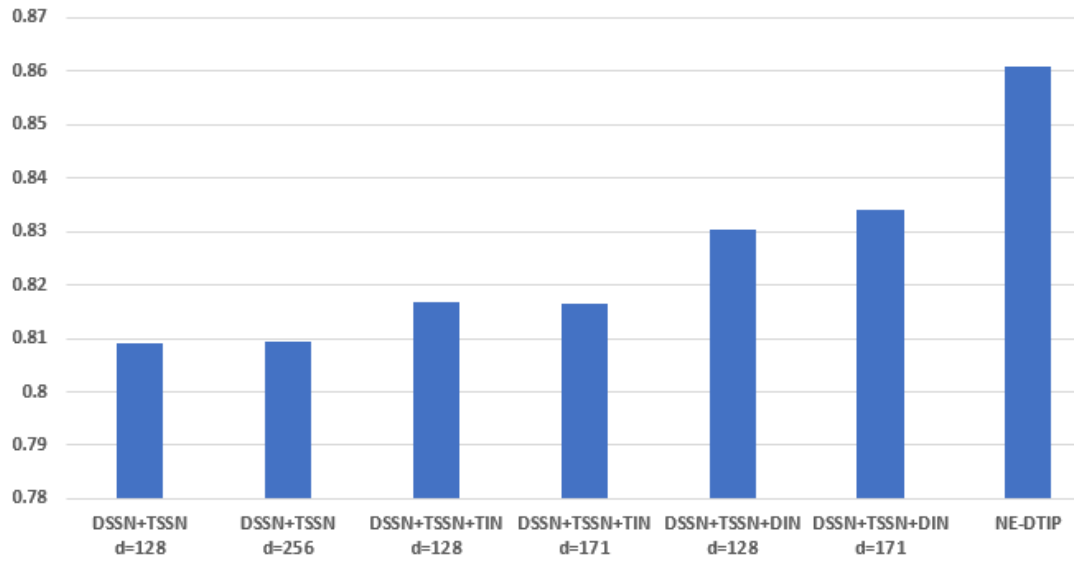


FIGURE 5.2: The AUPR scores of our method on the largest dataset DT-IN. The vertical axis indicates the AUPR scores and the horizontal axis indicates the experimental settings. It can be seen that, there is a performance gain by incorporating the implicit networks, and a large portion of the performance gain is from DIN.

targets have been developed. The first method is using node embeddings from DSSN and TSSN to construct feature vectors and this method is named as DSSN+TSSN. The second method is using DSSN, TSSN and TIN to construct feature vectors and this method is named as DSSN+TSSN+TIN. The second method is using DSSN, TSSN and DIN to construct feature vectors and this method is named as DSSN+TSSN+DIN. The embedding dimension d of DSSN+TSSN is set as 128 and 256, the embedding dimensions d of DSSN+TSSN+DIN and DSSN+TSSN+TIN are set as 128 and 171, the embedding dimension d of NE-DTIP is set as 128 as previous. The reason for the additional dimension setting $d = 256$ for DSSN+TSSN and 171 for both DSSN+TSSN+TIN and DSSN+TSSN+DIN is to make the dimension of feature vectors of DSSN+TSSN, DSSN+TSSN+DIN, DSSN+TSSN+DIN and NE-DTIP to be similar.

The experimental result of the effect of the implicit networks is drawn in Figure 5.2. From Figure 5.2, there are two observations. First, there is a performance gain in incorporating DIN and TIN. It can be seen that the performance gain in incorporating DIN and TIN, as NE-DTIP could achieve the

best results and DSSN+TSSN has the worst results. Second, the performance gain from incorporating DIN is larger than that of incorporating TIN. It can be seen that DSSN+TSSN+DIN is higher than that of DSSN+TSSN+TIN in both experimental settings. This suggests that the DIN may be more informative than the TIN.

5.3.4 Case study on novel drug-target interaction predictions

We conduct a case study on the DT-IN dataset to find whether the novel DTIs (excluding the known DTIs recorded in the dataset) predicted by our method can be supported by recent studies. In this experiment, all the known DTIs are fed into our method to make DTI predictions and the hyper-parameters used are the same as mentioned previously. The top 100 DTIs predicted by our method are shown in Figure 5.3.

Supporting studies have been searched over the top-20 predicted DTIs. It is interesting to find that six of them are supported by recent studies. Sunitinib inhibits EPHB2 (Martinho et al., 2013), GSK3B (Calero et al., 2014), and SYK (Noé et al., 2016). The treatment of using Sunitinib substantially increases ErbB3 (Harvey and Leinwand, 2015). Bosutinib is an inhibitor to KDR (Brown, Nhola, and Herrmann, 2017). Haloperidol would down regulate CHRM2 (Swathy and Banerjee, 2017). It is worth mentioning none of those six predictions is in the top-20 predictions if the implicit networks are not incorporated.

5.3.5 Parameter sensitivity

In the proposed method, there are many hyper-parameters that could be tuned. Most of those parameters have been previously studied, in this section, the impact of the two newly introduced parameters are studied.

