

Constructing networks and learning node features from noisy data

by

Junyao Kuang

B.S., Harbin Institute of Technology, 2016

M.Eng., Harbin Institute of Technology, 2018

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2023

Abstract

Over the past several decades, there has been a rapid increase in the production of data from scientific experiments and human behaviors, thanks to advancements in technology. The data, which is often collected in various forms and may contain errors, can be effectively analyzed using networks that store both node interactions and properties. The accuracy of network structure estimation plays a crucial role in understanding complex systems and downstream research, such as drug discovery, online commodity recommendations, and gene function analysis. This dissertation contributes to the current understanding of network structure by investigating various network estimation approaches using heterogeneous node interaction and activity data.

First, we propose a maximum *a posteriori* (MAP) estimation approach to reconstruct a target layer in a multilayer network. In this work, we have improved the SimHash algorithm to compute similarities between layers in multilayer networks. The layers that are most similar to the target layer are used to determine the parameters of the conjugate prior. This model allows us to predict missing links and direct experiments to identify nodes with similar functions. We evaluate the approach using two real-world multilayer networks, and the results demonstrate that the MAP estimation is effective in reconstructing the target layer, even when there are a large number of missing links.

Second, to capture the correlation between nodes, we establish a method for constructing a gene co-expression network for the *Anopheles gambiae* transcriptome from 257 unique studies obtained with different methodologies and experimental designs. We introduce the sliding threshold approach to select node pairs with high Pearson correlation coefficients. The resulting network, which we name AgGCN1.0, is robust to random removal of conditions and has similar characteristics to small-world and scale-free networks. Analysis of network sub-graphs revealed that the core is largely comprised of genes that encode components of the

mitochondrial respiratory chain and the ribosome, while different communities are enriched for genes involved in distinct biological processes.

Building upon the network construction approach, we introduce *GeCoNet-Tool*, a comprehensive gene co-expression network construction and analysis tool. The network construction part of the tool offers multiple options for processing gene co-expression data obtained using different technologies. The output is an edge list with the option to assign weights to each link. In the network analysis part, users can generate a table that includes various network properties such as communities, cores, and centrality measures.

Finally, to identify distance relationships between nodes, we propose an unsupervised learning framework that learns node vectors and constructs networks from noisy node activity data. The framework involves designing a scheme to generate random node sequences from node context sets, which are derived from raw node activity data. We utilize a three-layer neural network to train the node sequences and obtain node vectors, which enable us to construct networks and identify nodes with synergistic roles. Additionally, we propose an entropy-based approach to select the most relevant neighbors for each node in the resulting network. The effectiveness of our method is validated using both synthetic and real-world data.

To summarize, this dissertation focuses on developing computational models to extract network structures from heterogeneous data. The models are suitable for the analysis of both node interaction and activity data. The results of this study are expected to enhance our understanding of network structures and benefit the study of complex systems with rich but noisy data.

Constructing networks and learning node features from noisy data

by

Junyao Kuang

B.S., Harbin Institute of Technology, 2016

M.Eng., Harbin Institute of Technology, 2018

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2023

Approved by:

Major Professor
Caterina Scoglio

Copyright

© Junyao Kuang 2023.

Abstract

Over the past several decades, there has been a rapid increase in the production of data from scientific experiments and human behaviors, thanks to advancements in technology. The data, which is often collected in various forms and may contain errors, can be effectively analyzed using networks that store both node interactions and properties. The accuracy of network structure estimation plays a crucial role in understanding complex systems and downstream research, such as drug discovery, online commodity recommendations, and gene function analysis. This dissertation contributes to the current understanding of network structure by investigating various network estimation approaches using heterogeneous node interaction and activity data.

First, we propose a maximum *a posteriori* (MAP) estimation approach to reconstruct a target layer in a multilayer network. In this work, we have improved the SimHash algorithm to compute similarities between layers in multilayer networks. The layers that are most similar to the target layer are used to determine the parameters of the conjugate prior. This model allows us to predict missing links and direct experiments to identify nodes with similar functions. We evaluate the approach using two real-world multilayer networks, and the results demonstrate that the MAP estimation is effective in reconstructing the target layer, even when there are a large number of missing links.

Second, to capture the correlation between nodes, we establish a method for constructing a gene co-expression network for the *Anopheles gambiae* transcriptome from 257 unique studies obtained with different methodologies and experimental designs. We introduce the sliding threshold approach to select node pairs with high Pearson correlation coefficients. The resulting network, which we name AgGCN1.0, is robust to random removal of conditions and has similar characteristics to small-world and scale-free networks. Analysis of network sub-graphs revealed that the core is largely comprised of genes that encode components of the

mitochondrial respiratory chain and the ribosome, while different communities are enriched for genes involved in distinct biological processes.

Building upon the network construction approach, we introduce *GeCoNet-Tool*, a comprehensive gene co-expression network construction and analysis tool. The network construction part of the tool offers multiple options for processing gene co-expression data obtained using different technologies. The output is an edge list with the option to assign weights to each link. In the network analysis part, users can generate a table that includes various network properties such as communities, cores, and centrality measures.

Finally, to identify distance relationships between nodes, we propose an unsupervised learning framework that learns node vectors and constructs networks from noisy node activity data. The framework involves designing a scheme to generate random node sequences from node context sets, which are derived from raw node activity data. We utilize a three-layer neural network to train the node sequences and obtain node vectors, which enable us to construct networks and identify nodes with synergistic roles. Additionally, we propose an entropy-based approach to select the most relevant neighbors for each node in the resulting network. The effectiveness of our method is validated using both synthetic and real-world data.

To summarize, this dissertation focuses on developing computational models to extract network structures from heterogeneous data. The models are suitable for the analysis of both node interaction and activity data. The results of this study are expected to enhance our understanding of network structures and benefit the study of complex systems with rich but noisy data.

Table of Contents

List of Figures	xii
List of Tables	xv
Acknowledgements	xvi
Preface	xvii
1 Introduction	1
1.1 Background	1
1.2 Motivation	5
1.3 Research Questions and Contributions	6
1.4 Dissertation organization	9
2 A maximum <i>a posteriori</i> approach for multilayer network reconstruction and missing link prediction	11
2.1 Background	11
2.2 Layer reconstruction in multilayer networks	14
2.2.1 Maximum <i>a posteriori</i> based stochastic model	14
2.2.2 Similarity and layer comparison	17
2.2.3 Identify the parameters of the Gamma distribution prior	19
2.3 Experimental results and discussion	21
2.3.1 Numerical results on layer comparison	22
2.3.2 Validation of layer reconstruction and link prediction	23
2.4 Summary	27

3	A correlation coefficient approach to construct gene co-expression networks from heterogeneous data	29
3.1	Background	29
3.2	Model Formulation	32
3.2.1	Description of the gene expression data set	32
3.2.2	Expression data set pre-processing and normalization	33
3.2.3	Edge selection based on a sliding Pearson correlation coefficient threshold	36
3.2.4	Edge weight assignment	39
3.2.5	Network selection	41
3.3	Methodology verification and robustness test	42
3.3.1	Methodology verification	42
3.3.2	Robustness test	43
3.4	Analysis of the AgGCN1.0 largest connected component	44
3.4.1	Node centralities	45
3.4.2	Communities	48
3.4.3	Cores	50
3.5	Network architecture is based on biological function	53
3.5.1	The Network core is enriched in genes required for Oxidative Phosphorylation and Translation	53
3.5.2	Network communities are enriched for gene sets with functions in distinct biological processes	54
3.6	Summary	57
4	A software package for gene co-expression network construction and analysis . . .	60
4.1	Background	60
4.2	Implementation	60
4.2.1	Data processing and network generation	61
4.2.2	Network analysis	63

4.3	Results	64
4.4	Summary	65
5	Learning node features and network structure from noisy node activity data . . .	66
5.1	Background	66
5.2	Feature vectors from node activity data	69
5.2.1	Generate context sets from node activity data	70
5.2.2	Generate random node sequences	72
5.2.3	The skip-gram model	73
5.3	Network structure from trained node vectors	75
5.4	Results of Feature learning	77
5.4.1	Results on synthetic data	78
5.4.2	Results on real data	86
5.5	Results of network structure extraction	90
5.5.1	Compare the GTE and REM	90
5.5.2	Study the REM approach with AUC metrics on real data	94
5.6	Summary	95
6	Conclusion and Future work	97
6.1	Conclusion	97
6.2	Future Work	99
	Bibliography	101
A	Tensor factorization	125
B	Comparison between the MAP and MLE	127
B.1	Estimation based on the Pearson correlation	127
B.2	The choice of the number of bits	127
B.3	The number of similar layers on the estimation accuracy	129

C	Comparison between the MLE algorithm and entropy-based approaches	131
---	---	-----

List of Figures

1.1	A diagram of a multilayer network.	3
1.2	The diagram illustrates the procedure of constructing a network from node activity data.	4
2.1	Three layers of the FAO (a) network and HVR (b) network.	21
2.2	Layer comparison of the FAO network (a) and HVR network (b)	22
2.3	Eigenvector centrality-based SimHash algorithm	24
2.4	AUC vs. the dimension of node vectors	25
2.5	Comparison between the MLE and MAP methods on the FAO network . . .	26
2.6	Comparison between the MLE and MAP methods on the HVR network . . .	26
3.1	Mean and variance distributions of gene expression values in the data set . .	34
3.2	The distribution of the number of paired elements of all the gene pairs . . .	35
3.3	Comparison of the probability density function of PCCs with 10, 50, 150, and 250 paired elements	37
3.4	Top 0.5%, 1%, 2%, and 3% PCC points with respect to different numbers of paired elements	38
3.5	Edge weight rescaling	39
3.6	Means and medians of the PCCs above the threshold in each interval	40
3.7	The number of edges from the reshuffled data sets	42
3.8	The distribution of remaining edges in the network after removing 1%, 5% and 15% conditions	44
3.9	Node strength distribution of the network	45
3.10	Centrality measures for the AgGCN1.0 LCC	49

3.11	The 15 major communities of the LCC in AgGCN1.0	51
3.12	Coreness	52
3.13	The network core is enriched for genes that function in oxidative phosphorylation and translation	54
4.1	The <i>GeCoNet-Tool</i> user interface.	62
4.2	Examples of figure outputs from <i>GeCoNet-Tool</i> using <i>An. gambiae</i> expression data	65
5.1	The three-layer neural network model (The skip-gram model).	74
5.2	The node vectors trained from the first case study are visualized via PCA. .	80
5.3	The node vectors trained from the second case study are visualized via the PCA.	81
5.4	Visualization of node vectors trained from the first example with 20% missing values.	83
5.5	The visualization of node vectors trained from the Pearson correlation network.	86
5.6	The properties of the two real datasets	87
5.7	Visualization of the gene vectors via t-SNE, the genes are colored by the expression values	88
5.8	The visualization of node vectors trained from the PCC network	89
5.9	Experimental results when different thresholds are applied	91
5.10	Edge density analysis of the REM	91
5.11	The degree distributions of networks generated from the GTE and REM . .	93
5.12	The comparison of node degrees between densely and sparsely connected networks	93
B.1	The estimation results are based on the Pearson correlation of the weighted digest	128
B.2	Study the choice of the number of digits (based on the Hamming distance) .	128

B.3	The number of similar layers on the estimation results	129
-----	--	-----

List of Tables

3.1	Optimized parameters	39
3.2	Properties of the entire network with different sliding thresholds	41
3.3	statistical results of the original and removed networks	44
3.4	Topological properties of the co-expression network	47
3.5	Nodes distribution in each community and GO term enrichment	55
5.1	The synthetic dataset 1 (Data.1)	78
5.2	The synthetic dataset 2 (Data.2)	79
5.3	The synthetic dataset 3 (Data.3)	79
5.4	The synthetic dataset 4 (Data.4)	79
5.5	The relative distance between G_1 and G_2 and classification accuracy	81
5.6	The influence of missing values on the distance between G_1 and G_2 and classification accuracy	82
5.7	The relative distance between G_1 and G_2 and classification accuracy w.r.t. the variation of $\beta_w \in \{0.05, 0.1, 0.15, 0.2\}$	84
5.8	The relative distance between G_1 and G_2 and the classification accuracy of the PCC networks	85
5.9	The relative distance between G_1 and G_2 w.r.t. ρ	86
5.10	The prediction accuracy (%) w.r.t. ρ	87
5.11	The properties of networks constructed with the GTE and REM	92

Acknowledgments

First and foremost, I would like to express my sincere gratitude to Dr. Caterina Scoglio for her continuous guidance, encouragement, and patience in helping me accomplish my research projects. Dr. Scoglio's advice and wisdom in research have helped me navigate through numerous challenges during my research. Her advice will be a treasure for my future research and career development.

I want to express my appreciation to Dr. Kristin Michel for her generous funding support and valuable advice. With her help, I have had the opportunity to start my career in the field of bioinformatics. I am also grateful for Dr. Hongyu Wu and Dr. Don Grunbacher. This work would not have been accomplished without their insightful comments and encouragement.

I would also like to take this opportunity to acknowledge my friends, Jacob Friesen, Chen Li, Qihui Yang, Chunlin Yi, Bo Liu and all other friends for their help and friendship.

Preface

This dissertation, *Network construction and feature learning from noisy data*, is submitted for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering at Kansas State University. The research has been performed under the supervision of Prof. Caterina Scoglio.

To the best of my knowledge, this work is original, except where acknowledgements and references are indicated. Most of the work has been presented in the following published or submitted peer-reviewed journals:

1. Kuang, Junyao, and Caterina Scoglio. "Layer reconstruction and missing link prediction of a multilayer network with maximum a posteriori estimation." *Physical Review E* 104, no. 2 (2021): 024301.
2. Kuang, Junyao, Nicolas Buchon, Kristin Michel, and Caterina Scoglio. "A global *Anophelesgambiae* gene co-expression network constructed from hundreds of experimental conditions with missing values." *BMC Bioinformatics* 23, no. 1 (2022).
3. Kuang, Junyao, Caterina Scoglio, and Kristin Michel. "Feature learning and network structure from noisy node activity data." *Physical Review E* 106, no. 6 (2022): 064301.

Chapter 1

Introduction

1.1 Background

Networks are a way to view the interactions between various components of a complex system and analyze their relationships, as described in various works such as^{1;2}. These interactions can be analyzed through techniques like centrality analysis and community partition to identify key players in the system and understand the relationships between different parts³. In the context of gene co-expression networks, researchers can use centrality measures and community analysis to identify synergistic genes and understand their role in biological processes⁴. By simulating the dynamics of networks, researchers can gain insights into the states and behavior of a system^{5;6}. For example, epidemiologists can use epidemic models to analyze the spread of infectious diseases and make predictions, as seen with the COVID-19 pandemic⁷. At the beginning of 2020, researchers had predicted the spread of COVID-19 using susceptible-infected-susceptible models and warned policymakers and the public to be prepared and reduce international travel^{8–10}.

Prior to conducting any analysis on networks, researchers must first accurately estimate network structures from various types of data (measurements)^{3;11}, including node-to-node interactions and properties that have been measured at the individual node level. As a result, the task of constructing a network can be divided into two categories based on the

data types involved. These categories include (1) the estimation of network structures based on pairwise interactions between nodes, and (2) the estimation of network structures from node activity data.

Most empirical networks can be obtained by measuring the direct interactions between each pair of nodes in a complex system^{3;12}. For example, in a web graph, a directed link is established if there is a hyperlink between two web pages^{13;14}. However, direct interactions between nodes may not always be present or uniform, and the various measurements provide different perspectives on the interactions or individual nodes, resulting in a more accurate and nuanced representation of network structure than any single measurement could alone³. For example, the interactions among people are influenced by numerous factors, including frequently visited venues, occupations, ages, shared hobbies, etc.³. As a result, the same group of people can be represented in different social networks, such as face-to-face, co-working, co-authorship, online friendship networks, etc.^{3;15–18}. Researchers often assemble a multilayer network, where each layer represents a specific type of interaction (as shown in Fig. 1.1). However, collecting the entire multilayer network can be expensive and time-consuming, leading to the need for efficient algorithms to predict edges in the layer of interest from easily measurable layers, also known as estimating a target network for a multilayer network.

On the other hand, measuring the connections between every pair of nodes may not always be feasible^{2;3;12}. Instead, various experiments may be conducted to measure the properties of individual nodes^{19;20}. In this scenario, the produced data is abundant and noisy, representing the properties of individual nodes rather than the interactions between pairs of nodes. The challenge of the objective is how to accurately infer the structure of the underlying network from such noisy node activity data²¹. This problem is common in constructing brain networks, financial networks, ecological networks, biological networks, and climate networks^{19;20;22–31}. The data collected in these areas is often distributed in different ranges, may contain missing or repeated observations, and is often presented in different forms. Some data could be unreliable and error-prone³, making it hard to detect nodes with similar properties and attribute them to specific functions. Therefore, accurately

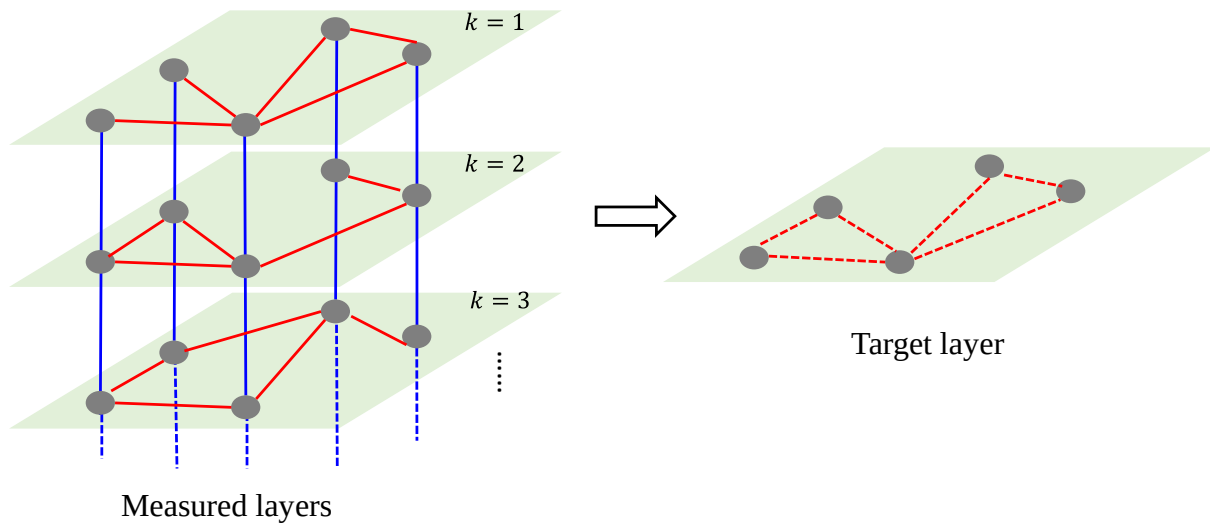


Figure 1.1: A diagram of multilayer network. A multilayer network consists of a set of nodes that are common across all layers, where each layer represents a unique type of interaction. The edges in the measured layers can be used to infer the edges in the target layer (layer of interest).

learning node features and estimating network structures from noisy node activity data is crucial for understanding networked systems (as illustrated in Fig. 1.2). The node features are d -dimensional vectors learned from node activity data, allowing researchers to identify nodes with similar functions.

Gene co-expression data often suffer from unreliability and errors, posing challenges for researchers studying biological activities. Next Generation Sequencing (NGS) technology has enabled the collection of large amounts of gene expression data in various experiments, allowing biologists, entomologists, and epidemiologists to study the expression levels of genes in different conditions or parts of the body³¹⁻³⁴. In the studies of *Anopheles gambiae*, entomologists conduct different types of experiments and use microarray and NGS technologies to observe the expression levels of genes^{2;35-66}. However, these noisy measurements of independent genes can make it difficult to identify their interactions and co-regulation in biological activities. Gene co-expression networks offer a way to analyze gene functions by exploring the relationships between genes in a given tissue sample^{31;34}. Despite the challenges posed by unreliable data, researchers are actively investigating methods to reduce their impact on network construction⁶⁷⁻⁷¹. Accurate extraction of network structures from noisy experi-

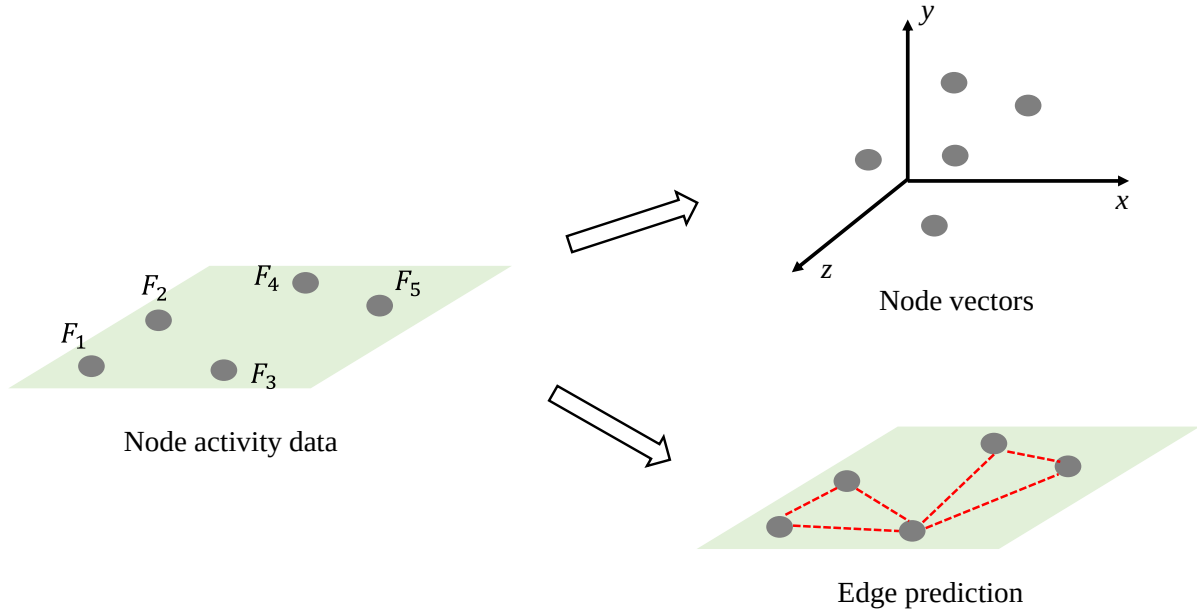


Figure 1.2: The diagram illustrates the procedure of constructing a network from node activity data. Each node i ($1 \leq i \leq 5$) is represented by its node activity data, F_i , a vector composed of measurements obtained from various experimental conditions using different technologies. It is noteworthy that F_i may contain missing values and disproportionate amounts of noise. The ultimate objective is to transform the node activity data into high-dimensional node vectors, enabling the extraction of network structures.

mental data is crucial for a better understanding of complex systems and the relationships between their different components⁷².

The objective of this dissertation is to make contributions to the field of network structure estimation by proposing a novel methodology for estimating the structure of a target layer in multilayer networks. The work also extends the application of Pearson correlation to the analysis of diverse gene expression data, explores the use of neural networks for node feature learning, and introduces the Rényi entropy as a tool for extracting network structure from node features. The proposed objectives are expected to advance the understanding and techniques in network structure estimation and contribute to the development of new approaches in this research area.

1.2 Motivation

In the past years, it has been demonstrated that networks are powerful tools for analyzing relationships between elements within complex systems. For instance, networks have been successfully applied in a variety of fields, such as identifying critical papers in citation networks, individuals in social networks, and web pages in web graphs. While networks are powerful tools, it is important to note that obtaining network structures directly is not always possible. Many scientific experiments yield noisy measurements of elements in various forms, which can complicate the process of constructing accurate networks.

Empirical experiments often measure pairwise node interactions from various angles. For example, in the study of the social network of university students, the measurements are the physical proximity of students in different circumstances^{3;73}. While these measurements provide valuable insights into the pairwise interactions between students, they may convey different types of information. To fully capture the complexity of such systems, we can construct a multilayer network from these rich observational data. However, it can be challenging to obtain a specific layer of interest in real-world studies. In such cases, accurately estimating the target layer from a multilayer network can aid our understanding of networked systems.

At the same time, some empirical observations provide measurements of individual node properties. Analyzing relationships between nodes often requires estimating a network structure from noisy node activity data. One such example is the study of gene expression data, where high-throughput technologies such as microarray and RNA-seq enable simultaneous measurement of expression levels for thousands of genes⁷⁴. Constructing and analyzing gene expression networks provides a systemic view of gene expression relationships, illuminating connections between gene pairs. However, the large amounts of data produced by these technologies are often susceptible to measurement errors due to various factors, including temperature, humidity, and experimenter bias. To study relationships between nodes, an accurate estimation of the network structure is necessary.

The main objectives of this research are to develop scalable techniques for estimating

network structures and analyzing node features from large, diverse, and noisy data. The focus of this dissertation will be four-fold. First, it will explore the estimation of network structure from node interaction data by developing a maximum *a posteriori* estimation approach to reconstruct a specific layer in a multilayer network. Second, the research will focus on developing methods to construct a network from noisy *Anopheles gambiae* gene co-expression data, where edges represent correlations. The third objective of the work is to develop a user-friendly toolbox for correlation network construction and analysis. This will make the techniques developed in this research accessible to a broader audience and facilitate their adoption in various fields. Finally, this dissertation explores whether neural networks can be used to learn node features and extract network structure from noisy node activity data. The proposed techniques in this dissertation are expected to be versatile and can be extended to diverse fields, such as constructing networks from brain, financial, ecological, and climate data. Overall, the outcomes of this dissertation will provide valuable insights into the analysis of large and heterogeneous data sets^{19;20;22-31}.

1.3 Research Questions and Contributions

The first research question addresses the prediction of missing links in a network of interest, which is considered as the target layer of a multilayer network. This study aims to solve the problem of estimating network structure from node pairwise interaction data and further improve the estimation accuracy of the target layer. The expected outcomes are advancements in the methodology of layer reconstruction, specifically in reconstructing a target layer of multilayer network.

The second research question focuses on the challenge of handling missing values when constructing correlation networks from gene expression data. Conventional approaches for network construction have limitations in addressing missing values, which can compromise the accuracy of the constructed network. To overcome this challenge, this study aims to develop new data processing techniques to standardize heterogeneous gene expression data, rendering it more suitable for processing and analysis. The expected outcomes of this re-

search include new techniques for processing and analyzing gene expression data, which can provide more accurate estimates of gene co-expression networks. These techniques will improve our understanding of biological systems and have practical applications in fields such as immune system analysis and new drug discovery.

Based on the developed correlation network estimation and analysis scheme, we expect to create an integrated software package for analyzing gene expression data. This package will allow users to construct gene co-expression networks with the desired density and provide various tools for network analysis. We aim to provide a user-friendly interface through which users can customize the properties of their gene co-expression network and tailor their network analyses to their specific research needs. The expected outcome of this work is a toolbox that empowers users to efficiently analyze gene expression data, enabling them to uncover new insights into biological systems.

The fourth research question focuses on exploring the potential of neural network models in learning node representations from raw gene expression data. This work will investigate the use of neural network-based techniques for learning node embeddings that reveal the distance relationship between the nodes. The outcomes are expected to provide new insights into learning node features from raw data rather than from a previously constructed network. Analytical results will demonstrate the usefulness of node embeddings in understanding and recovering nodes with similar properties.

The final research question of this dissertation focuses on the extraction of a meaningful network structure from node embeddings obtained in the previous research question. The objective is to develop a scalable method to preserve nodes' roles, such as hubs and leaf nodes, and minimize isolated nodes' occurrence. The expected outcomes of this study will contribute to the understanding and improvement of network structure extraction from node embeddings. The developed method will also be applicable to a wide range of network types, including social networks, biological networks, and financial networks.

The major contributions of the dissertation are as follows:

1. Develop a maximum *a posteriori* estimation-based model for reconstructing a specific

layer in a multilayer network. The SimHash algorithm is employed to calculate the similarities between different layers. Layers with similar structures are used to determine the parameters of the conjugate prior, which enables us to make accurate predictions for the target layer. The developed model is capable of predicting missing links in gene co-expression networks and guiding experiments to find potential links⁷⁵.

2. Design a robust approach to construct gene co-expression networks for *Anopheles gambiae* using the Pearson correlation method. To handle the noisy and heterogeneous data, we introduce a sliding threshold approach to select edges for the network, which is more effective in handling missing values and random removal of conditions compared to conventional approaches. The robustness of our approach is tested and confirmed through various simulation experiments. Additionally, we perform gene ontology (GO) enrichment analysis on the resulting network, which confirms the network structure and shows that genes with similar biological functions tend to cluster together in the network⁷⁴. Overall, this work provides a reliable framework for analyzing gene expression data and uncovering the underlying biological relationships among genes.
3. Develop an executable software package *GeCoNet-tool* for analyzing raw gene expression data. The package integrates correlation network construction and analysis, providing users with the ability to construct gene co-expression networks with desired density and analyze the network with various tools. The user-friendly interface allows users to customize the properties of their gene co-expression network, including selecting the correlation method, setting edge thresholds, and performing network analysis, such as centrality analysis and community detection. The package also includes visualization tools to facilitate the interpretation of the network results.
4. Design a scalable and novel neural network-based approach for learning node representations from heterogeneous and noisy data. Random node sequences are generated for each condition to avoid re-scaling and normalizing data from various technologies. The node sequences are trained using natural language processing models to

obtain node vectors of a consistent dimension, capturing the features of the nodes. The dimension-reduced node vectors reveal the similarity between nodes, with those having similar properties mapped together. The network structure is constructed by selecting neighbors for each node based on similarities computed from the embedded node vectors, allowing for the inclusion of all genes in the network. The method shows promise in learning useful node representations directly from raw data, without relying on pre-constructed networks⁷⁶.

5. Design a novel approach based on entropy to extract edges in an underlying network structure. This approach is based on the concept of Rényi entropy and utilizes the distribution of similarity values to select edges. The proposed methodology is capable of constructing networks without isolating nodes while preserving the roles of nodes, such as hubs and leaf nodes. This innovative method provides a more comprehensive understanding of network structure and has the potential to improve the accuracy of network reconstruction⁷⁶.

1.4 Dissertation organization

The dissertation is structured as follows: Chapter 1 provides an introduction to the background, motivation, research questions, and main contributions of the study. Chapter 2 presents a maximum *a posteriori* estimation-based model for reconstructing a specific layer in a multilayer network. In Chapter 3, we describe the methodology for constructing gene co-expression correlation networks from heterogeneous and noisy *Anopheles gambiae* gene expression data, and evaluate the effectiveness of the network structure in uncovering biological activities. Chapter 4 details the development of an executable software package for network construction and analysis from raw gene expression data. In Chapter 5, we propose a novel neural network-based approach for learning node features and introduce an entropy-based method for extracting network structure from heterogeneous and noisy data. Finally, Chapter 6 concludes by summarizing the main contributions of the dissertation, analyzing

the limitations, and suggesting directions for future work.

Chapter 2

A maximum *a posteriori* approach for multilayer network reconstruction and missing link prediction ¹

2.1 Background

Network science has a wide range of applications in different fields, including the analysis of information diffusion, infectious disease spread, and gene co-expression. By studying the relationships between nodes in a network, researchers can analyze the system’s robustness and predict future outcomes^{77;78}. For example, epidemiologists can use network analysis to predict the spread of COVID-19 and provide policymakers with early intervention strategies to curb its spread⁷⁹. In studying gene co-expression data, one can construct gene co-expression networks to discover crucial genes (nodes) by choosing genes with high degree centralities, closeness centralities, or eigenvector centralities. Recently, multilayer networks have become increasingly popular in biological research, where different layers represent various types of molecular interactions^{80;81}. However, obtaining a complete multilayer network through biological experiments can be expensive and time-consuming. In many cases, a layer of interest

¹This chapter is a slightly modified version of a published article⁷⁵.

may be incomplete with missing links. To address this, researchers aim to reconstruct the target layer using information from the other layers. This can be done by estimating the target layer based on the existing layers. In other words, we can estimate the target layer through existing layers^{21;80;82-84}. After constructing the target layer, researchers can devote restricted resources to detecting high-probability edges.

Various methods have been proposed for reconstructing networks and predicting missing links, with many of them based on generative models^{3;5;21;80;82;85-96}. Generative models reconstruct the network topology by fitting a stochastic network model^{21;87;88;97}, using a maximum-likelihood estimation (MLE) algorithm to determine the optimal parameters that describe the network. For example, the authors of^{87;88} presented a degree-correlated stochastic block model to reconstruct a single-layer network, where edges are computed through the tensor product of node vectors, with node vectors' entries being the nodes' degrees in each community. In²¹, the authors extended the single-layer stochastic block model to multilayer networks, using it to predict missing links and detect overlapping communities. They validated the algorithm on two real multilayer networks, hiding 20% of links and non-links. However, when there are many layers, trying all possible layer combinations to find the interdependent layers can be burdensome.

In multilayer networks, there could be structurally similar layers. Therefore, it is possible to take advantage of similar layers to help reconstruct the target layer. We propose comparing the target layer with the remaining layers if the target layer is partially known. In the literature, multiple methods have been proposed to compare networks⁹⁸⁻¹⁰². For instance, the DeltaCon method introduced in^{99;100} compares the affinity scores of every node pair in two networks. However, this method is sensitive to changes in the number of edges, and removing edges leads to a significant change in the distance. Other methods for network comparison, such as vertex/edge overlapping, vertex/edge vector similarity, and the SimHash algorithm, are reviewed and compared in papers^{101;102}. The vertex/edge overlapping method applies the rule that two graphs are similar if they share many vertices and edges, but it may not be sensitive to changes in high-quality vertices, topology, and other properties of networks. The vertex/edge vector similarity method compares the node/edge weight vectors

of two networks but may not be sensitive to changes in topology and other properties. To take advantage of the features of networks, the SimHash algorithm is introduced to compare networks. The PageRank¹⁰³ together with edges are used as network features in the SimHash algorithm to compare web page networks.

This chapter proposes a MAP-based method for target layer reconstruction and link prediction, which aims to overcome the limitations of the MLE algorithm and entropy-related approaches that rely heavily on known information of the target layer. The MAP algorithm considers layers that are structurally similar to the target layer to compute the parameters of the conjugate prior, resulting in more consistent results than the MLE method. The first contribution of this paper is to discover an incomplete target layer by computing its edges through the inner product of node vectors. The optimal entries of node vectors are obtained by maximizing the posterior probabilities of the stochastic model. We demonstrate that increasing the dimension of node vectors can improve model accuracy, but there is a diminishing improvement for large vector dimensions. Another contribution is that we introduce the eigenvector centrality-based SimHash algorithm to detect structurally similar layers (interdependent layers). The eigenvector centralities of nodes are extracted as network features, which allow us to recover the structure of networks, as shown in the experimental results. The proposed method assumes that the number of edges between any pair of nodes follows a Poisson distribution^{21;87}, and the Gamma distribution is the conjugate prior to the Poisson distribution¹⁰⁴. We compute the parameters of the conjugate prior through the adjacency matrices of similar layers, weighting the contributions of the similar layers by their similarities. The number of edges between each pair of nodes is calculated as the inner product of the node vectors. In our experiments, we show that similar layers are critical in improving the robustness of link predictions.

The chapter is organized as follows. In section two, we introduce the MAP method for target layer reconstruction and link prediction. We then propose the eigenvector centrality-based SimHash algorithm for identifying structurally similar layers, followed by a discussion on the process for determining the parameters of the conjugate prior under different conditions. In section three, we evaluate the performance of the eigenvector centrality-based

SimHash algorithm on two real-world multilayer networks. Next, we evaluate the MAP algorithm-based target layer reconstruction on these networks and compare the results with those obtained using the MLE algorithm. Finally, we summarize the chapter in section four.

2.2 Layer reconstruction in multilayer networks

2.2.1 Maximum *a posteriori* based stochastic model

In this section, we define the stochastic model for both directed and undirected multilayer networks. The adjacency matrix of the target layer is denoted by A . The reconstruction goal is to estimate A with partial knowledge of the target layer and other layers in the multilayer network. To reconstruct the target layer, a set of parameters is required to describe the model, which we denote as θ . Based on Bayes' theorem, the posterior probability of θ is given by:

$$P(\theta | A) = \frac{P(A | \theta)P(\theta)}{P(A)}, \quad (2.1)$$

where $P(\theta | A)$ is the posterior probability of θ , $P(A | \theta)$ is the likelihood of A under θ , $P(\theta)$ is the prior probability of θ and $P(A)$ is the marginal likelihood that contains all the information of the network. Since $P(A)$ is a constant, $P(\theta | A)$ is proportional to the product of $P(A | \theta)$ and $P(\theta)$. Therefore, we have

$$P(\theta | A) \propto P(A | \theta)P(\theta). \quad (2.2)$$

For any pair of nodes in the network, we use E_{ij} to denote the expected number of links (which could be fractional) between node i and node j . In contrast to unweighted networks where the entries of the adjacency matrix are denoted by 0 or 1, the entries E_{ij} are real numbers, thus we can interpret network A as a weighted network.