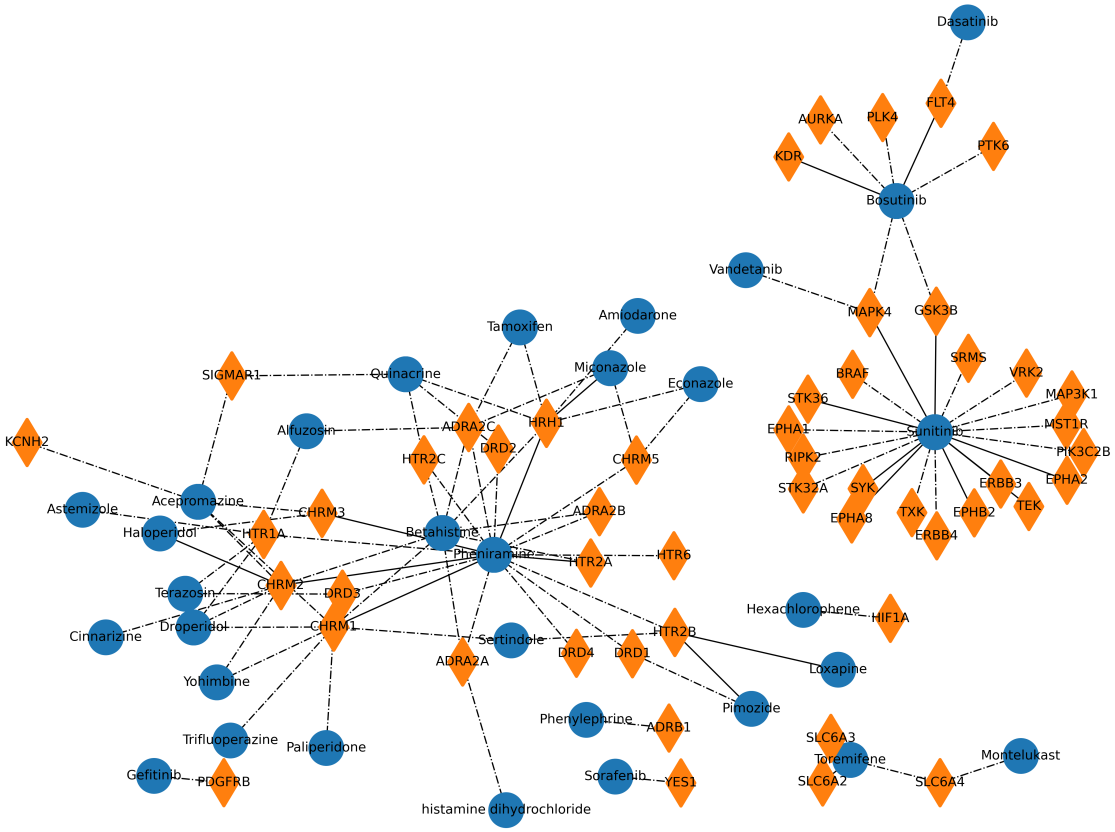


FIGURE 5.3: The top 100 DTIs predicted by the proposed method on DT-IN dataset. Each circle represents a drug and each diamond represents a target. The edges between drugs and targets are the novel DTI predictions (i.e., new DTIs not recorded in the dataset). The top 20 DTI predictions are drawn in the solid lines, while the rest of DTI predictions are drawn in the dashed lines.

TABLE 5.3: The parameter sensitivity analysis result of α on the DT-IN dataset.

		$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.1$	$\alpha = 0.4$	$\alpha = 1.0$
NR	AUPR	0.2713	0.2807	0.2906	0.2870	0.2906
	SD	0.1086	0.1019	0.0992	0.1075	0.1058
	WCT	7	7	7	8	9
GPCR	AUPR	0.4603	0.4546	0.4398	0.4129	0.4267
	SD	0.0601	0.0511	0.0628	0.0500	0.0487
	WCT	76	81	82	96	110
IC	AUPR	0.5844	0.6064	0.6077	0.6042	0.6056
	SD	0.0377	0.0359	0.0397	0.0398	0.0368
	WCT	269	259	264	287	311
EN	AUPR	0.7589	0.7952	0.7963	0.7845	0.7760
	SD	0.0280	0.0245	0.0248	0.0251	0.0243
	WCT	1244	1264	1419	1560	1898
DT-IN	AUPR	0.8547	0.8608	0.8610	0.8536	0.8491
	SD	0.0140	0.0153	0.0145	0.0151	0.0159
	WCT	7413	8125	9147	12557	17876

The wall-clock time is measured in seconds

The density parameter

For the proposed method, the effect of α on DTI prediction is investigated. Here, $\alpha \in \{0.01, 0.04, 0.1, 0.4, 1.0\}$ is investigated, and other parameters as described in Section 5.3.2. The effect of α on DTI prediction is shown in Table 5.3. It can be seen that the proposed method performs the best at either $\alpha = 0.04$ or $\alpha = 0.1$ in most datasets, except for GPCR. The reason for the performance drop of setting $\alpha = 0.01$ is that many informative relations are removed. However, the reason for the performance drop of setting α=0.4 or α=1.0 is that most uninformative relations are kept. By keeping those uninformative edges, node pairs with very small similarity scores could co-exist in the training node pair set in the modified DeepWalk. As a result, the distance between the node embeddings do not reflect the structure/sequence similarity between the nodes. This is further verified by an additional performance loss in setting $\alpha = 1$ compared to set $\alpha = 0.4$.

The Wilcoxon signed-rank test is conducted on dataset DT-IN on the four

pairs of α settings. In this test, the null hypothesis is that the mean ranks of the results from those α settings are the same. The four pairs are ($\alpha = 0.1$, $\alpha = 0.01$), ($\alpha = 0.1$, $\alpha = 0.04$), ($\alpha = 0.1$, $\alpha = 0.4$), ($\alpha = 0.1$, $\alpha = 1.0$), and the corresponding p-values are 0.03, 0.95, 0.02 and 0.00 respectively. Using 0.05 as the cutoff of p-value, the null hypothesis of ($\alpha = 0.1$, $\alpha = 0.01$), ($\alpha = 0.1$, $\alpha = 0.04$), and ($\alpha = 0.1$, $\alpha = 1.0$) can be rejected, while the null hypothesis of ($\alpha = 0.1$, $\alpha = 0.04$) can not be rejected.

The wall-clock time (WCT) of the experiments are also shown in Table 5.3. The measurement of WCT is conducted on a computer with an Intel CPU 7700HQ, and parallel computing is utilised for random walks and SGNS. It is worth mentioning that the impact on WCT of α would be more significant without parallel computing. It can be seen that, with the increase of α, the WCT would increase as well, although, there is an exceptional case of IC at $\alpha = 0.04$. In general, there is an advantage of setting $\alpha = 0.04$ or $\alpha = 0.1$, as those parameters show a good trade-off between performance and efficiency.

The ratio between positive samples and negative samples in training the classifier

In this section, the effect of the ratio r, i.e., the positive samples and negative samples on the performance of DTI prediction is discussed. The experiment results are shown in Figure 5.4. From Figure 5.4, it can be seen that the AUPR would increase gradually with the increase of r until $r = 8$. After that, the performance difference is marginal. Hence, it is beneficial to select an unbalanced ratio between positive samples (from known DTIs) and negative samples (from unknown DTIs). The classifier is able to learn powerful parameters to make predictions with an unbalanced ratio r.

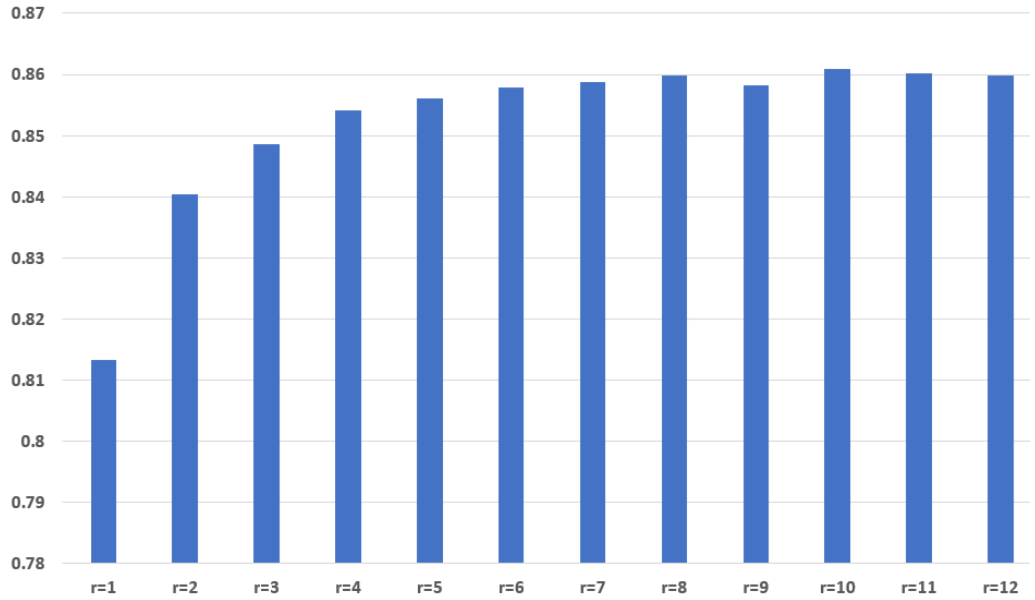


FIGURE 5.4: The parameter sensitivity analysis results of r that is the ratio between positive samples and negative samples in training the classifier. The vertical axis indicates the AUPR scores, and the horizontal axis indicates the experimental settings. The experiments are conducted on the dataset DT-IN. It can be seen that the performance of the proposed method keeps increasing until $r = 8$, after that the difference is marginal.

5.4 Chapter summary

In this work, we propose a novel DTI prediction method, namely NE-DTIP, by additionally considering the implicit networks. Compared against four state-of-the-art methods, the proposed method outperforms the previous methods on three out of five datasets, especially on the datasets with densely connected implicit networks. We also conduct a case study to verify the top twenty DTI predictions by the proposed method on the latest dataset, and it is interesting to find that six out of twenty novel DTIs are supported by recent studies. Further experiments suggest that there is a significant performance gain in incorporating the implicit networks, we have also conducted Wilcoxon signed-rank tests for two pairs: NE-DTIP and DSSN+TSSN with d = 128; NE-DTIP and DSSN+TSSN with d = 171. The p-values of the Wilcoxon signed-rank test of both tests are 0.00, as a result, the performance gain is unlikely to be caused by a few outliers.

Chapter 6

Discussion

This thesis provides a number of applications of the graph-based machine learning methods on multi-omics data analysis. In Chapter 1, we introduce the multi-omics data that are LC-MS untargeted data, the gene expression data, the DTI data. Then, the research problems in the analysis of those data are discussed. Based on the research gap from the previous methods, three research questions are proposed. The literature review of the three research problems are shown in Chapter 2. In Chapter 3, a method for answering the research question from LC-MS untargeted data is introduced. In Chapter 4, a method for answering the research question in gene expression data is introduced. In Chapter 5, a method for answering the research question in DTI data is introduced. This chapter summarises the contributions of this thesis, discusses the limitations of the proposed methods, and introduces some potential future directions for all of the research problems.

6.1 Contributions & conclusions

This thesis focuses on the understanding of multi-omics data. The following types of biological data are studied: the LC-MS untargeted data, the gene expression data, and the DTI data. In this section, the contributions of this thesis are discussed.

6.1.1 Metabolite annotation

In Section 1.4.1, the problem of metabolite annotation for LC-MS untargeted data is defined, and we identify the research gap as the grouping of peaks is non-overlapping in the previous methods. In Chapter 3, we present MAMD, a method for annotating the peaks from LC-MS untargeted data to address this research gap. There are two main contributions in Chapter 3:

- We propose a metabolite annotation method that allows overlapping between the peak modules.
- The peaks from LC-MS untargeted data are represented using an attributed PSN.

MAMD is capable of detecting overlapping peak modules that is not possible on other methods. A peak could belong to more than one peak module, and each peak module is annotated with a metabolite. Moreover, there is an annotation score associated with each peak module. Hence, the output is a ranked list of annotations for each peak.

We have conducted experiments on the standard mixture datasets to evaluate the performance of our approach. The proposed method can outperform the previous methods in most of the experiment settings. The parameter sensitivity analysis suggests the proposed method is insensitive to the settings of the hyper-parameters.

6.1.2 Network biomarker detection

In Section 1.4.2, the problem of network biomarker detection is defined, and we identify the research gap as there are strict constraints in the shapes of the topological modules, and the gene functional modules may not meet those constraints. To address this issue, we represent NBD-GMMC. The main contribution in Chapter 4 is:

- We propose a network biomarker detection method that would allow high freedom of the shapes of the topological modules.