Before substituting any parameters into expression (2.2), we make the following assumptions. First, we assume that the links in the target layer are independent and iden-

tically distributed, meaning that the number of edges between node i and node j does not affect the relationship between node i and node k . Additionally, we assume that the number of links between any pair of nodes is extracted from a Poisson distribution, i.e., $P(A_{ij} | E_{ij}) = \frac{e^{-E_{ij}} (E_{ij})^{A_{ij}}}{A_{ij}!}$. We can rewrite expression (2.2) after substituting E_{ij} and A_{ij} as

$$P(\theta | A) \propto \prod_{i,j} \frac{e^{-E_{ij}} (E_{ij})^{A_{ij}}}{A_{ij}!} P(E_{ij}). \quad (2.3)$$

The prior probability $P(E_{ij})$ can be neglected in the MLE algorithm since it is a constant. In the MAP algorithm, we must specify the prior distribution of $P(E_{ij})$. The conjugate prior distribution for the Poisson distribution is the Gamma distribution:

$$\begin{aligned} P(E_{ij}) &= \frac{\beta_{ij}^{\alpha_{ij}}}{\Gamma(\alpha_{ij})} E_{ij}^{\alpha_{ij}-1} e^{-\beta_{ij} E_{ij}} \\ &\propto E_{ij}^{\alpha_{ij}-1} e^{-\beta_{ij} E_{ij}}, \end{aligned} \quad (2.4)$$

where α_{ij} , β_{ij} and $\Gamma(\alpha_{ij})$ are the shape parameter, the scale parameter, and the Gamma function of the Gamma distribution, respectively. Section 2.2.3 introduces a procedure to determine α_{ij} and β_{ij} through the layers with high similarities. Substituting the conjugate Gamma distribution into expression (2.3), we have

$$P(\theta | A) \propto \prod_{i,j} e^{-(\beta+1)E_{ij}} (E_{ij})^{A_{ij}+\alpha-1}. \quad (2.5)$$

Note that we have left out constant terms.

The problem now has been simplified to finding the parameters E_{ij} that can maximize the posterior probability. However, an expression for E_{ij} has not been specified. In this work, we represent the nodes in the target layer as vectors, and compute the links between nodes using the inner product of their corresponding vectors. The expected number of links

E_{ij} can be computed as follows:

$$E_{ij} = \sum_{z=1}^K s_{iz} t_{jz}, \quad (2.6)$$

where s_{iz} and t_{jz} are respectively the z th entry of node i 's vector and node j 's vector. Here, we use s and t to denote source and target nodes. K is the dimension of the vector. Some MLE-related works use tensor factorization to decompose the links, and the tensor dimension is interpreted as the number of overlapping communities. As a result, there should be an optimal number of communities that can maximize the estimation accuracy. However, this cannot be rigorously achieved and the tensor factorization can be simplified to the dot product according to our analysis in appendix A.

Expression (2.5) is still intractable after substituting E_{ij} with equation (2.6). We take the logarithmic form of expression (2.5), which gives

$$\begin{aligned} L(\theta \mid A) &= \sum_{i,j} [(A_{ij} + \alpha_{ij} - 1) \log E_{ij} - (\beta_{ij} + 1) E_{ij}] \\ &= \sum_{i,j} [(A_{ij} + \alpha_{ij} - 1) \log \sum_{z=1}^K s_{iz} t_{jz} \\ &\quad - (\beta_{ij} + 1) \sum_{z=1}^K s_{iz} t_{jz}], \end{aligned} \quad (2.7)$$

where $L(\theta \mid A)$ is the log posterior.

To find the maximized posterior for expression (2.7), we apply the Jensen's inequality $\log \bar{x} \geq \overline{\log x}$ ³, which gives

$$\begin{aligned} \log \sum_{z=1}^K s_{iz} t_{jz} &= \log \sum_{z=1}^K q_{ijz} \frac{s_{iz} t_{jz}}{q_{ijz}} \\ &\geq \sum_{z=1}^K q_{ijz} \log \frac{s_{iz} t_{jz}}{q_{ijz}} \\ &= \sum_{z=1}^K q_{ijz} (\log s_{iz} t_{jz} - \log q_{ijz}). \end{aligned} \quad (2.8)$$

Equality is satisfied when

$$q_{ijz} = \frac{s_{iz}t_{jz}}{\sum_z s_{iz}t_{jz}}. \quad (2.9)$$

After substituting Eqs. (2.8) and (2.9) into Eq. (2.7), Eq. (2.7) can be simplified to

$$\begin{aligned} L(\theta \mid D) = \sum_{i,j,z} [(A_{ij} + \alpha_{ij} - 1)q_{ijz} \log s_{iz}t_{jz} \\ - (\beta_{ij} + 1)s_{iz}t_{jz}]. \end{aligned} \quad (2.10)$$

Taking the derivative of Eq. (2.10) with respect to s_{iz} and t_{jz} and equating to zero, we obtain the optimal node vectors that maximize the posterior probability:

$$s_{iz} = \frac{\sum_j (A_{ij} + \alpha_{ij} - 1)q_{ijz}}{\sum_j (\beta_{ij} + 1)t_{jz}}, \quad (2.11)$$

$$t_{jz} = \frac{\sum_i (A_{ij} + \alpha_{ij} - 1)q_{ijz}}{\sum_i (\beta_{ij} + 1)s_{iz}}. \quad (2.12)$$

The procedure to obtain the optimized values of s_{iz} and t_{jz} is to first assign random initial values for s_{iz} and t_{jz} . Then, we iteratively update Eqs. (2.9), (2.11), and (2.12) until Eq. (2.7) converges. However, before applying the above iteration, we need to identify the values of α_{ij} and β_{ij} , which are introduced in Section 2.2.2 and Section 2.2.3.

2.2.2 Similarity and layer comparison

In section 2.2.1, we detailed the procedure for computing the optimal node vectors by maximizing the posterior probability. To perform the posterior probability maximization, we require the parameters α_{ij} and β_{ij} for the prior distribution. We propose to compute α_{ij} and β_{ij} through the layers of the multilayer network that are structurally similar to the target layer. However, it is important to note that there are missing links in the target layer, and

the percentage of missing links is not known. Therefore, an effective network-comparing method must not be significantly affected by the percentage of missing links.

There are several types of centralities that can be used to characterize networks, including degree centrality, eigenvector centrality, and closeness centrality. Degree centrality quantifies the importance of a node by counting the number of links it has. At the same time, the eigenvector centrality can be regarded as an extension of degree centrality in which a node's importance is also affected by its neighbors' importance. Closeness centrality measures the average distance between a node and all other nodes in the network. By using one or more of these centralities, we can capture important features of a network. To compare the target layer with other layers in the multilayer network, we can compare the features of the layers using centralities. For our purposes, we select eigenvector centrality as the network feature, which is then used in the SimHash algorithm to compute similarities.

The SimHash algorithm works as follows^{101;102}. The feature of a network can be represented as a set of token-value pairs $\{(v_i : w_i)\}$, where v_i is a node and w_i is its measure under the feature, such as eigenvector centrality. It is worth noting that both the target layer and the layer to be compared have the same set of tokens. In cryptography, any message can be encrypted to a unique binary number, known as a digest. Similarly, we can represent each token with a unique binary number with ϕ bits ($2^\phi > \text{the number of } v_i$). For each binary number (digest), we map every 1 to w_i , and 0 to $-w_i$. Thus, each token is mapped to a weighted digest with ϕ digits. To obtain the weighted digest of the entire network, we sum up all the weighted digests of the tokens. It is important to note that there is no carry in the summation.

To measure the similarities between the target layer and the other layers, we can compare the digests of the layers. A simple way to compare the digests is to convert the weighted digests to binary digests. The binary digest of a network can be obtained by setting positive digits to 1 and negative digits to 0. The similarity between the target layer m and any other

layer r can be measured by

$$\mu_{m,r} = 1 - \frac{\text{Hamming}(H_m^d, H_r^d)}{\phi}, \quad (2.13)$$

where $\mu_{m,r}$ is the similarity between layer m and r , H_m^d is the binary digest of the target layer, and H_r^d is the binary digest of the layer to be compared, respectively. The similarity score ranges from 0 to 1, with 1 indicating the two layers are identical and 0 indicating they are completely dissimilar. By comparing the similarity scores of the target layer with all other layers, we can identify the most similar layers.

An alternative way to obtain the similarities is by computing the Pearson correlation coefficient of the weighted digests. The estimation results based on the Pearson correlation are shown in Appendix. B. In this work, we measure the similarity between the target layer and the other layers through binary-based digests. The influence of the bit number is discussed in Appendix. B.

2.2.3 Identify the parameters of the Gamma distribution prior

In this section, we introduce the procedure to determine the parameters α_{ij} and β_{ij} of the conjugate prior. We discuss this problem in two cases.

In the first scenario, we assume that the structure of the target layer is partially known, meaning that some entries of the adjacency matrix are missing or unknown. In this case, we can use the layer comparison method presented earlier to compare the target layer with other layers in the multilayer network. The top L' layers with the highest similarities can be selected to estimate the parameters of the conjugate priors. The parameters of the Gamma distribution prior are computed as

$$\begin{cases} \alpha_{ij} = \max(\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r, 1) \\ \beta_{ij} = \max(\sum_{r=1}^{L'} \mu_{m,r}, \frac{\sum_{r=1}^{L'} \mu_{m,r}}{\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r}), \end{cases} \quad (2.14)$$

where m denotes the target layer, $\mu_{m,r}$ is the similarity between the target layer m and any

layer r . A_{ij}^r is the adjacency matrix of layer r in L' . In equations (2.11) and (2.12), since A_{ij} could be zero, if α_{ij} is less than one, we obtain a negative s_{ij} . Thus, we need to limit the range of α_{ij} . If $\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r \in (0, 1)$, we set $\alpha_{ij} = 1$, and set β_{ij} to $\frac{\sum_{r=1}^{L'} \mu_{m,r}}{\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r}$ to maintain the means of the Gamma distribution prior unchanged. If $\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r = 0$, we will set $\alpha_{ij} = 1$, and set β_{ij} equal to a large number to ensure the MAP algorithm converges.

When the structure of the target layer is completely unknown, a heuristic approach can be used to determine the parameters of the conjugate prior. This approach is based on identifying other functionally similar layers, which can be obtained from additional published networks or data. These layers can be combined with the target layer to form a new multi-layer network, and the proposed MAP algorithm can be applied to compute the parameters of the conjugate prior. In this case, the similarities between the target layer and the functional layers cannot be determined through any network comparison algorithm. Therefore, it is assumed that the similarities between these layers are all one. We assign the parameters of the Gamma distribution as

$$\begin{cases} \alpha_{ij} = \sum_{r=1}^{L'} A_{ij}^r \\ \beta_{ij} = L'. \end{cases} \quad (2.15)$$

Similarly, if $\sum_{r=1}^L A_{ij}^r < 1$, we will set $\alpha_{ij} = 1$, and set β_{ij} to a large number to make the MAP algorithm converge.

In this scenario, we have no structural information about the target layer, so setting $A_{ij} = 0$ in equations (2.11) and (2.12) would be equivalent to assuming that all edges in the target layer are absent. To address this issue, we can instead set the entries of A_{ij} as the ratio of α_{ij} and β_{ij} , i.e., $A_{ij} = \alpha_{ij} / \beta_{ij}$. We assign these entries as the average presence over the other layers to avoid the assumption that the target layer has no edges.

2.3 Experimental results and discussion

In this section, we present the evaluation of the method proposed in section 2.2. Firstly, we apply the SimHash algorithm to two real networks where links are removed uniformly at random at different percentages. Secondly, we evaluate the effectiveness of the MAP algorithm for the two real multilayer networks and compare it to the MLE algorithm.

The first real network we use to evaluate our proposed MAP algorithm is the FAO (Food and Agriculture Organization) trade network¹⁰⁵. The FAO multilayer network is composed of 364 layers, each representing product trading among 214 countries. A link is detected between two nodes in a layer if there is trading of the corresponding product between the two countries. We show three layers of the FAO network in Fig. 2.1 through the KiNG software¹⁰⁶. As the layers in the FAO network are not ordered in any specific way, we evaluate the method by assuming the target layer to be one of the first nine layers of the FAO network. Note that we use all 363 remaining layers in our experiments for identifying similar layers.

The second real network (HVR network) we use to evaluate the proposed MAP algorithm has nine layers and 307 nodes¹⁰⁷, representing malaria parasite genes. The nine layers correspond to nine highly variable regions on the genes themselves. An edge is detected if two genes share an exact match of significant length in the highly variable region. Fig. 2.1 shows three layers of the HVR network.

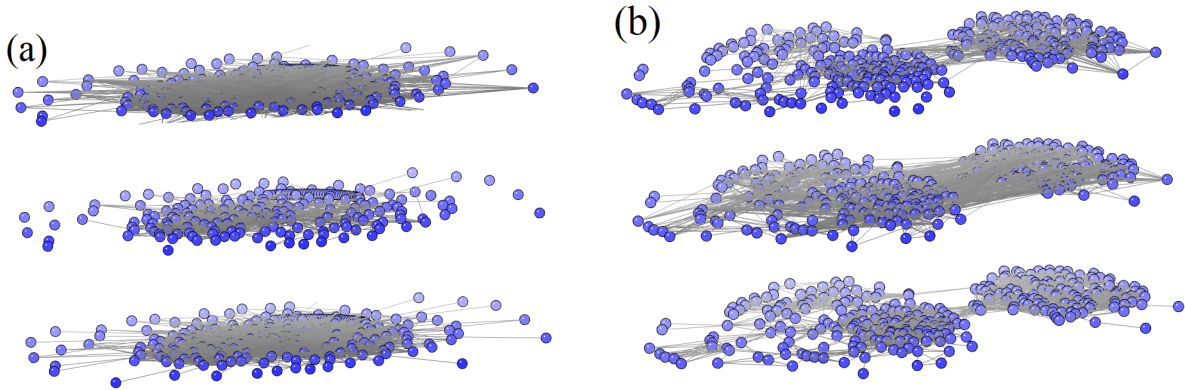


Figure 2.1: *Three layers of the FAO (a) network and HVR (b) network. Nodes in different layers share the same plane coordinates.*

2.3.1 Numerical results on layer comparison

The SimHash algorithm was evaluated on both the FAO and HVR networks by obtaining the binary digest of each layer based on the eigenvector centrality of the nodes. The similarity between any two layers was computed using Eq. (2.13). The results are shown in Fig. 2.2, where each layer is set as the target layer, and the similarities between the target layer and all the other layers are shown in interquartile ranges (IQR).

For the FAO network, only the first nine layers were used as target layers, and each IQR bar contains 363 similarity values. The similarities obtained through the SimHash algorithm are between 0 and 1 where a similarity of 0 means the two layers have completely opposite eigenvector centrality distributions, a similarity of 0.5 means the two layers are independent, and a similarity approaching 1 means the two layers are similar. In the FAO network, the similarities range between 0.5 and 1, indicating that there are no layers with completely opposite eigenvector centrality distributions. In contrast, in the HVR network, most layers are independent, as the similarities are all less than 0.8, except for layers 7 and 9.

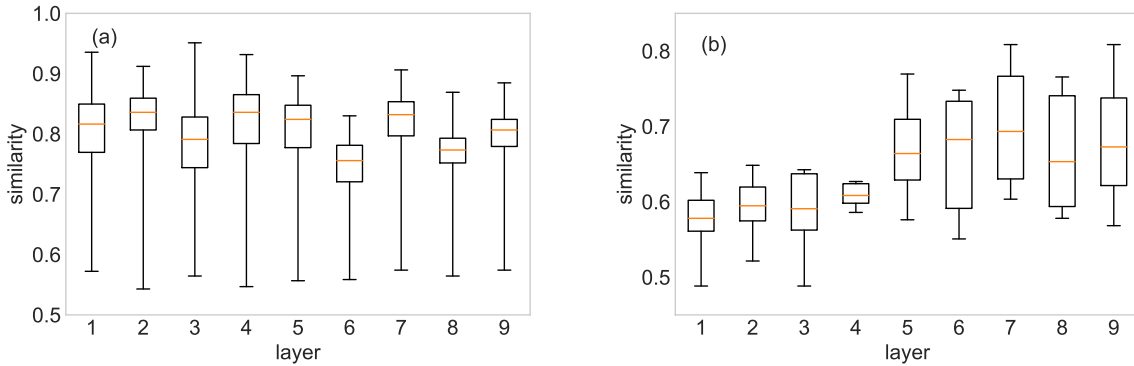


Figure 2.2: *Layer comparison of the FAO network (a) and HVR network (b). (a) shows the similarities between the target layer and all the other 363 layers in the FAO network. (b) shows the similarities between the target layer and the other eight layers in the HVR network. Note that each similarity value is averaged over 10 runs.*

The network-comparing method should be able to identify structurally similar layers even in the presence of missing links. To this end, we randomly removed 20%, 40%, 60% and 80% of edges from the target layer to generate incomplete networks and compared them with the target network. In Fig. 2.3(a), each layer of the FAO network (first nine layers) and HVR

network is set as the target network, respectively, and we randomly remove 20%, 40%, 60% and 80% of edges from the target layer. The similarities between the incomplete networks with removed edges and the target layer are computed. From the top panels, we can observe that even with 80% of edges randomly removed, the similarities are still greater than 0.8.

In the second experiment, we demonstrate that missing links in the target layers have little impact on the similarity between the target layer and its similar layers. We select the five most similar layers for each target layer from all the other layers and then randomly remove 20%, 40%, 60% and 80% of edges from the target layer to generate incomplete networks. The incomplete networks are compared to the five similar layers, and we can obtain five similarities for different removal percentages. In the bottom panels of Fig. 2.3, we show the average of the five similarities. The results for the first nine layers of the FAO network are shown in Fig. 2.3(c). We observe that the similarities decrease slightly even when 80% of edges are randomly removed. For the HVR network, where the layers are heterogeneous, the similarities are maintained at low levels, even though we randomly remove different percentages of links.

2.3.2 Validation of layer reconstruction and link prediction

The MAP method we introduced in this work aims to layer reconstruction and missing link estimation. The similarity between the target layer and other layers can be obtained through the SimHash algorithm, as introduced above. In this part, we show the effectiveness of the MAP algorithm and compare the differences between the MAP algorithm and the MLE algorithm’s performance. Other methods, such as entropy-based approaches, are equivalent to the MLE algorithm and are discussed in Appendix. C.

Here, we use the receiver-operator characteristic (ROC) curve and the area under the curve (AUC) to evaluate the effectiveness of our method. The area under the ROC curve (AUC) is a summary statistic that quantifies the overall performance of the model. An AUC of 1 means the model is perfect, while an AUC of 0.5 indicates the model performs as well as random guessing.

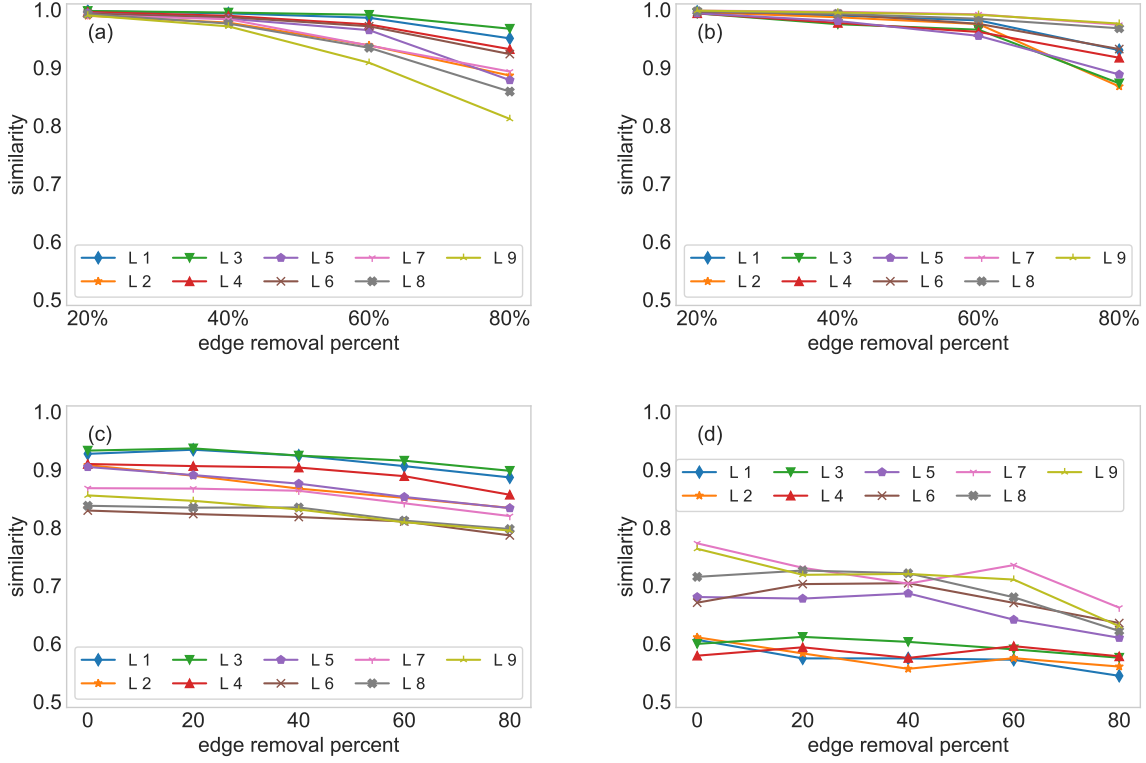


Figure 2.3: Eigenvector centrality-based SimHash algorithm. Panel (a) shows the comparison between the first nine layers of the FAO network and their incomplete counterparts, where 20%, 40%, 60% and 80% edges are removed uniformly at random from the target layer. Panel (b) shows the same experiment on the HVR network. Panel (c) compares the incomplete networks (the first nine layers of the FAO network) with the five most similar layers. The mean of the similarities is shown in the panel. Similarly, 20%, 40%, 60% and 80% edges are randomly removed from the target layer. Panel (d) shows the same experiment on the HVR network. Note that each similarity value is averaged over 10 runs.

The dimension of node vectors

The dimension of the node vectors is a crucial factor for the MAP algorithm's performance. In experiments, we did not find any particular number of communities that maximizes the estimation accuracy^{108–112}. For both the FAO network and HVR network, we remove 40% of edges from the target layer to generate incomplete networks. The five layers with the highest SimHash similarities are used to compute the parameters of the conjugate prior. The target layer is reconstructed through both the MLE algorithm and the MAP algorithm. In Fig. 2.4, we can observe an increased estimation accuracy with respect to the increment

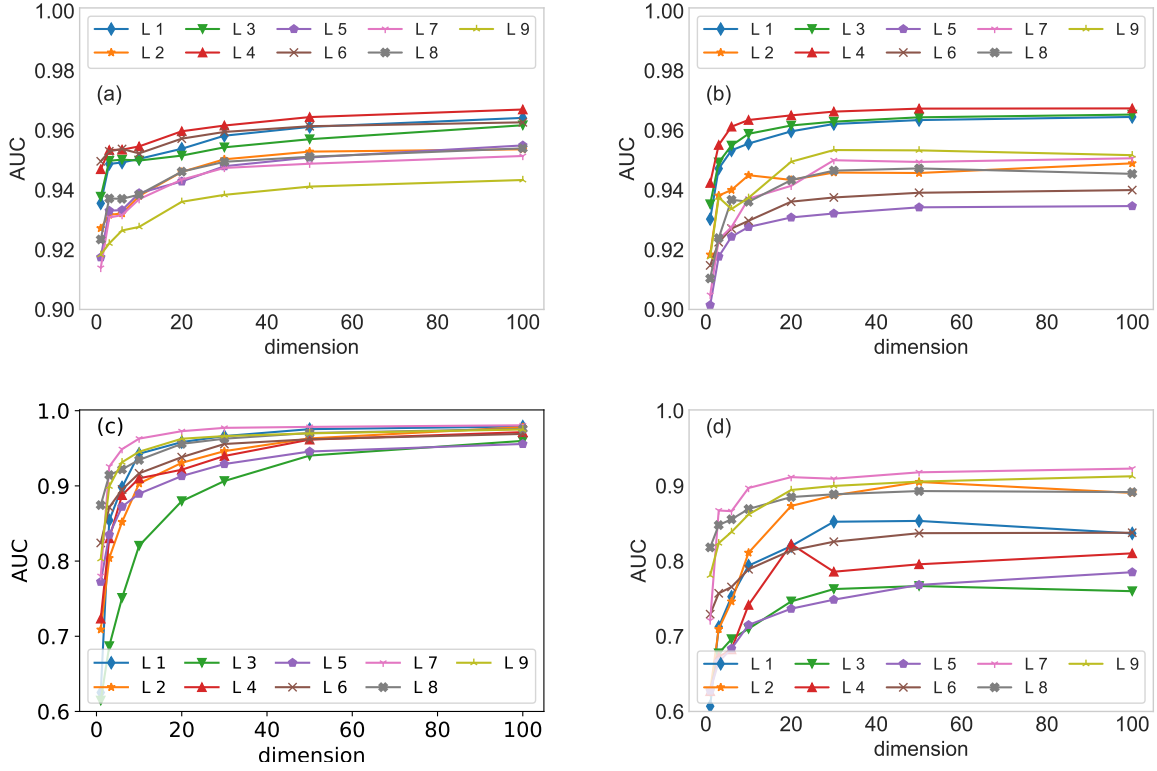


Figure 2.4: *AUC vs. the dimension of node vectors. Panel (a) shows the MLE method on the first nine layers of the FAO network, where each layer is set as the target layer, respectively. Panel (b) shows the results of the MAP method when the five layers with the highest similarities are adopted to compute the parameters of the conjugate prior. Panel (c) shows the results of the MLE method on the HVR network. Panel (d) is the result of the MAP method on the HVR network. Note that the results are averaged over 10 runs.*

of the number of dimensions, though the gain diminishes for dimensions greater than 40. In the following experiments, we use a dimension of 50, which balances running time and estimation accuracy. Apart from the dimension of node vectors, the estimation accuracy is also affected by the number of similar layers. The influence of the number of similar layers on the estimation accuracy is analyzed in Appendix. B.

Compare the MAP algorithm to the MLE algorithm

Both the MLE and MAP algorithms have their own advantages and disadvantages. The major difference between them is that the MAP method can incorporate prior information (other similar layers), while the MLE relies solely on the available information of the target

layer. To compare the performance of the two algorithms, we conducted experiments on both the FAO network (first nine layers) and the HVR network. The algorithms were implemented on incomplete layers generated by randomly removing 20%, 40%, 60%, 80%, and 100% of edges from the two networks.

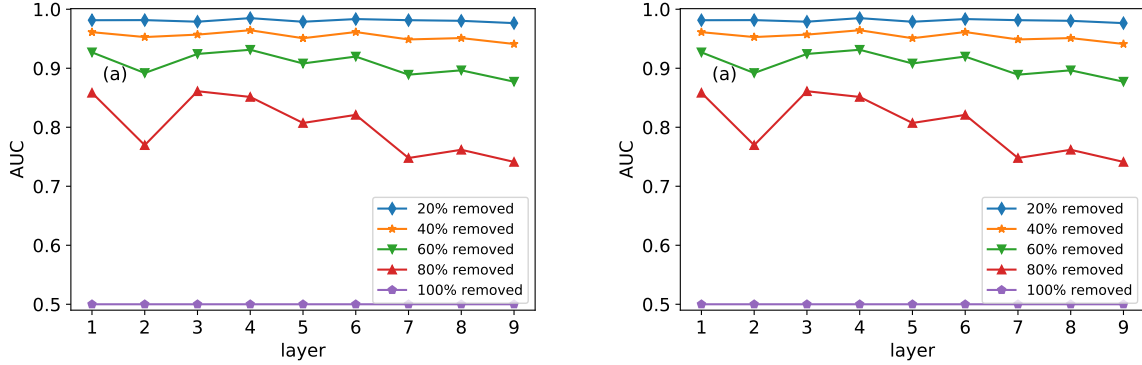


Figure 2.5: Comparison between the MLE and MAP methods on the FAO network. Panel (a) shows the results of the MLE method. Panel (b) shows the results of the MAP method. Results are averaged over 10 runs of cross-validations.

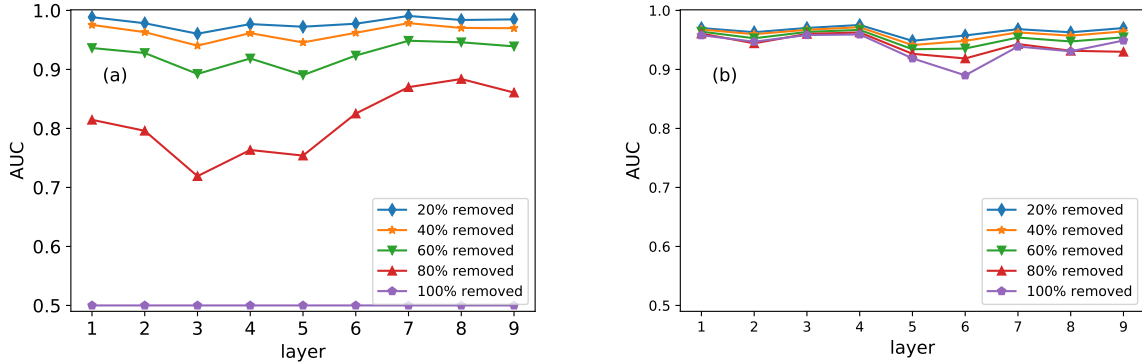


Figure 2.6: Comparison between the MLE and MAP methods on the HVR network. Panel (a) shows the results of the MLE method. Panel (b) shows the results of the MAP method. Results are averaged over 10 runs of cross-validations.

In Fig. 2.5(a) and Fig. 2.6(a), we can observe that the MLE algorithm's estimation is significantly affected by the missing links. However, the estimation robustness is greatly improved after incorporating similar layers to reconstruct the target layer, as shown in Fig. 2.5(b). On the contrary, the estimation accuracy deteriorates if we use layers with heterogeneous structures to reconstruct the target layer, which is shown in Fig. 2.6(b).

Overall, the MLE algorithm is suitable for cases where the available information of the target layer is relatively sufficient, and the missing links do not significantly affect the estimation accuracy. In contrast, the MAP algorithm can handle incomplete layers with missing links and is more robust when the available information of the target layer is limited. By incorporating similar layers' information, the MAP algorithm can provide better estimates of missing link weights and reconstruct the target layer's structure more accurately. Therefore, the choice of algorithm depends on the specific application and the available information regarding the target layer.

2.4 Summary

In this chapter, we have introduced a novel algorithm based on the maximum *a posteriori* (MAP) estimation for target layer reconstruction in multilayer networks. Our approach leverages the structural similarity between layers to reconstruct the target layer. First, we derived the MAP estimation for target layer reconstruction in multilayer networks in section 2.2. Second, we introduced the SimHash algorithm, which is based on eigenvector centrality, to detect structurally similar layers. The SimHash algorithm compares network features, making it robust to missing links. Third, we presented two scenarios for obtaining the parameters of the Gamma conjugate prior. In the first scenario, where the target layer is partially known, we used the SimHash algorithm to identify similar layers. In the second scenario, where the target layer is completely unknown, we used functionally similar layers to compute the parameters of the conjugate prior.

In section 2.3, we demonstrated the effectiveness of our approach. First, we showed that the SimHash algorithm returns consistent similarity levels for different percentages of missing links. Second, we demonstrated that the estimation accuracy could be improved by increasing the dimensionality of node vectors, with the gain diminishing for dimensions greater than 40. Third, we found that increasing the number of similar layers improves the consistency of estimation results. Finally, we compared the performance of the MLE and MAP methods on two real networks. Our experimental results showed that the MLE method

performs better when there are less than 20% missing links, while the MAP method returns more consistent results when the percentage of missing links is 40% or more.

Chapter 3

A correlation coefficient approach to construct gene co-expression networks from heterogeneous data¹

3.1 Background

Network analysis has been used widely in different areas of science^{19;20;22-31}, including for the construction of gene co-expression networks (GCNs) to predict gene function and regulation. A GCN is composed of genes represented as nodes, and significant co-expression of pairs of genes is indicated by links. Links are determined by measuring the co-expression patterns of genes under different conditions^{4;31}. High throughput technologies, such as microarray and RNA-seq, allow measuring simultaneously the expression levels for thousands of genes. Network construction and analysis of gene expression data then provide a system-level view of gene expression relationships, identifying the connection between each pair of genes.

Gene expression data are commonly organized into a gene expression matrix that consists of rows representing m genes and columns representing n conditions (or samples). To construct a gene expression network, first, a similarity score is calculated for each gene pair.

¹This chapter is a slightly modified version of a published article⁷⁴.

If the expression matrix is complete, the matrix consists of $m \times n$ data points, and each gene has an expression vector of length n . Thus, comparison of the expression of any given gene pair is based on n number of paired elements between their expression vectors, with a paired element between gene a and b defined as the expression value pair $a_i b_i$, where i is a specific condition, with $i = [1, 2, \dots, n]$. The similarity score between gene pairs in GCNs is commonly calculated by the Pearson correlation coefficient (PCC)^{31;113;114}. The PCC outperformed other means of similarity scores when constructing large gene co-expression networks¹¹⁵. Once the similarity score is calculated between all gene pairs, a similarity matrix $m \times m$ is assembled, with s_{ab} elements representing the similarity score between genes a and b . Second, based on the similarity matrix, an adjacency matrix is built, which defines network links using a threshold T , with a link existing if the similarity score is greater than T .

Several studies used the PCC to construct a GCN by setting a fixed threshold to determine co-expression between genes, which is appropriate for homogeneous expression data sets, where (1) expression values were obtained with the same technology, (2) the expression value distribution is comparable across all conditions, and (3) the number of paired elements is similar for all gene pairs^{4;31;114}. However, neither of these three prerequisites is likely to be met, when assembling a GCN based on a large number of data sets obtained with different technologies and under distinct experimental conditions. A striking example of such a heterogeneous gene expression data set is that of *An. gambiae*, which underlies the global expression map constructed by MacCallum *et al.*². This data set does not fulfill any of the three prerequisites detailed above for the following reasons: (1) The types of expression values obtained from distinct technologies and downstream analyses are different². For example, in two-channel microarrays, expression values are expressed as ratios between experimental and control conditions. In contrast, in single-channel microarrays and RNA-seq, the intensity of read numbers presents expression values. (2) As a consequence, the distributions of expression values vary widely between the different technologies. For instance, the expression values obtained with RNA-seq are overdispersed, ranging from zero to tens of thousands³⁵. (3) Missing values are ubiquitous in experiments. In some data

sets, the expression of only a subset of genes was sampled for specific purposes (e.g., the *An. gambiae* detox chip¹¹⁶). The fact that biological data sets do not meet these criteria presents a major challenge to network biology in general.

Prerequisites (1) and (2) can be met by applying normalization methodologies, including median shift² and z-score normalization. However, the missing value problem poses a separate challenge for the following reason. According to¹¹⁷, under the null hypothesis that two genes are not correlated, the number of paired elements between their expression vectors significantly affects the PCC density distribution. The theoretical analysis shows that the PCC distribution with 50 paired elements has a much lower variance than the distribution with ten conditions, which means that the PCC for any given gene pair tends to be lower when there are more paired elements in a given data set. In the *An. gambiae* expression matrix, the number of paired elements for each gene pair is not identical. Thus, a fixed threshold to determine links introduces a bias, as the co-expression of genes with a smaller number of paired elements is favored. One way to overcome this challenge was proposed by Lee *et al.*¹¹³. In their two-step protocol, initially, individual networks are constructed for each homogeneous sub-data set, by calculating PCC similarity matrices and using a fixed threshold to select links. The final GCN is then aggregated from the individual networks, by confirming a link if it exists multiple times across the individual networks.

In this chapter, we propose a novel method to construct a GCN for *An. gambiae* genes based on several hundred conditions from different publications and platforms. Specifically, to avoid favoring links between gene pairs with a low number of paired elements, we propose a sliding threshold-based method to construct a global *An. gambiae* GCN. To construct the network, we first apply z-score normalization to produce equal variances and means across all conditions. Second, we compute the PCCs of all gene pairs and divide the gene pairs into 26 different groups according to their number of paired elements. Third, we select links based on a sliding threshold, such that, for each group, only the gene pairs within the top 0.5th percentile of PCCs will be connected by a link. The resulting network, which we name AgGCN1.0, remains robust with random removal of up to 15% of conditions. The AgGCN1.0 has similar characteristics to small-world and scale-free networks, as it contains

hub genes that are co-expressed with many other genes. Analysis of the network reveals that both the architecture of the core sub-network and the network communities are based on gene function, supporting the power of the proposed method for GCN construction.

3.2 Model Formulation

3.2.1 Description of the gene expression data set

The data set used for constructing the *An. gambiae* GCN, *Anopheles-gambiae_EXPR-STATS_VB-2019-02*, is based on the data set used by MacCallum *et al.*², which was updated by addition of several new conditions. *Anopheles-gambiae_EXPR-STATS_VB-2019-02* is available through Vectorbase (vectorbase.org;^{64;65}) at <https://tinyurl.com/mr38a7hj>. *Anopheles-gambiae_EXPR-STATS_VB-2019-02* is based on the AgamP4.11 annotation of the *An. gambiae* PEST genome, and includes log2-transformed expression values of 13,080 genes across 291 conditions, collected from 35 data sets^{35–63;66;118–122}. Each publication contributed, on average, eight conditions to the data set, ranging from 1⁵⁷ to 52⁴³.

The experimental methodologies varied widely among publications, exploring gene expression changes using different experimental platforms that sampled either across the entire or various sub-sections of *An. gambiae* transcriptome, using total RNA collected from various life stages, tissues, and physiological conditions. Expression data obtained with single-channel microarrays represented 62% of the data set^{37–39;41–45;48–51;54}, of which 84% were obtained with the Affymetrix GeneChip® Plasmodium/Anopheles Genome Array^{37–39;42;43;45;48–51}. Expression data for the remainder of the conditions were assessed with either dual-channel microarrays (23% of conditions,^{36;40;46;47;52;54–59;63;118–122}) or by RNAseq (15% of conditions,^{60–63;66}). With regards to life stages, 84% of all conditions were analyzed using samples derived from adults, 91% of those from female mosquitoes, a bias that is easily explained by the fact that only adult female mosquitoes transmit vector-borne disease pathogens¹²³. Similarly, tissue-specific analyses focused most commonly on the adult female midgut (33% of the conditions that sampled individual tissues^{36;38;40;42;46;47;61;119}), as it constitutes a major bottle-neck for

vector-borne pathogens after being ingested by the mosquito with a blood meal from an infected individual^{124;125}. Likewise, the conditions sampled a variety of physiologies that are integral to vector biology and its control, including blood feeding (9% of conditions,^{38;60;63;122}), parasite infection (14% of conditions,^{40;46;47;51;52;61;119}, and insecticide resistance (6% of conditions,^{35;55–57;59;118;120;121}).