The details of the proposed method are introduced in Chapter 4. The experiments from multiple datasets suggest the proposed method can achieve competitive performance compared against the most popular method. Moreover, we conduct the gene enrichment analysis on the network biomarkers detected by the proposed method. The results suggest the network biomarkers can be annotated with gene functions that are related to cancer.

6.1.3 Drug-target interaction prediction

In Section 1.4.3, the problem of DTI prediction is defined and, we identify the research gap as the implicit networks have been ignored in the prediction of novel DTIs. To address this issue, we represent NBD-GMMC. The main contribution in Chapter 5 is:

- We propose a DTI prediction method that incorporates the implicit networks.

The details of the proposed method are introduced in Chapter 5. The experiment results suggest that NE-DTIP can outperform other methods in the datasets with a large amount of implicit edges in the implicit networks. We have applied NE-DTIP on the dataset DT-IN with all the edges. Of the top-20 novel DTI predictions that are not recorded in DT-IN, six of them can be supported by recent studies, and none of those six DTIs appears in the top-20 predictions without the incorporation of implicit networks.

6.2 Limitations

6.2.1 Limitation of the proposed metabolite annotation method

In Chapter 3, the metabolite annotation method for LC-MS untargeted data has been proposed. There are two major limitations of the proposed method. The first limitation is that the proposed method requires a candidate metabolite list and a candidate adduct list as an input. For a measured peak, if the metabolite and the adduct are not in the those corresponding candidate lists, then, the measured peak can not be correctly annotated. The second limitation is the proposed method is computationally expensive as discussed in Section 3.2.6. Although the proposed method can annotate peaks solely using the first stage of MAMD, it has been found that there is a performance gain in recall via incorporating the second stage of MAMD.

6.2.2 Limitation of the network biomarker detection method

In Chapter 4, a network biomarker detection method from gene expression data has been proposed. A major limitation of the proposed method is that there are many hyper-parameters that need to be selected in the detection of the topological modules. Although, the proposed method would allow high freedom of the gene topological modules, many hyper-parameters are required to determine the scaling of the module detection, such as the embedding dimension, and the effect of embedding dimension on the scaling of module detection is shown in Section 4.3.6.

6.2.3 Limitation of the drug-target interaction prediction method

In Chapter 5, a DTI prediction method from the DTI data is described. However, there are three major limitations of the proposed method. The first is that the proposed method does not perform well on some of the datasets, in which the numbers of edges in the implicit networks are relatively small. The second limitation is the scalability issue, the proposed method requires DSSN and TSSN as input. However, it is very computationally expensive to calculate all the pairwise similarities between drugs and targets. Third, there is a lack of generalisation ability of the proposed method. The proposed method could not deal with unseen drugs or targets.

6.3 Future directions

6.3.1 Future directions for metabolite annotation

For metabolite annotation, future works can be conducted in the following three directions. The first direction could be the incorporation of high-order metabolite co-existence. In MAMD, only direct metabolite reaction pairs are incorporated, however, the metabolite pathway information could be beyond metabolite pairs. One potential direction is to the high-order metabolite co-existence. Second, the experiments results from both positive mode and negative mode can be combined to annotate peaks, as the metabolite which does exist in samples should leave evidence in both modes, and that information could be utilised to annotate peaks. Third, fine turning the rewards of candidate ionised products. Since there are few datasets with known compounds, the candidate ionised product rewards of annotations are assigned based on the common choices of existing algorithms to avoid over-fitting. With more

real world data with known compounds, we could tune the annotation rewards of candidate ionised products in MAMD.

6.3.2 Future directions for biomarker detection

Future research can be conducted in the following directions. First, the GCN could be embedded into hyperbolic space (Nickel and Kiela, 2017). This could potentially reveal the hub genes, while including high-order connectivity, unlike the previous works that only consider direct edges in detecting hub genes. Second, the hyper-parameters could be determined automatically. As discussed in Section 6.2.2, the hyper-parameters could affect the scaling of the module detection process. Third, the low dimensional representations of genes can be learned via applying deep neural networks on the gene expression values. One advantage is that can greatly reduce the computational cost when dealing with a large number genes, as the computational complexity of constructing the GCN is $O(|V|^2)$.

6.3.3 Future directions for drug-target interaction prediction

For DTI prediction, future works can be conducted in the following directions. The first direction is to increase the robustness of incorporating the implicit relations. In NE-DTIP, if the implicit network is sparse, the feature vectors from network embedding would not be informative. As a result, the performance of DTI prediction would be compromised. The second direction is to incorporate the implicit networks from additional information such as drug-drug interaction network, target-target interaction network and drug-side-effect network, as that information has been shown to increase the performance of DTI prediction. Third, a DTI network could be a signed network, since the interaction between a drug-target pair could function as inhibition or exhibition. However, the signs of edges are often ignored. Thus, the third

direction is to develop a computational method that could predict not only the existence of the DTIs, but also the behaviour of the DTIs. Finally, many recent methods employ deep neural networks to learn feature vectors directly from the chemical structure of drugs and protein sequence targets. The feature vectors constructed using deep neural networks have been shown to be powerful to predict DTIs. A direction is to combine the feature vectors constructed using deep neural networks with the embeddings of DIN and TIN to predict novel DTIs.

Bibliography

Abdiansah, Abdiansah and Retantyo Wardoyo (2015). "Time complexity analysis of support vector machines (SVM) in LibSVM". In: *International journal computer and application* 128.3, pp. 28–34.

Akoglu, Leman, Hanghang Tong, and Danai Koutra (2015). "Graph based anomaly detection and description: a survey". In: *Data mining and knowledge discovery* 29.3, pp. 626–688.

Alonso, Arnald et al. (2011). "AStream: an R package for annotating LC/MS metabolomic data". In: *Bioinformatics* 27.9, pp. 1339–1340.

Archana, S and K Elangovan (2014). "Survey of classification techniques in data mining". In: *International Journal of Computer Science and Mobile Applications* 2.2, pp. 65–71.