3.2.2 Expression data set pre-processing and normalization

This initial data set was partially conflated as it contained 21 conditions^{36;41;54} that were combined from other conditions in the same data set (e.g., a conflated condition presented a ratio of two other conditions in the data set). Therefore, these 21 conditions did not represent new data, and thus were removed from further analysis. In addition, one platform, the dual-channel microarray *LIV A. gambiae DETOX 0.25k*¹¹⁶, interrogated the expression of only 226 genes or 1.7% of the *An. gambiae* transcriptome. The inclusion of the 13 conditions that used the *LIV A. gambiae DETOX 0.25k* microarray^{118–122} led to a large number of missing values in the data set, and therefore were excluded from the analyses. After their removal, the final data set (Table S1, in Supplementary Materials <https://github.com/KSUNetSE/AgGCN1.0>) consisted of gene expression data across 257 conditions collected from 30 publications^{35–63;66}.

However, direct comparison of expression values across the 257 conditions in the data set was not possible, as the data distribution varied among conditions, not only due to the platforms used but also due to variations between experimental designs and procedures (Table S4, in Supplementary Materials <https://github.com/KSUNetSE/AgGCN1.0/>). Fig. 3.1 shows the means and variances of expression values in each of the 257 conditions. The log2 mean gene expression values ranged from -0.9 to 10.9, with means for dual-channel microarray data usually around 0, single-channel microarray data ranging between 4.2 and 10.9, and RNAseq data ranging from -0.9 to 2.5 (Table S2, in Supplementary Materials <https://github.com/KSUNetSE/AgGCN1.0/>). The variance of log2-transformed expression values ranged between 0.01 and 16.5, with no correlation between mean and variance across

the data set or data obtained by RNAseq. However, expression data obtained with dual-channel microarray platforms showed a strong positive correlation between expression mean and variance across conditions, while data from single-channel microarrays showed a strong negative correlation. This negative correlation can be largely explained by the data characteristics of individual dual-channel microarray platforms. Data obtained with both the OXFORD Anopheles gambiae Agilent 13k v1 and the Agilent A. gambiae 020449 44k v2 microarray had low means and high variance within conditions, data obtained with the Affymetrix GeneChip® Plasmodium/Anopheles Genome Array showed means and variances in the middle range, and those obtained with the ND Anopheles gambiae Nimblegen 65k v1 microarray had high means and low variance. These observed differences in means and variance across experiments would result in gene expression correlation based on experimental design rather than on underlying gene regulation.

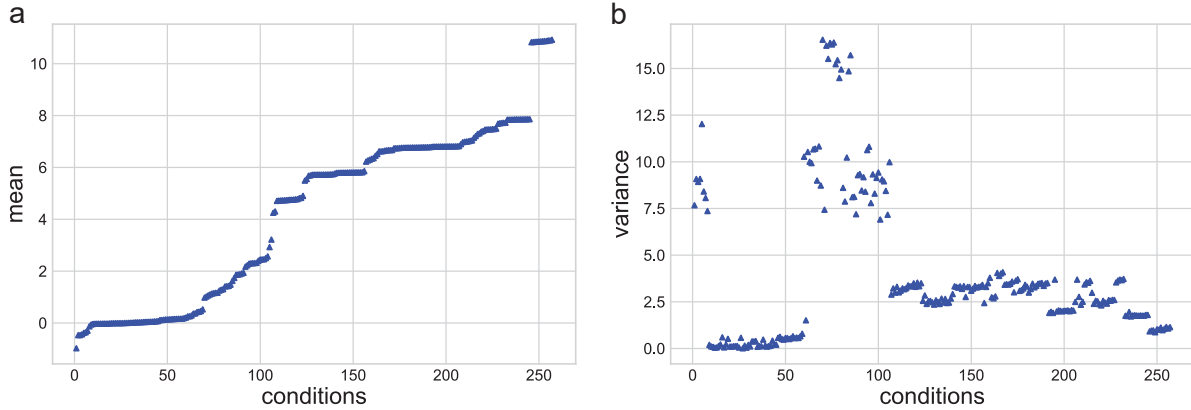


Figure 3.1: *Mean and variance distributions of gene expression values in the data set. Means of expression values for each condition are shown in increasing order (a). The corresponding variance of each mean is shown in (b).*

To equalize the means and variance of the data across all conditions, we performed a normalization step using the z-score, which is expressed as follows

$$\hat{x}_i(k) = \frac{x_i(k) - u_i}{\delta_i} \quad (3.1)$$

where $\hat{x}_i(k)$ is the normalized expression value of gene k under condition i , $x_i(k)$ is the raw expression value of gene k under condition i , u_i and δ_i are the mean and the standard

deviation of the expression value for all values in condition i , respectively.

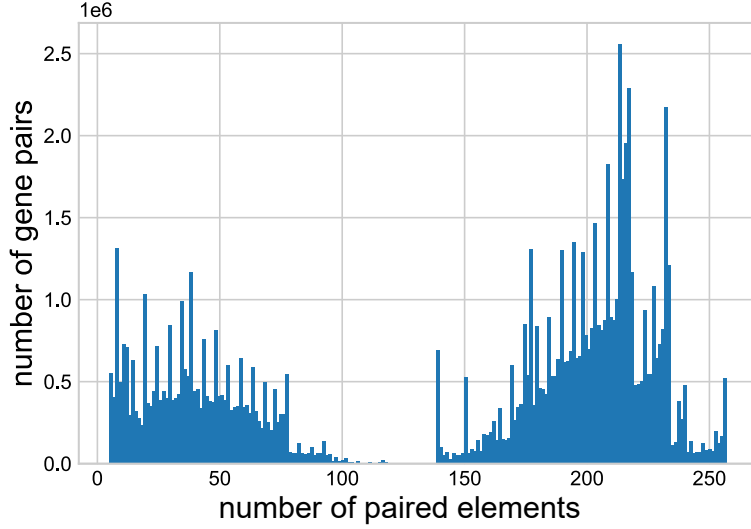


Figure 3.2: *The distribution of the number of paired elements of all the gene pairs.*

Another property of the *An. gambiae* gene expression matrix was the highly variable number of paired elements between all possible gene pairs (Fig. 3.2). A paired element between gene a and b is defined as the expression value pair $a_i b_i$, where i is a specific condition, with $i = [1, 2, \dots, n]$. If either a_i or b_i is missing in the expression matrix, there is no paired element between gene a and b for condition i . The distribution of paired elements across the entire z-score-normalized data set was highly heterogeneous, displaying a bimodal distribution with a gap between 118-139 paired elements. This gap is explained by the properties of the platforms used to obtain the gene expression data. The transcripts of 2,813 annotated genes in the data set were not represented on the Affymetrix GeneChip® Plasmodium/Anopheles Genome Array, which was used to analyze gene expression in 139 out of 257 total conditions^{38;39;41–43;45;48–51}. Therefore, each of these 2,813 genes in the data set had an expression vector length that is ≤ 118 , and the 10,433 genes represented on the array had a vector length ≥ 139 . While z-score normalization addressed the differences in means and variance of the expression matrix, the variable paired element length remained and had important implications on edge selection and edge weight assignment, which are discussed below.

3.2.3 Edge selection based on a sliding Pearson correlation coefficient threshold

To determine the edges of the AgGCN1.0, we used the Pearson correlation coefficient (PCC), which is a co-expression measure used commonly to detect edges for nodes in gene co-expression networks^{113;117;126}. We calculated PCCs for all possible gene pairs in the data set, and constructed an initial gene expression correlation matrix, which included all PCCs. From this matrix, we removed the PCCs from all gene pairs whose expression vectors had a paired element length of ≤ 4 .

The PCC r_{xy} was calculated as

$$r_{xy} = \frac{\sum_{i=1}^n (\hat{x}_i - u_x)(\hat{y}_i - u_y)}{\sqrt{\sum_{i=1}^n (\hat{x}_i - u_x)^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - u_y)^2}} \quad (3.2)$$

where n is the number of paired elements between gene x and y , $\hat{x}_i$ is the normalized expression value of gene x under condition i , u_x is the mean of gene x across all paired elements, $\hat{y}_i$ is the normalized expression value of gene y under condition i , u_y is the mean of gene y across all paired elements.

From Eq. (3.2), it is apparent that the range of the PCC is $[-1, 1]$. It is assumed that the two vectors are negatively and positively correlated when r_{xy} close to -1 or 1, respectively. If the two node-vectors are assumed independent, the density function of PCC is then expressed as

$$P(r) = \frac{1}{\sqrt{\pi}} \frac{\Gamma(\frac{v+1}{2})}{\Gamma(\frac{v}{2})} (1 - r_{xy}^2)^{\frac{v-2}{2}}, \quad (3.3)$$

where $\Gamma()$ is the gamma function, and $v = n - 2$ is the degrees of freedom.

Fig. 3.3 shows the probability density distribution of the PCC as a function of paired element length. The standard deviation of PCCs decreases with an increase in the number of paired elements. As an example, selecting the top 3rd percentile of all PCCs at a given

paired element number leads to a threshold PCC of ≥ 0.61 for ten paired elements, and a threshold PCC of ≥ 0.12 for 250 paired elements (Fig. 3.3).

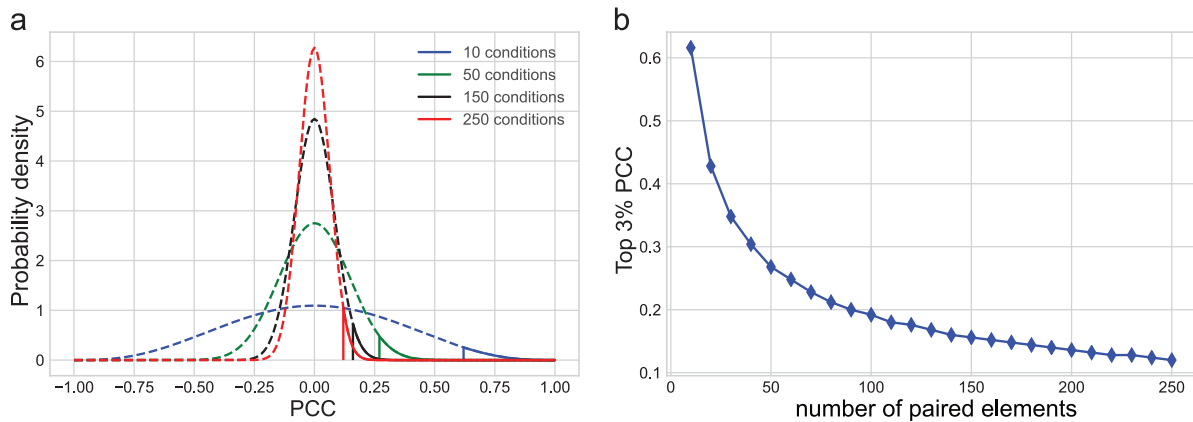


Figure 3.3: Comparison of the probability density function of PCCs with 10, 50, 150, and 250 paired elements. (a) Probability density functions of PCC, where the vertical lines are the top 3% thresholds. (b) The top 3% PCC with respect to the number of paired elements.

In GCN construction, an edge between a gene pair is included if their expression vectors are deemed to have a *high* PCC, most commonly using a fixed threshold, such as 0.8¹¹³. However, a fixed threshold is only valid when the number of paired elements in the expression matrix is constant. Given the heterogeneous distribution of paired element length in the *An. gambiae* expression matrix, a fixed threshold of PCC would favor edges between gene pairs with fewer paired elements and not capture the edges with larger numbers of paired elements, which arguably have stronger experimental support.

To avoid this problem, we selected gene pairs among the top percentages of PCC values and used a sliding threshold based on the number of paired elements. To do so, we divided the PCCs into 26 groups according to the interval of the paired element length values, i.e., [4, 10], [11, 20], [21, 30], ..., [251, 257]. The scattered points in Fig. 3.4 show the top 0.5%, 1%, 2%, and 3% of PCCs of all gene pairs in the 26 intervals. To assign a threshold of correlation between the expression vectors of any given paired element length, we used a curve that fitted the scattered points. The gene pairs with PCCs above the fitted curve were

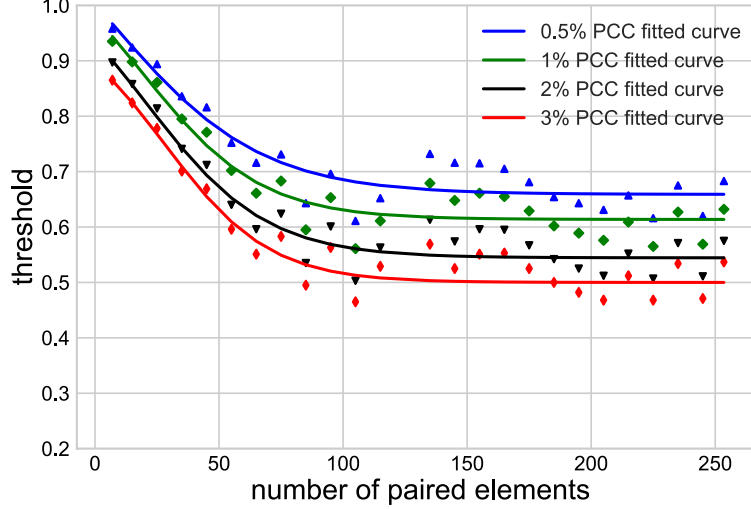


Figure 3.4: Top 0.5%, 1%, 2%, and 3% PCC points with respect to different numbers of paired elements. Scattered points are the top percentages computed from the data set, and solid blue lines are the fitted curves.

assigned edges in the GCN. The equation for the curve is

$$f^{thres}(x) = \alpha - \frac{1}{\eta + \lambda e^{-\frac{x}{\beta}}} \quad (3.4)$$

where α , η , λ , and β are the four parameters that were fitted, and x is the number of paired elements. This equation provided a good trade-off between the accuracy of the fitting and the number of parameters to estimate. We optimized the four parameters α , η , λ , and β iteratively, by (1) fixing parameters λ and β , and optimizing α and η , and (2) fixing α and η , and optimizing λ and β . We evaluated the performance of the four parameters by maximizing the coefficient of determination, R^2 . The solid lines in Fig. 3.5 show the fitted curves for the top 0.5%, 1%, 2%, and 3% points, and the optimized parameters α , η , λ , and β are shown in Table 3.1. Optimized parameters did not change substantially when a higher interval number was chosen. For example, using 52 instead of 26 intervals to calculate the top 3% sliding threshold curve resulted in the same values for the α , η , and minimally changed λ , and β (52 intervals: 7.6 and 21.1; 26 intervals: 7.5 and 21.2).

Edges were included in the network if their PCC (1) was above the sliding threshold, and (2) had a p -value smaller than δ/m , where δ and m are the significance level equal to 0.05,

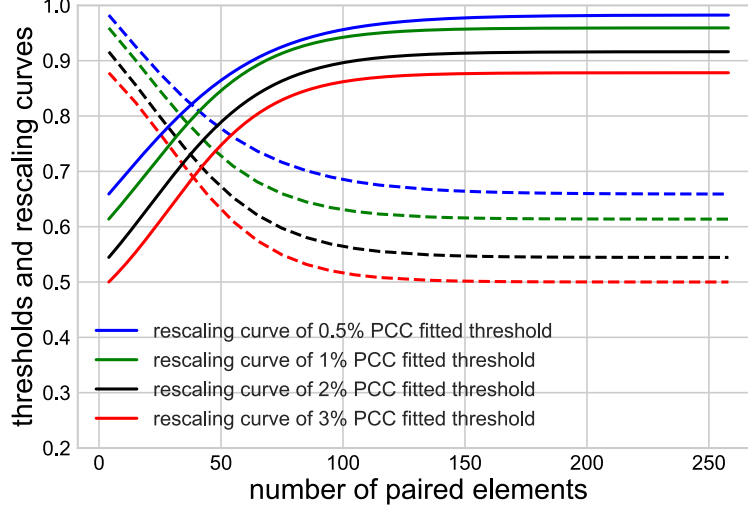


Figure 3.5: *Edge weight rescaling. Sliding PCC-threshold curve (dashed lines) and rescaling curves (solid lines).*

and the number of gene pairs of each interval, respectively (Bonferroni-correction, [127;128](#)).

Table 3.1: *Optimized parameters*

Parameters	α	η	λ	β
0.5%	1.28	1.61	2.0	30
1%	1.14	1.90	4.3	23.7
2%	1.10	1.80	4.3	24.0
3%	1.00	2.00	7.5	21.2

3.2.4 Edge weight assignment

With the goal of providing a more informative network structure that identifies the relative strength of co-expression between gene pairs, we constructed a weighted network. To do so, we assigned a weight to every detected edge, representing the similarity score of co-expression based on PCC and the number of paired elements. As the edges detected with fewer paired elements tended to have higher PCCs than the edges with more paired elements (Fig. [3.4](#)), we rescaled the PCCs to equalize the means and medians of the PCCs across the 26 intervals. Fig. [3.5](#) shows the PCC-rescaling curves for the sliding thresholds. The dashed lines are the PCC-sliding threshold curves represented by Eq. [3.4](#), and the solid lines are

the corresponding rescaling curves, given by the following equation

$$f^{rescale}(x) = \min(f^{thres}(z)) + \max(f^{thres}(z)) - f^{thres}(x), \quad (3.5)$$

where $z \in [4, 257]$, z is the range of the number of paired elements. This curve reduced the weight of those edges that were detected with a smaller number of paired elements. The edge weight was computed as

$$Weight_{edge}(x) = PCC \times f^{rescale}(x) \quad (3.6)$$

We assigned weights to all edges that were selected based on the sliding thresholds of top 0.5%, 1%, 2%, and 3% PCCs. Fig. 3.6 confirms that the rescaling curve normalized the means and medians of edge weights across all intervals. As expected, the distribution of the PCC means and medians of the selected edges across the 26 intervals of paired element lengths followed a similar pattern to those observed for the PCC threshold curves (Fig. 3.6a and Fig. 3.4). After the rescaling of edge weights using Eq. (3.6), the means and medians of the edge weights in each interval were indeed similar, and thus comparable across the intervals (Fig. 3.6b), confirming the validity of the rescaling approach.

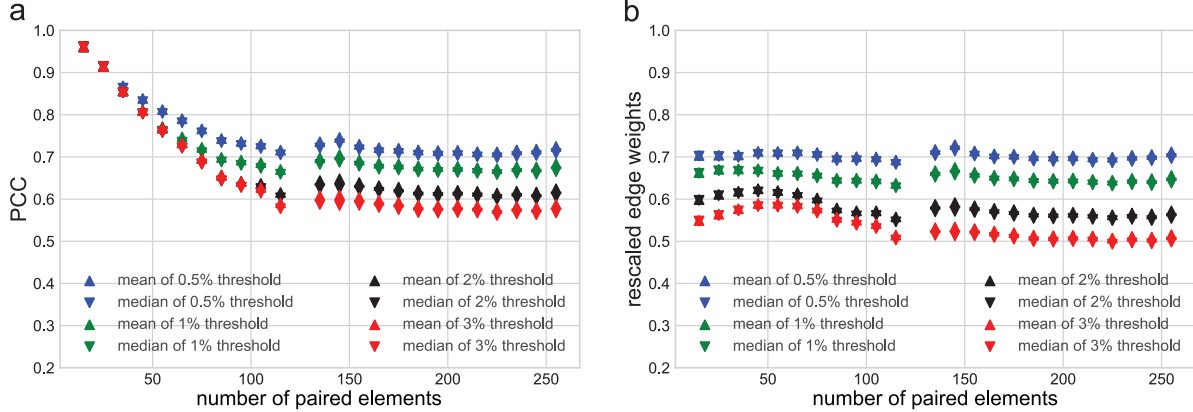


Figure 3.6: Means and medians of the PCCs above the threshold in each interval. (a) unnormalized means and medians of PCCs. (b) means and medians of rescaled edge weights.