Bajusz, Dávid, Anita Rácz, and Károly Héberger (2015). "Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?" In: *Journal of cheminformatics* 7.1, pp. 1–13.

Barabási, Albert-László, Natali Gulbahce, and Joseph Loscalzo (2011). "Network medicine: a network-based approach to human disease". In: *Nature reviews genetics* 12.1, pp. 56–68.

Bishop, Christopher M (2006). "Pattern recognition". In: *Machine learning* 128.9.

Bleakley, Kevin and Yoshihiro Yamanishi (2009). "Supervised prediction of drug–target interactions using bipartite local models". In: *Bioinformatics* 25.18, pp. 2397–2403.

Blei, David M, Michael I Jordan, et al. (2006). "Variational inference for Dirichlet process mixtures". In: *Bayesian analysis* 1.1, pp. 121–143.

Blondel, Vincent D et al. (2008). "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10, P10008.

Botstein, David et al. (2000). "Gene Ontology: tool for the unification of biology". In: *Nat genet* 25.1, pp. 25–9.

Broeckling, Corey D et al. (2014). "RAMClust: a novel feature clustering method enables spectral-matching-based annotation for metabolomics data". In: *Analytical chemistry* 86.14, pp. 6812–6817.

Bron, Coen and Joep Kerbosch (1973). "Algorithm 457: finding all cliques of an undirected graph". In: *Communications of the ACM* 16.9, pp. 575–577.

Broomhead, David S and David Lowe (1988). *Radial basis functions, multivariable functional interpolation and adaptive networks*. Tech. rep. Royal Signals and Radar Establishment Malvern (United Kingdom).

Brown, Marie et al. (2011). "Automated workflows for accurate mass-based putative metabolite identification in LC/MS-derived metabolomic datasets". In: *Bioinformatics* 27.8, pp. 1108–1112.

Brown, S-A, L Nhola, and Joerg Herrmann (2017). "Cardiovascular toxicities of small molecule tyrosine kinase inhibitors: an opportunity for systems-based approaches". In: *Clinical Pharmacology & Therapeutics* 101.1, pp. 65–80.

Calero, Raul et al. (2014). "Sunitinib suppress neuroblastoma growth through degradation of MYCN and inhibition of angiogenesis". In: *PloS one* 9.4, e95628.

Cao, Shaosheng, Wei Lu, and Qiongkai Xu (2015). "Grarep: Learning graph representations with global structural information". In: *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 891–900.

Cao, Shaosheng, Wei Lu, and Qiongkai Xu (2016). "Deep neural networks for learning graph representations". In: *Thirtieth AAAI conference on artificial intelligence*.

Chang, Chih-Chung and Chih-Jen Lin (2011). "LIBSVM: A library for support vector machines". In: *ACM transactions on intelligent systems and technology (TIST)* 2.3, pp. 1–27.

Chen, Cheng et al. (2021). "DNN-DTIs: Improved drug-target interactions prediction using XGBoost feature selection and deep neural network". In: *Computers in Biology and Medicine* 136, p. 104676.

Chen, Tianqi and Carlos Guestrin (2016). "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Chen, YZ et al. (2012). "PPAR signaling pathway may be an important predictor of breast cancer response to neoadjuvant chemotherapy". In: *Cancer chemotherapy and pharmacology* 70.5, pp. 637–644.

Chen, Zheng and Heng Ji (2010). "Graph-based clustering for computational linguistics: A survey". In: *Proceedings of TextGraphs-5-2010 Workshop on Graph-based Methods for Natural Language Processing*, pp. 1–9.

Clauset, Aaron, Mark EJ Newman, and Cristopher Moore (2004). "Finding community structure in very large networks". In: *Physical review E* 70.6, p. 066111.

Conesa, Ana and Stephan Beck (2019). "Making multi-omics data accessible to researchers". In: *Scientific data* 6.1, pp. 1–4.

Cox, David R (1972). "Regression models and life-tables". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2, pp. 187–202.

Cui, Peng et al. (2018). "A survey on network embedding". In: *IEEE Transactions on Knowledge and Data Engineering* 31.5, pp. 833–852.

Currie, Erin et al. (2013). "Cellular fatty acid metabolism and cancer". In: *Cell metabolism* 18.2, pp. 153–161.

Daly, Rónán et al. (2014). "MetAssign: probabilistic annotation of metabolites from LC–MS data using a Bayesian clustering approach". In: *Bioinformatics* 30.19, pp. 2764–2771.

Dam, Sipko van et al. (2018). "Gene co-expression analysis for functional classification and gene–disease predictions". In: *Briefings in bioinformatics* 19.4, pp. 575–592.

Davis, Jesse and Mark Goadrich (2006). "The relationship between Precision-Recall and ROC curves". In: *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240.

Del Carratore, Francesco et al. (2019). "Integrated probabilistic annotation: a Bayesian-based annotation method for metabolomic profiles integrating biochemical connections, isotope patterns, and adduct relationships". In: *Analytical chemistry* 91.20, pp. 12799–12807.

Deng, Hongbo, Michael R Lyu, and Irwin King (2009). "A generalized co-hits algorithm and its application to bipartite graphs". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 239–248.

Dittwald, Piotr et al. (2013). "BRAIN: a universal tool for high-throughput calculations of the isotopic distribution for mass spectrometry". In: *Analytical chemistry* 85.4, pp. 1991–1994.

Dunn, Warwick B et al. (2013). "Mass appeal: metabolite identification in mass spectrometry-focused untargeted metabolomics". In: *Metabolomics* 9.1, pp. 44–66.

Fateen, Waleed et al. (2021). "Characterisation of dysplastic liver nodules using low-pass sequencing and the proof of principle on liquid biopsies". In: *Journal of Pathology*.

Fiorito, Veronica et al. (2020). "The multifaceted role of heme in cancer". In: *Frontiers in oncology* 9, p. 1540.

Fortunato, Santo (2010). "Community detection in graphs". In: *Physics reports* 486.3, pp. 75–174.