3.2.5 Network selection

To determine the PCC threshold for final network construction, we built and analyzed four networks with the top 0.5%, 1%, 2%, and 3% sliding thresholds as introduced in Section 3.2.3. Table 3.2 shows the statistical properties of the four networks. The curated data set contained expression data for 13,080 genes (nodes) annotated in the *An. gambiae* genome. Table 3.2 shows the number of nodes connected with at least one edge, and the node percentage represents the corresponding percentage of genes present in the network. The LCC is the largest connected component of the network. The network density is defined as

$$density = \frac{\text{number of edges}}{\frac{n(n-1)}{2}}. \quad (3.7)$$

where n is the number of nodes in the network.

All four networks contained more than 90% of nodes, with a small node percentage increase with increasing thresholds. In contrast, the number of edges and network density nearly doubled with each threshold increase. Based on these results, we selected the network with the most stringent edge selection criterion (top 0.5% sliding threshold), as the AgGCN1.0 network. This network maintains its structure with a large number of nodes connected with the smallest number of edges. The AgGCN1.0 network, therefore, contains nearly all genes encoded in the *Ag. gambiae* genome and shows co-expression between pairs of genes only when their expression vectors have a very high correlation.

Table 3.2: *Properties of the entire network with different sliding thresholds*

Sliding threshold	0.5%	1%	2%	3%
No. of nodes	11825	12290	12415	12431
Node percentage	90.4%	94.0%	94.9%	95%
No. of edges	382168	685759	1307104	1896825
Network density	0.55%	0.91%	1.70%	2.46%
No. of connected components	13	4	5	5
No. of nodes in LCC	11794	12278	12401	12417
LCC node percentage	90.2%	93.9%	94.8%	94.9%
No. of edges in LCC	382144	685745	1307089	1896810
LCC density	0.55%	0.91%	1.70%	2.46%

3.3 Methodology verification and robustness test

Next, we validated the methodology of network construction by testing for systematic errors in the procedure and studied the robustness of the AgGCN1.0 network by randomly removing different percentages of conditions from the data set.

3.3.1 Methodology verification

With the construction of the AgGCN1.0 completed, we next verified that the network structure was based on the underlying expression matrix rather than based on a systematic error in the method of construction. To this end, we randomized the expression values under the same condition and reconstructed the network with the method introduced in Section 3.2 using the following procedure:

- Step 1: Randomly reshuffle the expression values for each condition.
- Step 2: Compute the PCCs for all of the gene pairs.
- Step 3: Use the top 0.5% PCCs fitted sliding threshold to select edges.
- Step 4: Repeat step 1 to step 3 100 times.

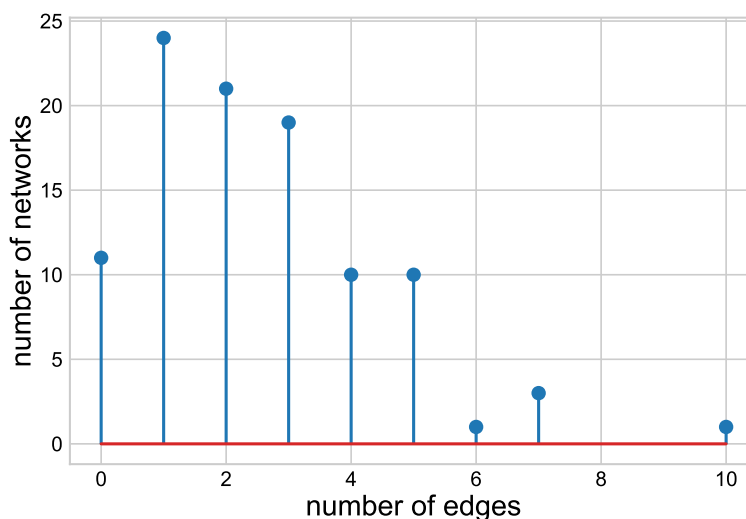


Figure 3.7: *The number of edges from the reshuffled data sets.*

Fig. 3.7 shows the distribution of the number of edges of the 100 networks. The AgGCN1.0 network contains 382,168 edges, while the number of edges recovered in the reconstructed networks is smaller or equal to ten. Based on this result, we did not detect a systematic error in the proposed network construction method and concluded that the structure of AgGCN1.0 is indeed based on its expression matrix.

3.3.2 Robustness test

The AgGCN1.0 network is based on a meta-analysis of expression data obtained from many conditions, and it is unclear how sensitive the overall network structure is to the number of conditions included in the data set. To evaluate network structure sensitivity, we randomly removed an increasing number of conditions and reconstructed the network. The network is considered robust if the edges in the reconstructed networks recapitulate the majority of those in the original network.

Specifically, we randomly and iteratively (100x) removed 3 (1%), 13 (5%), and 39 (15%) conditions and used the top 0.5% PCCs fitted sliding threshold of the AgGCN1.0 to detect edges. Table 3.3 shows the comparison between the original network and the reconstructed networks. The average number of edges in the reconstructed networks dropped with an increasing number of removed conditions, with a 15% decrease in conditions leading to a reduction of 11.3% in the number of edges. Over 90% of the edges present in the AgGCN1.0 network continued to be detected with 13 conditions removed. However, the number of overlapping edges decreased on average by 22.3% when 39 conditions were removed. To compare more specifically how the removal of conditions influenced edge loss in the AgGCN1.0 network, we determined which AgGCN1.0 edges in each of the 26 paired element length intervals were retained in a sample network that was constructed after the removal of 3 (1%), 13 (5%), and 39 (15%) conditions, respectively (Fig. 3.8). Overall, edges were largely retained as long as the paired element length was greater than 50. As expected, the removal of 39 conditions did not recover edges with fewer paired elements and thus overall low experimental support. Together, these results show that the network construction methodology is robust

with respect to random removal of up to 15% conditions. These results also demonstrate that the sensitivity of the AgGCN1.0 structure to the underlying number of conditions is mostly limited to the loss of those edges derived from the correlation of expression vectors with few paired elements.

Table 3.3: *statistical results of the original and removed networks*

Removed conditions	Gene pairs	Ave. edges	Ave. edges in common (%)
0	83,781,071	382,168	382,168 (100%)
3	83,768,208	381,010	366,104 (95.8%)
13	83,706,380	373,820	347,205 (90.9%)
39	83,511,993	350,306	300,939 (78.7%)

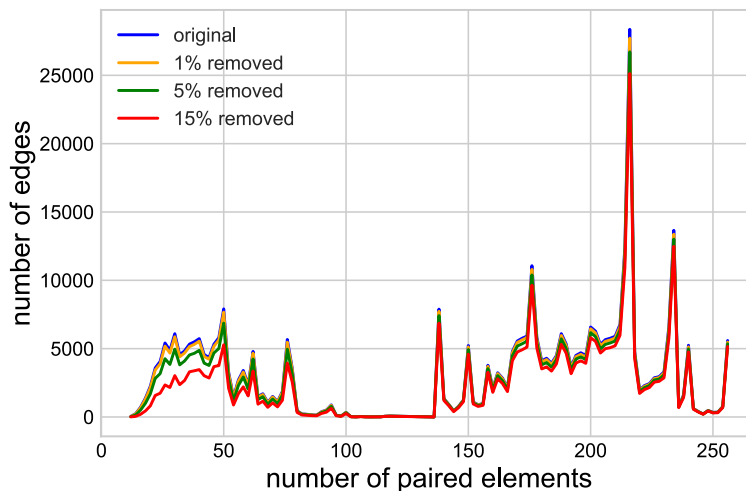


Figure 3.8: *The distribution of remaining edges in the network after removing 1%, 5% and 15% conditions. The horizontal axis is the number of paired elements and the vertical axis is the number of edges.*

3.4 Analysis of the AgGCN1.0 largest connected component

The AgGCN1.0 network constructed through the method introduced in Sec. 3.2 using a top 0.5% sliding threshold is composed of 13 connected components (table 3.2), the largest of

which (the LCC) consists of 11,794 nodes and 382,144 edges. In this section, we characterized the LCC network by computing node centralities and detecting its core and communities.

3.4.1 Node centralities

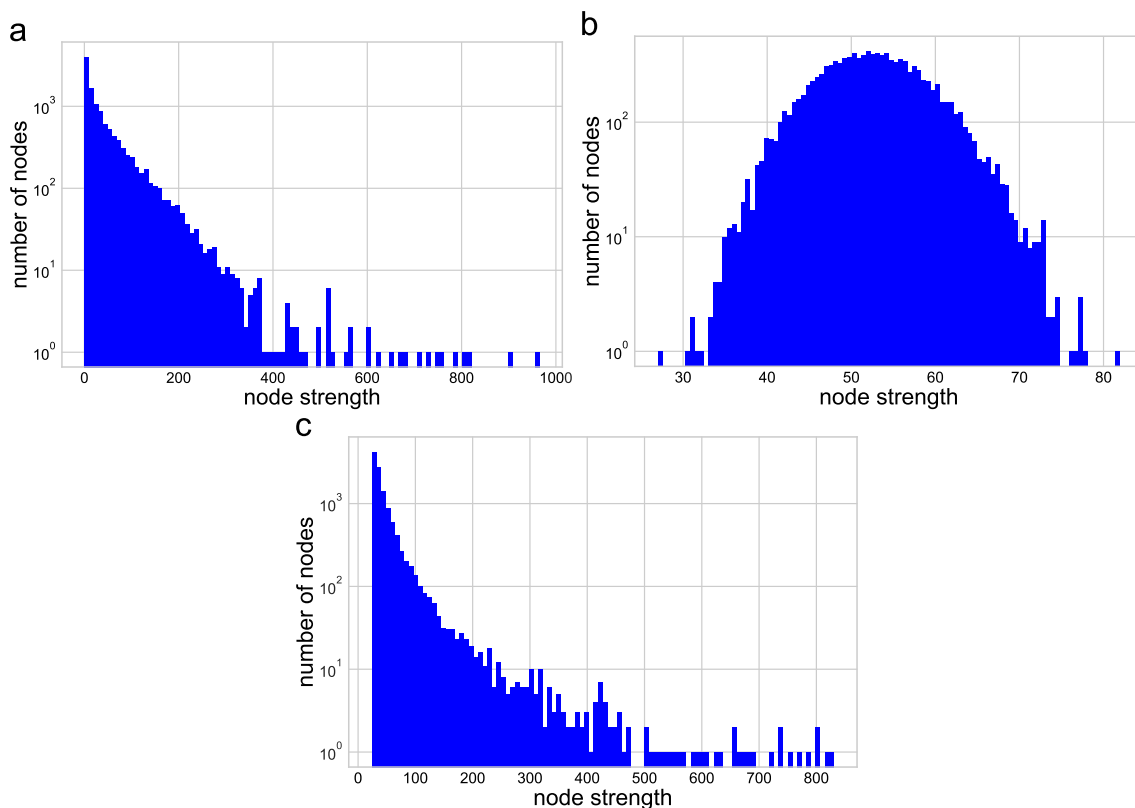


Figure 3.9: Node strength distribution of the network. (a) The strength distribution of the AgGCN1.0 LCC network. (b) The strength distribution of the ER random network. (c) The strength distribution of the BA random network.

Centralities of network nodes are calculated to detect nodes that can potentially play a critical role in network connectivity, evolution, and dynamics. This general feature extends to GCNs, where node centrality measures have been used successfully to identify genes essential for organism survival^{129;130}. The simplest node centrality is the node degree, which is defined as the number of links incident on a node. When links are weighted, the node degree becomes the node strength, defined as the sum of the weights of all node's edges. The LCC node strength has an average of 45.7 and spans a wide range of values, included between 0.65 and 965.4. To better understand the characteristics of the AgGCN1.0 LCC, we

compared its centralities with the corresponding centralities of two networks generated using the Erdos-Renyi (ER) ¹³¹ and the Barabasi-Albert (BA) ¹³² random network models. The ER network is the simplest network model, where for any pair of nodes a link is built with a given probability. The BA network is a model of a scale-free growing network, where new nodes connect based on preferential attachment and was used here to build a network with nodes with large degrees (hubs). By design, the ER and BA networks had the same number of nodes and a similar number of links as the LCC (Table 3.4). Since the AgGCN1.0 LLC is a weighted network with weights approximately in the interval $[0.65, 0.97]$, in the generation of the two random networks, we assigned weights to the links uniformly at random from the interval $[0.65, 0.97]$.

Table 3.4 shows the average, minimum, and maximum node strengths of the three networks. While the average value remained similar across the three networks, the range of variability of the ER network was much smaller than those of the two other networks. The similarity of the LCC network with the BA network was also confirmed by comparing the corresponding node strength distributions (Fig.3.9). These distributions were similar, both showing the presence of *hubs*, i.e., nodes with very large strength values. The presence of hubs is a characteristic of many real networks and provides robustness to the structure with respect to random perturbations of nodes. To measure quantitatively how well the AgGCN1.0 network satisfied the scale-free property, we adopted the model fitting index R^2 on the log-log strength distribution, obtaining values of 0.884 for the AgGCN1.0 LCC and 0.967 for the BA network. The latter value is close to one because the BA network is scale-free by construction. This numerical test confirmed that the AgGCN1.0 LCC can be considered approximately a scale-free network.

Two other properties of the three networks are computed and compared in Table 3.4, namely the average clustering coefficient and the average shortest path length. While the clustering coefficient shows how well connected are the neighbors of a given node, the average shortest path length is a measure of the distance of node pairs in the network. The AgGCN1.0 LCC had a much higher clustering coefficient than the ER network and the BA network, which is typical of real networks with communities. In computing the classical shortest path,

a small edge weight represents a shorter distance between two nodes. In contrast, in the AgGCN1.0 LCC, higher edge weights represent closer connections. Therefore, the reciprocal of the edge (link) weight is used as the link length to calculate all shortest paths. We then used these shortest paths to compute the average shortest path length, betweenness, and closeness. In Table 3.4, the average shortest path length of the AgGCN1.0 LCC is compared with that of the ER network and the BA network. The average shortest path of the AgGCN1.0 LCC was only slightly longer in comparison. The two random network models are characterized by the small-world property, which means that there is a path between a pair of nodes that involves only a few short edges and the clustering coefficient is not small. Overall, we can conclude that the AgGCN1.0 LCC also presents some small-world characteristics. Many biological networks, including GCNs show some degree of this property (e.g. ¹³³⁻¹³⁵), and it may reflect an evolutionary advantage of such a structure. One possibility is that small-world networks are more robust to random perturbations than other networks and this would provide an advantage to biological systems that are subject to damages, such as gene mutations.

Table 3.4: *Topological properties of the co-expression network*

Properties	AgGCN1.0 LCC	ER network	BA network
Edges	382,144	382,723	376,384
Ave. node strength	45.7	52.6	51.7
Min. node strength	0.65	27.0	24.3
Max. node strength	965.4	82.0	875.3
Ave. clustering coefficient	0.24	0.0046	0.019
Ave. shortest path	4.8	3.1	2.9

Other centrality measures computed in this analysis, in addition to the node strength, are the following three ¹³⁶:

- Eigenvector centrality: the centrality of a node is determined by the entry of the eigenvector corresponding to the largest eigenvalue of the adjacency matrix representing the AgGCN1.0 LCC.
- Betweenness centrality: the centrality of a node is determined by the number of shortest

paths that pass through the node itself.

- Harmonic closeness centrality: the centrality of a node is based on its distance to all other nodes. Closeness centrality is the sum of the shortest path distance reciprocals of a node to all other nodes. It is calculated as

$$C(x) = \sum_y \frac{1}{d(x, y)} \quad (3.8)$$

where $C(x)$ and $d(x, y)$ are the closeness of node x and the shortest path distance between node y and x , respectively.

Fig. 3.10 shows the four node centralities in the network by visualizing the size of the node proportionally to its centrality. High-strength nodes were distributed in the areas where nodes are tightly connected (Fig.3.10a, the left and right clusters). However, only nodes in the right cluster also had relatively high node eigenvector centrality, as indicated by their bigger node size (Fig.3.10b). In contrast, only few nodes had a high betweenness centrality (Fig.3.10c), while many nodes had a similar harmonic closeness centrality (Fig.3.10d).

Overall, we did not identify any correlation between gene expression level and either of the four centrality measures reported here (data not shown). Thus, these centrality measures can potentially be used as additional node characteristics. However, since the AgGCN1.0 did not contain a single group of nodes that is characterized by high centrality for all four measures, it will be critical to define the gene property to be studied and determine which measure more closely represents such property to detect the key nodes (genes). Additionally, if an evolutionary process would target the top central nodes in the AgGCN1.0 LCC network, these nodes would be different on the basis of the selected centrality, providing a diversity advantage.

3.4.2 Communities

A community in a network is a subgraph that is highly connected internally and loosely connected with other subgraphs. Community detection for the AgGCN1.0 LCC is essential,

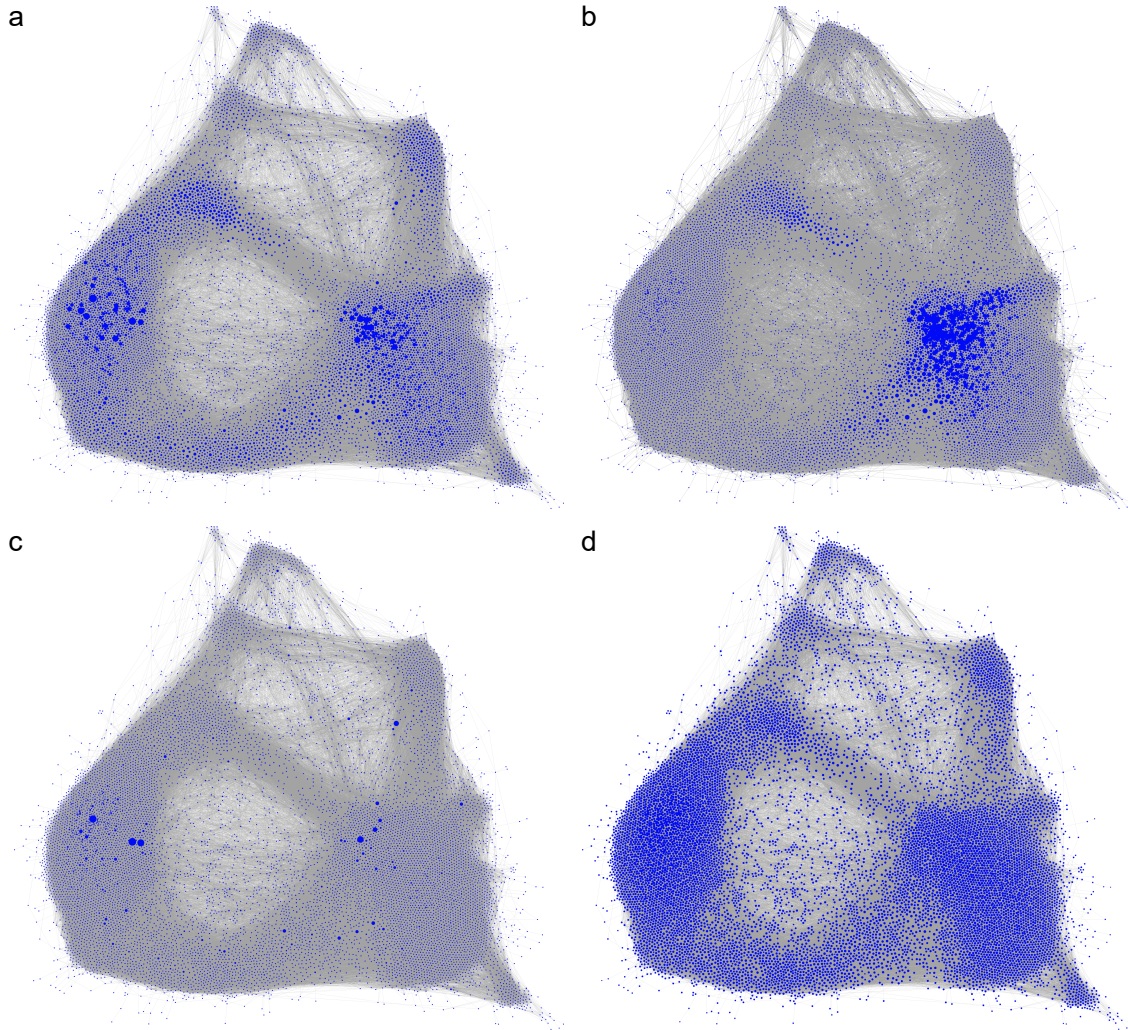


Figure 3.10: Centrality measures for the AgGCN1.0 LCC. The layout of the network is generated through Gephi with ForceAtlas 2.0 algorithm¹³⁷. (a) node strength centrality. (b) eigenvector centrality. (c) betweenness centrality. (d) harmonic closeness centrality. The size of a node is proportional to its centrality value.

since the identified communities can help discover the underlying biological processes that shape the network. In recent years, various community detection algorithms have been proposed. According to^{108;114;138}, the Louvain algorithm and the Infomap perform better than other methods, with the extra advantage of low computational complexity. However, the Infomap method tends to cut leaf nodes into isolated communities, which results in numerous tiny communities. For this reason, in this work, we adopted the Louvain algorithm to detect the communities. The Louvain algorithm is a modularity optimization method that hierarchically identifies how nodes are clustered. Modularity measures the difference

between the AgGCN1.0 LCC and a random network in terms of community existence. The modularity Q of a network is calculated as follows¹³⁸:

$$Q = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j), \quad (3.9)$$

where Q , k_i , k_j , m and A_{ij} are the modularity, the degrees of node i and j , the total number of edges in the network, and the weight of the edge between node i and j respectively. The Kronecker delta function δ is equal to 1 if c_i equals c_j , which means the two nodes are in the same community, while δ is equal to 0 when the two nodes are in different communities.

In the AgGCN1.0 LCC, we detected 15 communities using the Louvain algorithm, as shown in Fig. 3.11. Most of the nodes are included in 13 large communities, while two small communities contain less than 1% of nodes. In the network visualization shown in Fig. 3.11, nodes belonging to the same community were visualized in proximity, due to the algorithm selected to visualize the network. In particular, the layout of the network is based on the ForceAtlas 2.0 algorithm, which produces visual densities that denote structural densities¹³⁷. Visualization of the force-directed layout of the AgGCN1.0 LCC confirmed the existence of well-defined communities. Computing the average shortest path between pairs of communities revealed that communities visualized in proximity are also characterized by a relatively shorter average path length. For example, the average shortest path length between community 11 and community 2 was 3.276, as compared to 4.764 between community 11 and community 15.

3.4.3 Cores

The k -core of a graph is a maximal subgraph in which each node has at least k neighbors after removing nodes with degrees less than k repeatedly by starting with $k = 1$ and increasing k until no nodes are left in the network. The core of the network is the subgraph obtained with the maximum k such that there are still nodes in the subgraph, but with $k + 1$, all nodes are removed. In the case of a weighted network, node strength substitutes node degree, and the

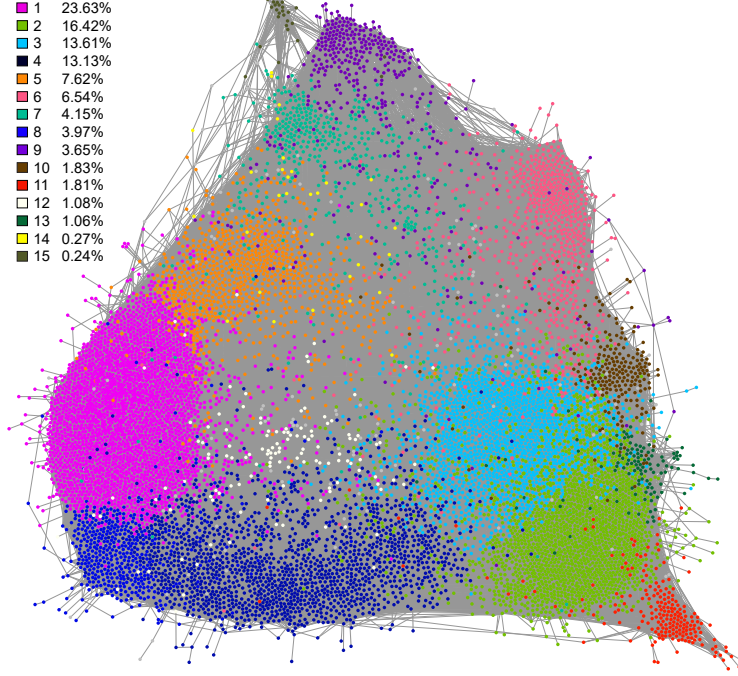


Figure 3.11: The 15 major communities of the LCC in AgGCN1.0. The network was visualized with Gephi 0.9.2 using the ForceAtlas 2.0 algorithm¹³⁷. Communities are numbered consecutively based on the number of nodes they contain and colored according to the panel on the left.

definition of coreness needs to be adapted. In the s -core decomposition, the s -core subgraph consists of all nodes i with node strengths $s(i) > s$, where s is a threshold value. We define the threshold value of the s_n -core as $s_{n-1} = \min s(i)$, among all nodes i belonging to the s_{n-1} -core network. The s_n -core is found by the iterative removal of all nodes with strengths $s(i) \leq s_{n-1}$. Like k -core analysis, where node degrees are recalculated for every removal, the remaining nodes' strengths must also be recalculated in the weighted core¹³⁹.

The distribution of s -core sizes is shown in Fig. 3.12a, while in Fig. 3.12b, the red nodes are the subgraph corresponding to the final s -core. If the threshold was above $s=74.64$, all nodes were removed, showing the nodes within the final s -core are tightly connected. Green nodes remained at a threshold $s=48.54$, corresponding to the first discontinuous drop distribution of s -core sizes (Fig. 3.12a). When $s=69.20$, two separate components were maintained, representing two densely connected parts in the network (blue nodes). After we increased the threshold s , the blue node component in the bottom right disappeared. Note that cores

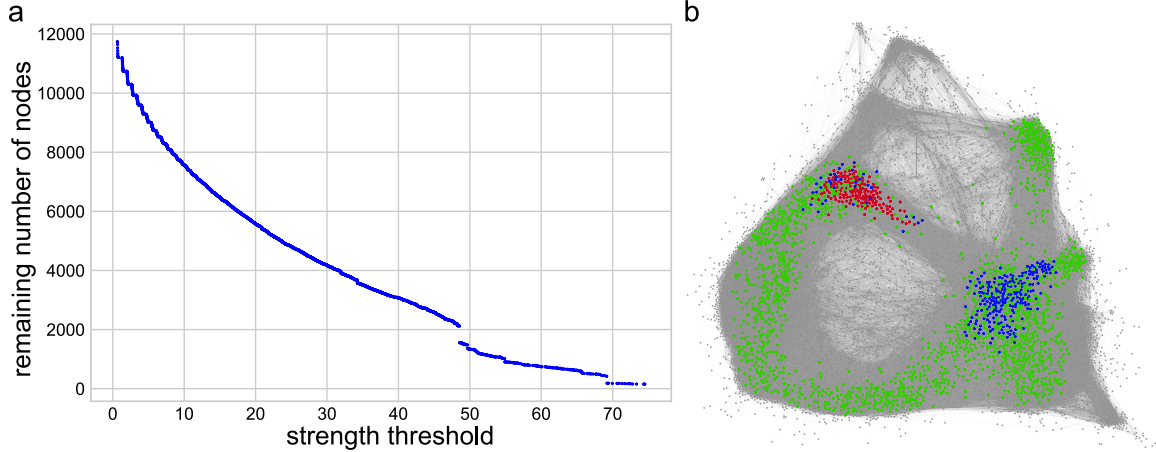


Figure 3.12: Coreness. (a) The number of nodes left with respect to the s -core threshold. (b) Core subgraphs for different strength thresholds. Core $s=48.54$ consists of 2122 nodes (green, blue and red nodes), and core $s=69.20$ consists of 416 nodes (blue and red nodes). The final core $s=74.64$ consists of 153 nodes (red nodes) .

with higher thresholds were included in those with lower thresholds. Nodes in the final core are the ones that remain in the network even when many redundant connections, i.e., many connections to other nodes with equal or smaller strength, were iteratively removed. The final core can be seen as the most critical and internal set of nodes that guarantee network connectivity.

The analysis of node characteristics in the final core revealed the following. Core nodes were, on average, highly expressed across most conditions, as compared to all nodes in the LCC, and also sampled across most conditions (≥ 230). The centrality measures (strength, eigenvector, closeness, and betweenness) of the core nodes were also, on average, significantly higher than those of the LCC. However, the core nodes did not include any of the top central nodes under any of the four centrality measures (strength, eigenvector, closeness, and betweenness). Taken together, these results show that the co-expression of core nodes is supported by strong expression across the majority of conditions in the network. In addition, overall gene regulation across the *An. gambiae* transcriptome results in a network structure that at its core maximizes all centrality measures rather than a specific one for each core node. This structure stabilizes the GCN, because a targeted perturbation of the highest centrality nodes would not affect the network core, thus providing another layer of network

robustness.

3.5 Network architecture is based on biological function

The analyses of the properties of the largest component of the AgGCN1.0 network identified a small-world, scale-free network with a small final core and distinct communities. This suggests coordinated behavior of gene expression across a large number of distinct experimental conditions, independent of any single specific condition. Previous studies have identified the network architecture of GCNs to be based on gene function, where co-expression modules represented sets of genes that function within the same biological processes (e.g. [140;141](#)). To determine whether individual subgraphs of the network were enriched for particular biological processes, we performed a Gene Ontology (GO) and KEGG pathway enrichment analysis in R using topGO [142](#).

3.5.1 The Network core is enriched in genes required for Oxidative Phosphorylation and Translation

The first subgraph we analyzed for GO and KEGG pathway enrichment was the final core $s = 74.64$, which consists of 153 genes, representing 1.3% of genes that comprise the largest connected component. The top GO categories significantly enriched in the s -core group encompass the biological processes of translation and oxidative phosphorylation (Fig. [3.13](#), Table S1). Indeed, the s -core contains 64 of the 131 genes identified to make up the ribosome of *An. gambiae* (KEGG pathway aga03010), and 38 of the 107 genes that comprise oxidative phosphorylation (KEGG pathway aga00190). We next analyzed GO enrichment in the two larger cores, Core $s = 48.54$ and core $s = 69.20$ (Fig. S2 and 3). Similarly to the final Core 74.64, the GO terms enriched the most belonged to the biological processes of mitochondrial electron transport and translation (Table S1). However, the enrichment was largely due

to the presence of genes in the final core, with the larger cores adding less than half of the enriched genes that make up the ribosome and mitochondrial electron transport chain. Given the fundamental need for ATP and proteins for all cellular function, it is perhaps not too surprising that the expression of these genes is most central and integrated across the entire *An. gambiae* transcriptome.

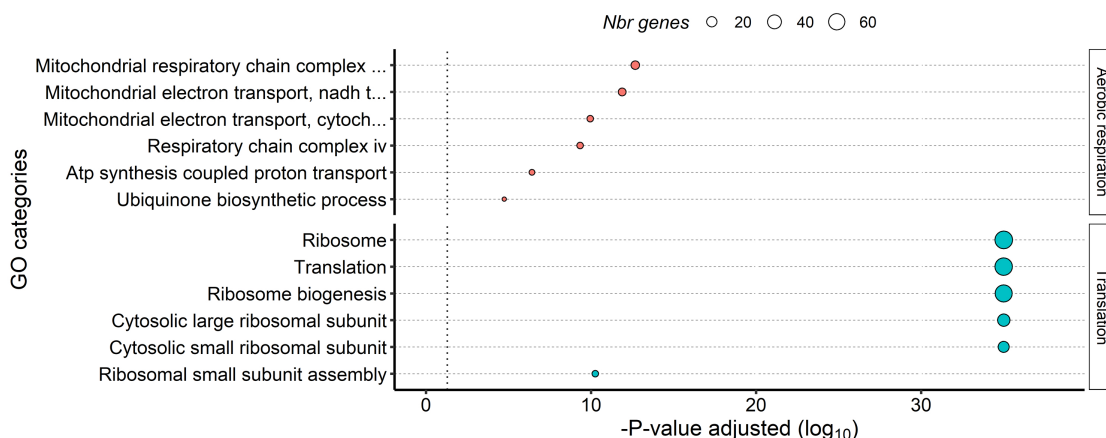


Figure 3.13: The network core is enriched for genes that function in oxidative phosphorylation and translation. The X-axis represents the statistical significance of the gene ontology (GO) categories (y-axis) after adjustment for multiple testing. The size of the dot is proportional to the number of genes in the given GO category.

3.5.2 Network communities are enriched for gene sets with functions in distinct biological processes

We next analyzed the community subgraphs for enrichment of GO terms and KEGG pathways among their annotated genes (results are summarized in Table 3.5, all data in Table S2, Fig. S3-S17). We found each community enriched for distinct biological processes, which dependent on community, ranged from fundamental cell functions to specialized physiologies executed by specific organs or tissues.

Manual analysis of enriched GO and KEGG terms within each community revealed an additional clustering of individual biological processes that together are required for fundamental cellular functions. For example, Community 1 is enriched for genes with GO terms

that broadly fall into the biological processes of DNA repair, Transcription, RNA processing, and Translation, indicating co-expression of genes that belong to the functionally related cellular functions of gene expression (Fig. S3). A second example is Community 5, which contains an overrepresentation of genes with GO term annotations belonging to glycolysis, tricarboxylic acid (TCA) cycle, and Oxidative phosphorylation, and cell redox homeostasis, indicating that the genes required for oxidative energy production are co-regulated (Fig. S7). In addition, Community 5 is also enriched for genes with function in translation and protein processing, suggesting that the generation of proteins and their turnover are co-regulated on a transcriptional level (Fig. S7). A third example is Community 8, which is enriched for genes that encode nuclear proteins that function in DNA replication and repair, Cytokinesis, and Cell cycle.

Table 3.5: *Nodes distribution in each community and GO term enrichment*

Community (% of network)	Enriched Biological Functions
1 (23.6)	DNA repair, Transcription, RNA processing, Translation
2 (16.4)	Neurogenesis/neuronal function, Sensory perception
3 (13.6)	Cuticle metabolism, Cytoskeleton organization, Muscle development
4 (7.62)	Intracellular Signal transduction, Protein phosphorylation, Neuron function
5 (6.54)	Aerobic respiration, Glycolysis/TCA cycle, Redox homeostasis, Translation, Protein processing
6 (4.15)	Neuronal function, Signal transduction, Sensory perception
7 (3.97)	Innate immunity, Lipid metabolism
8 (3.65)	DNA replication/repair, Cell cycle, Cell division, Meiosis/Oogenesis
9 (1.83)	Digestion, Drug metabolism, Chitin metabolism, Transmembrane transport, Innate immunity
10 (1.81)	Chitin metabolism, Lipid metabolism, Proteolysis
11 (1.08)	Cytoskeleton, Axoneme
12 (1.06)	Gluconate shunt, Signaling
13 (1.01)	Cuticle metabolism
14 (0.27)	Proteolysis
15 (0.24)	Salivary gland proteins, Smell*

* GO term “smell” due to enrichment of the D7 family salivary gland protein genes, which are odorant binding proteins.

Community 15 presents the most striking example of a set of co-expressed genes that have a highly specialized function, as it is enriched for genes with known functions in blood feeding

(Fig. S17). About 93% of all genes in Community 15 encode known salivary gland proteins. Indeed, this community contains 51% of all genes identified by Arca et al. as salivary gland protein genes¹⁴³. Furthermore, Community 15 is enriched specifically in salivary gland proteins that are expressed in adult female salivary glands, including the majority of the D7 family, all members of the SG1 family and SG7/SG7-2 family, as well as gVAG (gambiae Venom AllerGen), apyrase and 5'-nucleotidase. In contrast, salivary gland protein genes expressed in male and female salivary glands, including salivary gland amylase, maltase, members of the SG2 protein family, SG9, and the gene encoding the 55.3 kDa salivary gland protein are located in community 9.

Community 7 presents a second example of a set of co-expressed genes that have a highly specialized function, as it is highly enriched for genes encoding proteins with known function in innate immunity (Fig. S. 9). The *An. gambiae* genome encodes 347 canonical immunity genes belonging to the immune modules of recognition, modulation, signal transduction, and effectors¹⁴⁴. Of these, 23.6% (82 genes) are part of community 7, while the entire community only represents 4.1% of the network. Community 7 includes 20.5% of putative recognition genes, 37.9% of modulation genes, and 26.7% of effector genes, but only 1 (TOLL5D) of 53 annotated immune signal transduction genes. The largest immune protein family to be overrepresented is the CLIP-domain containing serine proteases (CLIPs)¹⁴⁵, with 37 of 88 of annotated CLIPs are present in Community 7.

In addition to higher-order clustering of biological processes within communities, we also compared enriched GO terms between communities. We found that in many instances, related biological processes are enriched in neighboring communities. For example, Communities 3, 10, and 13 map to the same region of the 2D visualized network (Fig. 12) and are enriched for genes with GO term annotations related to chitin metabolism. These GO terms are partially explained by the enrichment of Communities 3 and 10 for genes encoding CPR proteins, which are characterized by a Rebers and Riddiford Consensus (RR) domain and are major components of the insect cuticle. CPR proteins fall into two evolutionary subgroups, based on their RR domain type, which are referred to as RR1 and RR2. The *An. gambiae* genome encodes 55 RR1 and 102 RR2 genes¹⁴⁶, of which 47 RR1 and 92 RR2

genes are present in the AgGCN1.0. Community 3 contains 45% of CPR genes, while only containing 13.6% of genes in the network. Community 13 contains 19% of RR1 genes, while only containing 1.0% of genes in the network.