Fortunato, Santo and Darko Hric (2016). "Community detection in networks: A user guide". In: *Physics Reports* 659, pp. 1–44.

Gaulton, Anna et al. (2012). "ChEMBL: a large-scale bioactivity database for drug discovery". In: *Nucleic acids research* 40.D1, pp. D1100–D1107.

Goyal, Palash and Emilio Ferrara (2018). "Graph embedding techniques, applications, and performance: A survey". In: *Knowledge-Based Systems* 151, pp. 78–94.

Gribskov, Michael, Andrew D McLachlan, and David Eisenberg (1987). "Profile analysis: detection of distantly related proteins". In: *Proceedings of the National Academy of Sciences* 84.13, pp. 4355–4358.

Grover, Aditya and Jure Leskovec (2016). "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 855–864.

Guyon, Isabelle and André Elisseeff (2003). "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar, pp. 1157–1182.

Haggarty, Stephen J et al. (2003). "Multidimensional chemical genetic analysis of diversity-oriented synthesis-derived deacetylase inhibitors using cell-based assays". In: *Chemistry & biology* 10.5, pp. 383–396.

Hartuv, Erez and Ron Shamir (2000). "A clustering algorithm based on graph connectivity". In: *Information processing letters* 76.4-6, pp. 175–181.

Harvey, Pamela Ann and Leslie Anne Leinwand (2015). "Oestrogen enhances cardiotoxicity induced by Sunitinib by regulation of drug transport and metabolism". In: *Cardiovascular research* 107.1, pp. 66–77.

Hattori, Masahiro et al. (2003). "Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways". In: *Journal of the American Chemical Society* 125.39, pp. 11853–11865.

Hou, Chengbin et al. (2020). "GloDyNE: Global Topology Preserving Dynamic Network Embedding". In: *IEEE Transactions on Knowledge and Data Engineering*.

— (2022). "GloDyNE: Global Topology Preserving Dynamic Network Embedding (Extended Abstract)". In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)*.

Hu, ShanShan et al. (2019). "Predicting drug-target interactions from drug structure and protein sequence using novel convolutional neural networks". In: *BMC bioinformatics* 20.25, pp. 1–12.

Huang, Da Wei, Brad T Sherman, and Richard A Lempicki (2009). "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists". In: *Nucleic acids research* 37.1, pp. 1–13.

Huang, Da Wei et al. (2007). "DAVID Bioinformatics Resources: expanded annotation database and novel algorithms to better extract biology from large gene lists". In: *Nucleic acids research* 35.suppl_2, W169–W175.

Jin, Guangxu et al. (2008). "The knowledge-integrated network biomarkers discovery for major adverse cardiac events". In: *Journal of proteome research* 7.9, pp. 4013–4021.

Jin, Yu and Xing Qin (2020). "Co-expression network-based identification of biomarkers correlated with the lymph node metastasis of patients with head and neck squamous cell carcinoma". In: *Bioscience Reports* 40.2.

Kanehisa, Minoru and Susumu Goto (2000). "KEGG: kyoto encyclopedia of genes and genomes". In: *Nucleic acids research* 28.1, pp. 27–30.

Karypis, George and Vipin Kumar (1998). "A fast and high quality multilevel scheme for partitioning irregular graphs". In: *SIAM Journal on scientific Computing* 20.1, pp. 359–392.

Keiser, Michael J et al. (2007). "Relating protein pharmacology by ligand chemistry". In: *Nature biotechnology* 25.2, pp. 197–206.

Keiser, Michael J et al. (2009). "Predicting new molecular targets for known drugs". In: *Nature* 462.7270, pp. 175–181.

Keller, Bernd O et al. (2008). "Interferences and contaminants encountered in modern mass spectrometry". In: *Analytica chimica acta* 627.1, pp. 71–81.

Kernighan, Brian W and Shen Lin (1970). "An efficient heuristic procedure for partitioning graphs". In: *Bell system technical journal* 49.2, pp. 291–307.

Kim, Seon-Young and David J Volsky (2005). "PAGE: parametric analysis of gene set enrichment". In: *BMC bioinformatics* 6.1, pp. 1–12.

Kind, Tobias and Oliver Fiehn (2006). "Metabolomic database annotations via query of elemental compositions: mass accuracy is insufficient even at less than 1 ppm". In: *BMC bioinformatics* 7.1, p. 234.

Kuhl, Carsten et al. (2011). "CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets". In: *Analytical chemistry* 84.1, pp. 283–289.

Kuleshov, Maxim V et al. (2016). "Enrichr: a comprehensive gene set enrichment analysis web server 2016 update". In: *Nucleic acids research* 44.W1, W90–W97.

Langfelder, Peter and Steve Horvath (2008). "WGCNA: an R package for weighted correlation network analysis". In: *BMC bioinformatics* 9.1, p. 559.

Langfelder, Peter, Bin Zhang, and Steve Horvath (2008). "Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R". In: *Bioinformatics* 24.5, pp. 719–720.

Lastraioli, Elena, Jessica Iorio, and Annarosa Arcangeli (2015). "Ion channel expression as promising cancer biomarker". In: *Biochimica et Biophysica Acta (BBA)-Biomembranes* 1848.10, pp. 2685–2702.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444.

Lehmann, Erich Leo and Howard J D'Abrera (1975). *Nonparametrics: statistical methods based on ranks.* Holden-day.

Li, Dong et al. (2020). "Active module identification from multilayer weighted gene co-expression networks: a continuous optimization approach". In: *IEEE/ACM transactions on computational biology and bioinformatics*.

Li, Ze-Rong et al. (2006). "PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence". In: *Nucleic Acids Research* 34.suppl_2, W32–W37.

Liu, Yong et al. (2016). "Neighborhood regularized logistic matrix factorization for drug-target interaction prediction". In: *PLoS computational biology* 12.2, e1004760.

Luo, Yunan et al. (2017). "A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information". In: *Nature communications* 8.1, pp. 1–13.

MacQueen, James et al. (1967). "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA, pp. 281–297.

Martinho, Olga et al. (2013). "In vitro and in vivo analysis of RTK inhibitor efficacy and identification of its novel targets in glioblastomas". In: *Translational oncology* 6.2, 187–IN20.

Mei, Jian-Ping et al. (2013). "Drug–target interaction prediction by learning from local information and neighbors". In: *Bioinformatics* 29.2, pp. 238–245.

Mikolov, Tomas et al. (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Morris, Garrett M et al. (2009). "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility". In: *Journal of computational chemistry* 30.16, pp. 2785–2791.

Mortazavi, Ali et al. (2008). "Mapping and quantifying mammalian transcriptomes by RNA-Seq". In: *Nature methods* 5.7, pp. 621–628.

Mousavian, Zaynab et al. (2016). "Drug–target interaction prediction from PSSM based evolutionary information". In: *Journal of pharmacological and toxicological methods* 78, pp. 42–51.

Newman, Mark EJ (2006). "Finding community structure in networks using the eigenvectors of matrices". In: *Physical review E* 74.3, p. 036104.

Newman, Mark EJ and Michelle Girvan (2004). "Finding and evaluating community structure in networks". In: *Physical review E* 69.2, p. 026113.

Nickel, Maximillian and Douwe Kiela (2017). "Poincaré embeddings for learning hierarchical representations". In: *Advances in neural information processing systems*, pp. 6338–6347.

Noé, Galle et al. (2016). "Clinical and kinomic analysis identifies peripheral blood mononuclear cells as a potential pharmacodynamic biomarker in metastatic renal cell carcinoma patients treated with sunitinib". In: *Oncotarget* 7.41, p. 67507.

Ou, Mingdong et al. (2016). "Asymmetric transitivity preserving graph embedding". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1105–1114.

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena (2014). "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 701–710.

Platt, John et al. (1999). "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods". In: *Advances in large margin classifiers* 10.3, pp. 61–74.

Platten, Michael et al. (2019). "Tryptophan metabolism as a common therapeutic target in cancer, neurodegeneration and beyond". In: *Nature reviews Drug discovery* 18.5, pp. 379–401.

Pons, Pascal and Matthieu Latapy (2005). "Computing communities in large networks using random walks". In: *International symposium on computer and information sciences*. Springer, pp. 284–293.

Raghavan, Usha Nandini, Réka Albert, and Soundar Kumara (2007). "Near linear time algorithm to detect community structures in large-scale networks". In: *Physical review E* 76.3, p. 036106.

Ravasz, Erzsébet et al. (2002). "Hierarchical organization of modularity in metabolic networks". In: *science* 297.5586, pp. 1551–1555.

Rayhan, Farshid et al. (2020). "FRnet-DTI: Deep convolutional neural network for drug-target interaction prediction". In: *Heliyon* 6.3, e03444.

Reynolds, Douglas A (2009). "Gaussian mixture models." In: *Encyclopedia of biometrics* 741, pp. 659–663.

Rodriguez, Juan José, Ludmila I Kuncheva, and Carlos J Alonso (2006). "Rotation forest: A new classifier ensemble method". In: *IEEE transactions on pattern analysis and machine intelligence* 28.10, pp. 1619–1630.

Rogers, Simon et al. (2009). "Probabilistic assignment of formulas to mass peaks in metabolomics experiments". In: *Bioinformatics* 25.4, pp. 512–518.

Rogers, Simon et al. (2011). "Bayesian approaches for mass spectrometry based metabolomics". In:

Rosvall, Martin and Carl T Bergstrom (2007). "An information-theoretic framework for resolving community structure in complex networks". In: *Proceedings of the National Academy of Sciences* 104.18, pp. 7327–7331.

Saelens, Wouter, Robrecht Cannoodt, and Yvan Saeys (2018). "A comprehensive evaluation of module detection methods for gene expression data". In: *Nature communications* 9.1, pp. 1–12.

Scheltema, Richard A et al. (2009). "Simple data-reduction method for high-resolution LC–MS data in metabolomics". In: *Bioanalysis* 1.9, pp. 1551–1557.

Senan, Oriol et al. (2019). "CliqueMS: a computational tool for annotating in-source metabolite ions from LC-MS untargeted metabolomics data based on a coelution similarity network". In: *Bioinformatics* 35.20, pp. 4089–4097.

Shi, Zhiao, Catherine K Derow, and Bing Zhang (2010). "Co-expression module analysis reveals biological processes, genomic gain, and regulatory mechanisms associated with breast cancer progression". In: *BMC systems biology* 4.1, pp. 1–14.

Silva, Ricardo R et al. (2014). "ProbMetab: an R package for Bayesian probabilistic annotation of LC–MS-based metabolomics". In: *Bioinformatics* 30.9, pp. 1336–1337.

Smith, Colin A et al. (2006). "XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification". In: *Analytical chemistry* 78.3, pp. 779–787.

Smith, Temple F, Michael S Waterman, et al. (1981). "Identification of common molecular subsequences". In: *Journal of molecular biology* 147.1, pp. 195–197.

Song, Lin, Peter Langfelder, and Steve Horvath (2012). "Comparison of co-expression measures: mutual information, correlation, and model based indices". In: *BMC bioinformatics* 13.1, p. 328.

Subramanian, Aravind et al. (2007). "GSEA-P: a desktop application for Gene Set Enrichment Analysis". In: *Bioinformatics* 23.23, pp. 3251–3253.

Sumner, Lloyd W et al. (2007). "Proposed minimum reporting standards for chemical analysis". In: *Metabolomics* 3.3, pp. 211–221.

Swathy, Babu and Moinak Banerjee (2017). "Haloperidol induces pharmacoepigenetic response by modulating miRNA expression, global DNA methylation and expression profiles of methylation maintenance genes and genes involved in neurotransmission in neuronal cells". In: *PloS one* 12.9, e0184209.

Szklarczyk, Damian et al. (2015). "STRING v10: protein–protein interaction networks, integrated over the tree of life". In: *Nucleic acids research* 43.D1, pp. D447–D452.

Tachibana, Keisuke, Daisuke Yamasaki, Kenji Ishimoto, et al. (2008). "The role of PPARs in cancer". In: *PPAR research* 2008.

Tang, Jian et al. (2015). "Line: Large-scale information network embedding". In: *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 1067–1077.

Tanvir, Raihanul Bari and Ananda Mohan Mondal (2019). "Cancer Biomarker Discovery from Gene Co-expression Networks Using Community Detection Methods". In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, pp. 2097–2104.

Tanvir, Raihanul Bari et al. (2019). "Graph Theoretic and Pearson Correlation-Based Discovery of Network Biomarkers for Cancer". In: *Data* 4.2, p. 81.

Thafar, Maha A et al. (2020). "DTiGEMS+: drug–target interaction prediction using graph embedding, graph mining, and similarity-based techniques". In: *Journal of Cheminformatics* 12.1, pp. 1–17.

Uppal, Karan, Douglas I Walker, and Dean P Jones (2017). "xMSannotator: an R package for network-based annotation of high-resolution metabolomics data". In: *Analytical chemistry* 89.2, pp. 1063–1067.

Vogt, Martin et al. (2016). "Lessons learned from the design of chemical space networks and opportunities for new applications". In: *Journal of computer-aided molecular design* 30.3, pp. 191–208.

Wan, Fangping et al. (2019). "NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions". In: *Bioinformatics* 35.1, pp. 104–111.

Wang, Daixin, Peng Cui, and Wenwu Zhu (2016). "Structural deep network embedding". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1225–1234.

Wang, Lei et al. (2017a). "Computational methods for the prediction of drug-target interactions from drug fingerprints and protein sequences by stacked auto-encoder deep neural network". In: *International Symposium on Bioinformatics Research and Applications*. Springer, pp. 46–58.

Wang, Xiao et al. (2017b). "Community preserving network embedding". In: *Thirty-first AAAI conference on artificial intelligence*.

Wang, Yaping et al. (2019). "Bipartite network embedding via effective integration of explicit and implicit relations". In: *International Conference on Database Systems for Advanced Applications*. Springer, pp. 435–451.

Wang, Zhong, Mark Gerstein, and Michael Snyder (2009). "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature reviews genetics* 10.1, pp. 57–63.

Weber, Ralf JM and Mark R Viant (2010). "MI-Pack: Increased confidence of metabolite identification in mass spectra by integrating accurate masses and metabolic pathways". In: *Chemometrics and Intelligent Laboratory Systems* 104.1, pp. 75–82.

Weinstein, John N et al. (2013). "The cancer genome atlas pan-cancer analysis project". In: *Nature genetics* 45.10, p. 1113.

Wilcoxon, Frank (1992). "Individual comparisons by ranking methods". In: *Breakthroughs in statistics*. Springer, pp. 196–202.

Willett, Peter, John M Barnard, and Geoffrey M Downs (1998). "Chemical similarity searching". In: *Journal of chemical information and computer sciences* 38.6, pp. 983–996.

Wishart, David S et al. (2006). "DrugBank: a comprehensive resource for in silico drug discovery and exploration". In: *Nucleic acids research* 34.suppl_1, pp. D668–D672.

Wishart, David S et al. (2009). "HMDB: a knowledgebase for the human metabolome". In: *Nucleic acids research* 37.suppl_1, pp. D603–D610.

Wu, Xiaodan, Luonan Chen, and Xiangdong Wang (2014). "Network biomarkers, interaction networks and dynamical network biomarkers in respiratory diseases". In: *Clinical and translational medicine* 3.1, pp. 1–7.

Xia, Zheng et al. (2010). "Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces". In: *BMC systems biology*. Vol. 4. S2. Springer, S6.

Xie, Lei et al. (2012). "Novel computational approaches to polypharmacology as a means to define responses to individual drugs". In: *Annual review of pharmacology and toxicology* 52, pp. 361–379.

Yamanishi, Yoshihiro et al. (2008). "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces". In: *Bioinformatics* 24.13, pp. i232–i240.

Yang, Alexander and Emilio P Mottillo (2020). "Adipocyte lipolysis: from molecular mechanisms of regulation to disease and therapeutics". In: *Biochemical Journal* 477.5, pp. 985–1008.

Yap, Chun Wei (2011). "PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints". In: *Journal of computational chemistry* 32.7, pp. 1466–1474.

Yu, Lu et al. (2018). "Walkranker: A unified pairwise ranking model with multiple relations for item recommendation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1.

Yuan, Lushun et al. (2018). "Co-expression network analysis of biomarkers for adrenocortical carcinoma". In: *Frontiers in genetics* 9, p. 328.

Zeeberg, Barry R et al. (2003). "GoMiner: a resource for biological interpretation of genomic and proteomic data". In: *Genome biology* 4.4, pp. 1–8.

Zeng, Xiangxiang et al. (2020). "Network-based prediction of drug–target interactions using an arbitrary-order proximity embedded deep forest". In: *Bioinformatics* 36.9, pp. 2805–2812.

Zhang, Han et al. (2021). "A Network Embedding Based Approach to Drug-Target Interaction Prediction Using Additional Implicit Networks". In: *International Conference on Artificial Neural Networks*. Springer, pp. 491–503.

Zhao, Yang, Abhishek K Shrivastava, and Kwok Leung Tsui (2018). "Regularized Gaussian mixture model for high-dimensional clustering". In: *IEEE transactions on cybernetics* 49.10, pp. 3677–3688.

Zhou, Yingyao et al. (2019). "Metascape provides a biologist-oriented resource for the analysis of systems-level datasets". In: *Nature communications* 10.1, pp. 1–10.

Zhou, Zhi-Hua and Ji Feng (2017). "Deep forest: towards an alternative to deep neural networks". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3553–3559.

Zong, Nansu et al. (2017). "Deep mining heterogeneous networks of biomedical linked data to predict novel drug–target associations". In: *Bioinformatics* 33.15, pp. 2337–2344.