Other examples are of shared GO terms across neighboring communities are the enrichment of innate immunity genes in communities 7 and 9, as well as the enrichment of salivary gland protein genes in communities 15 and 9 (Fig. 3.13). The algorithm of the Force Atlas 2.0 network layout pulls together nodes that are connected by links, while repelling nodes¹³⁷, thus the expression of genes within neighboring communities is likely more similar than to communities that are more distant to each other.

3.6 Summary

In this chapter, we constructed a global gene co-expression network for *An. gambiae* based on the meta-analysis of 30 gene expression studies. The rich information produced by different experiments made it challenging with existing methodologies to analyze the relations of genes based on co-expression, as different methodologies were used to process the data. Current approaches cannot directly construct a GCN from many conditions that are tested with various methodologies. The raw expression values of each condition, therefore, required normalization before applying any network construction methods. In this work, we adopted the z-score normalization, by which the expression values in different conditions are normalized with zero mean and unity variance. The co-expression of genes was then quantified by the Pearson correlation coefficient, which is computed for each pair of genes based on the normalized expression. However, the number of paired elements between genes was heterogeneously distributed, given that only a subset of genes was tested under each condition, and missing values are ubiquitous in experimental data. Thus, a unique threshold or criterion was not appropriate for selecting edges. Instead, we categorized the PCCs into different intervals according to the number of paired elements, and the fitted sliding threshold was used to select edges for the GCN. In addition, the PCCs were rescaled by the reversed curve of the fitted sliding thresholds to obtain the edge weights.

Analyses of the resulting AgGCN1.0 network showed that it is robust with respect to random removal of up to 15% conditions. In addition, the sensitivity of the AgGCN1.0 structure to the underlying number of conditions mostly affected the loss of those edges derived from a few experimental data. Studies of the topological properties of the network showed that AgGCN1.0 is dominated by hub nodes and approximates a scale-free property. Scale-free properties are typical for real-world networks^{147;148}, including GCNs and other biological networks^{34;135}. This property likely provides network robustness and protects against random perturbations, e.g., mutations that protect the global architecture of the network. In addition, the global architecture is also protected against targeted perturbations, due to the node properties of the AgGCN1.0 network core, where all centrality measures are maximized rather than a specific one for each core node. It will be interesting to compare whether this feature is common of global GCNs, and whether it is a consequence of specific biological gene properties or functions.

Scale-free and small-world networks are characterized by the existence of communities, which are groups of nodes that are well-connected inside and loosely connected with other communities. This translates to GCNs, in which genes are often grouped into modules that are characterized by a similar function. The 15 communities of the AgGCN1.0 are indeed enriched for different GO and KEGG terms, thus constituting biologically meaningful groups. This finding provides additional validation of the updated GCN construction methodology reported in this study. Not surprisingly, many communities were enriched functions that constitute fundamental biological processes required for cellular function regardless of cell or tissue type. This is largely explained by the non-model organism status of mosquitoes. With limited functional characterization of lineage-specific genes, gene annotation often relies on orthology and thus biasing even further the inherent incompleteness of gene ontology¹⁴⁹. While mosquito-specific GO terms have been defined based on anatomy¹⁵⁰, several physiologies highly relevant to mosquito biology are either not included in the GO database (e.g., hematophagy) or are underutilized in annotations (e.g., host-seeking). Nevertheless, community structure in the AgGCN1.0 is clearly defined by functional clustering of genes.

The GO term and KEGG pathway enrichment analysis also revealed an interesting pat-

tern that may extend beyond the AgGCN1.0 to other gene co-expression networks. The architecture of the AgGCN1.0 at its *s*-core is defined by genes required for respiration and translation, both fundamental processes required for survival at the cellular level. The expression of these genes is further integrated into the expression of genes required for aerobic energy production and cellular protein genesis and homeostasis. Not surprisingly, these genes are well-conserved evolutionarily at the metazoa and arthropoda level¹⁵¹. This not only demonstrates integration of these processes at the transcriptional level, but also perhaps that the transcriptional regulation of these genes evolved early and has been maintained throughout evolution. In contrast, communities at the periphery of the network (e.g., communities 11 and 15) are enriched in genes that contribute to specialized biological processes, including blood feeding¹⁴³ and potentially sensory neuron and/or sperm function¹⁵². Genes in these communities also tended to be more lineage-specific, suggesting that integration of novel biological processes into the *An. gambiae* transcriptome may not require rewiring of the regulatory circuitry. In the future, it will be interesting to determine whether gene age is a principle that governs global gene co-expression patterns in *An. gambiae* and beyond.

In summary, this manuscript provides a correlation-based methodology to build GCNs from highly heterogeneous expression data. This methodology was then applied successfully to build a robust global GCN for *An. gambiae*, updating the previous meta-analysis performed by². This network is available freely to the scientific community at <https://github.com/KSUNetSE/AgGCN1.0>. Analysis of the AgGCN1.0 LCC suggests that the architecture of the *An. gambiae* transcriptome maximizes the integration of essential cellular processes and enables evolutionary flexibility to integrate the expression of novel biological functions.

Chapter 4

A software package for gene co-expression network construction and analysis

4.1 Background

In Chapter. 3, we developed a data processing scheme to construct a gene co-expression network for *An. gambiae*. Our experimental results demonstrated the effectiveness of this approach in revealing gene functions and patterns, even when dealing with different experimental technologies and a large number of missing values. In this chapter, we developed an integrated tool–*GeCoNet-Tool* for gene co-expression network construction and analysis. This tool streamlines the process of network construction and provides a user-friendly interface for network analysis.

4.2 Implementation

The *GeCoNet-Tool* is an open-source software package that combines gene co-expression network construction and analysis. It is an improved implementation of our previously

published work⁷⁴, which presents a scheme to study gene expression data across a large number of conditions. The tool is an executable file written in Python and features a user-friendly graphical interface that simplifies configuration for various data types (as illustrated in Fig. 4.1). The use of *GeCoNet-Tool* can be divided into two independent parts: network construction and network analysis.

4.2.1 Data processing and network generation

To generate a gene co-expression network edge list using *GeCoNet-Tool*, the user needs to provide a gene co-expression matrix in .csv format. This matrix should have N rows representing genes and M columns representing experimental conditions.

The tool offers various options for processing the input data based on the data type. For example, if the input data is obtained through RNA-seq¹⁵³, users can choose to remove zeros, re-scale expression values by log2, or normalize columns by z-score. Additionally, users can choose to save the processed data in .csv format. These options allow users to customize the data processing step to suit their specific needs and ensure accurate network construction.

The Pearson Correlation Coefficient (PCC) is calculated between every pair of genes based on the processed data using *GeCoNet-Tool*. If the user chooses to save the PCC matrix, it is stored as an upper triangular matrix^{4;114}. In our previous work⁷⁴, we found that the number of experimental conditions could significantly affect the PCC of two genes. Therefore, *GeCoNet-Tool* determines the number of paired elements between every pair of genes, and the user can save this data as an additional upper triangular matrix.

GeCoNet-Tool uses a classification system to categorize PCC values into different intervals based on the number of paired conditions for each gene pair. The user can specify the size of the intervals, referred to as the *bin size*, in the tool’s interface. To select edges based on PCC value, the user also needs to input the cutoff value expressed as the chosen top percentage of all PCCs in a given interval (e.g., 0.5%, 1%, or 2%). This cutoff value is used

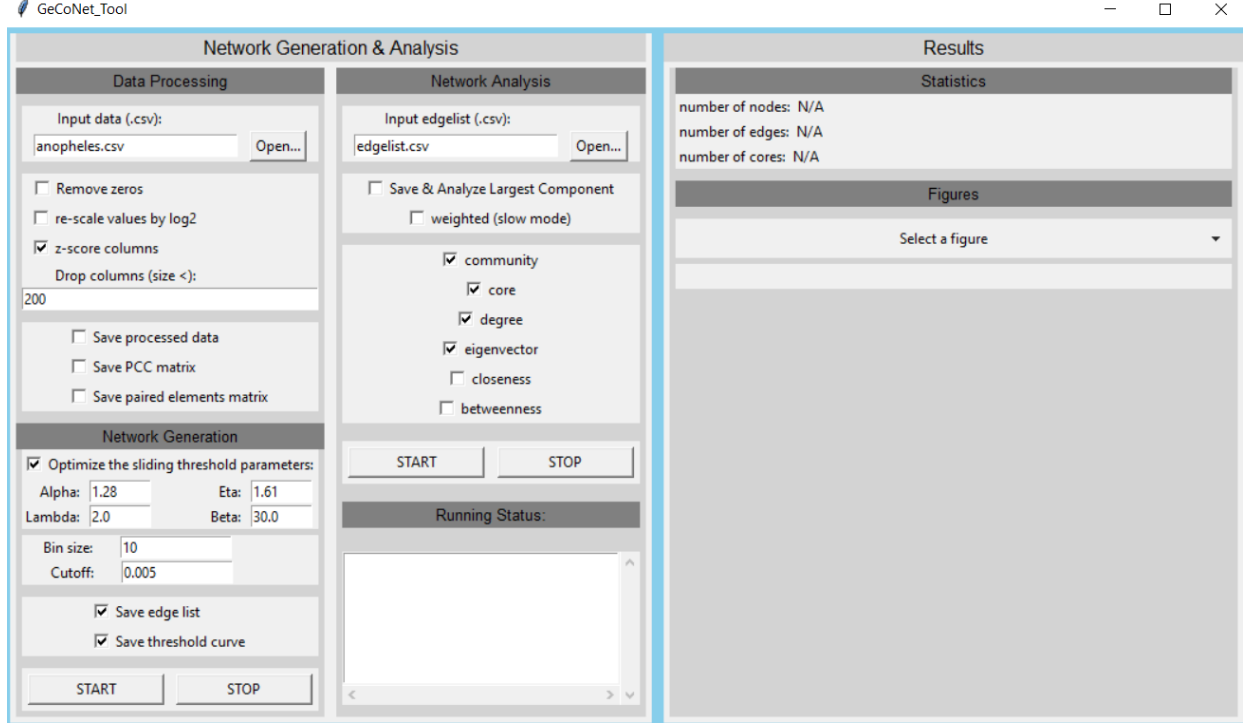


Figure 4.1: *The GeCoNet-Tool user interface.*

to determine the sliding threshold by fitting a curve to the PCC values:

$$f^{thres}(x) = \alpha - \frac{1}{\eta + \lambda e^{-\frac{x}{\beta}}} \quad (4.1)$$

where α , η , λ , and β are the four parameters that are estimated through the fitting process, and x represents the number of paired elements. This equation strikes a balance between fitting accuracy and the number of parameters estimated⁷⁴. Once the curve is fitted, the optimal values of the four parameters are updated and displayed in the input boxes labeled α , η , λ , and β in the tool's interface.

To optimize the four parameters for the fitted curve, *GeCoNet-Tool* uses an optimization algorithm instead of manual optimization, as in our previous work⁷⁴. The coefficient of determination (R-squared) for the fitted curve is displayed in the *Running Status* box to give users an indication of the goodness-of-fit for the curve. The edges of the co-expression network are selected through the fitted curve and based on the number of paired elements⁷⁴. The user can construct networks with different edge densities by adjusting the cutoff value.

This allows users to explore different co-expression network topologies and focus on specific genes or gene clusters of interest.

Finally, the user can choose to save the edge list and fitted threshold curve for further analysis. The edge list of the co-expression network and the threshold curve will be saved to the same folder as the input data. This enables the user to perform downstream analysis, such as centrality, gene ontology, and pathway enrichment analysis.

4.2.2 Network analysis

Once a gene co-expression network is constructed using *GeCoNet-Tool*, various properties of the network can be analyzed through the second part of the tool. *GeCoNet-Tool* enables users to analyze the following network properties:

- community: The community is defined as a subgraph that is highly connected internally and loosely connected to other subgraphs. *GeCoNet-Tool* detects the communities of the network through the Louvain algorithm¹⁰⁸.
- core: The core of the network is obtained by repeatedly removing nodes with a degree less than k by starting with $k = 1$ and increasing k until no nodes are left in the network¹³⁹.
- degree: *GeCoNet-Tool* calculates node degree if the network is unweighted and calculates node strength if the network is weighted¹.
- eigenvector: *GeCoNet-Tool* calculates the eigenvector centrality, which is determined by the entry of the eigenvector corresponding to the largest eigenvalue of the adjacency matrix of the network¹⁵⁴.
- betweenness: *GeCoNet-Tool* calculates the betweenness centrality, determined by the number of shortest paths that pass through the node itself¹⁵⁵.
- closeness: *GeCoNet-Tool* calculates the closeness centrality, which is based on the distances between nodes. Closeness centrality is the sum of the shortest path distance

reciprocals of a node to all other nodes¹⁵⁶.

In the package, the user can choose to analyze the entire network or only the largest connected component, and use unweighted or weighted edges for the analysis. The output of *GeCoNet-Tool* includes a table containing all properties that the user has selected, as well as figures of the node degree distribution, community distribution, and core distribution. These figures help the user to better understand the topological characteristics of the network.

4.3 Results

To keep the user informed, *GeCoNet-Tool* displays the running status in the *Running Status* window, which shows the progress of the current task. Meanwhile, network statistics, such as the number of nodes, edges, and core nodes, will be displayed in the *Results* window as soon as the network is constructed or analyzed. This allows the user to monitor the progress of their analysis and quickly obtain important network properties.

In the experiment, we used *An. gambiae* gene expression data and constructed a gene co-expression network using the default settings in *GeCoNet-Tool*. The gene expression data is publicly available through VectorBase (www.vectorbase.org) at the following URL: <https://tinyurl.com/mr38a7hj>. Figure 4.2(a) shows the sliding threshold curve with a cutoff value of 0.005, and the R-squared value is 0.908, indicating that the curve fits the raw data well.

In the second part, users can analyze the constructed network and produce a table to store the properties of the network. In the *An. gambiae* gene co-expression network, there are 12,660 nodes, 381,760 edges, and 164 core nodes. *GeCoNet-Tool* employs the force-directed algorithm Fruchterman-Reingold layout¹⁵⁷ to visualize the community distribution (Fig. 4.2(b)) and core nodes (Fig. 4.2(c)). However, the layout shown in the results window is not interactive. We recommend users to use interactive network visualization algorithms such as Gephi¹³⁷ to show the generated network and its properties. *GeCoNet-Tool* also generates a node degree distribution as shown in Fig. 4.2(d).

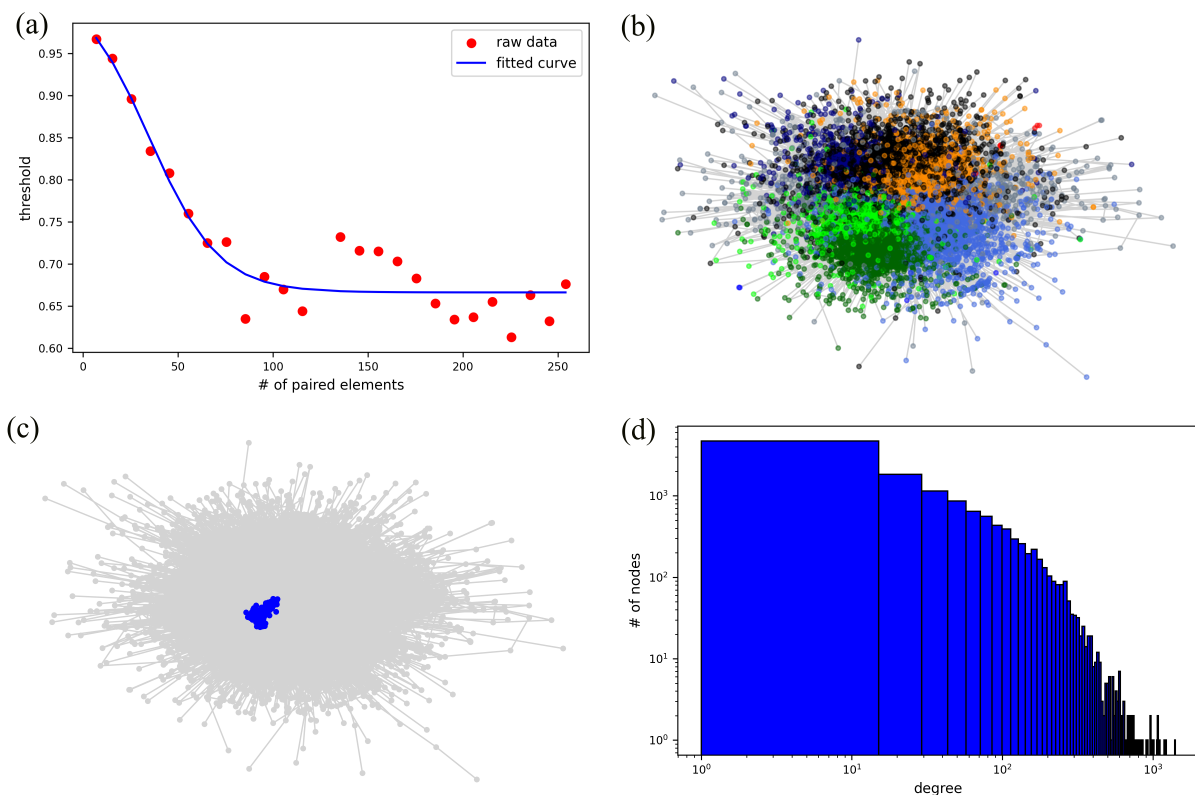


Figure 4.2: Examples of figure outputs from GeCoNet-Tool using *An. gambiae* expression data. (a) The fitted threshold curve, (b) communities, (c) core nodes, and (d) degree distribution of the *An. gambiae* gene co-expression network.

4.4 Summary

GeCoNet-Tool is a free and user-friendly research tool that does not require coding knowledge. The package comprises two parts: 1) network construction and 2) network analysis. In the first part, the relationship between every pair of nodes is evaluated by PCC and by the number of paired conditions. *GeCoNet-Tool* allows users to input different expression data types, such as RNA-seq and microarray. The user can choose how the data is processed, including removing zeros and log2-rescaling. In the second part, *GeCoNet-Tool* provides multiple tools for network analysis. For example, the community analysis will classify nodes into different communities to identify genes with similar biological functions.

Chapter 5

Learning node features and network structure from noisy node activity data¹

5.1 Background

A network is a system-level view of pairwise interactions between nodes, genes, or elements in a complex system^{5;21;76;79;82;87;89;90;97;158–160}. To analyze a networked system, the first step is to construct the network from data obtained through different technologies. In most cases, network structures can be determined through direct measurements where pairwise relationships between nodes can be observed directly. For instance, friendship networks can be probed through various ways including using questionnaires, checking Facebook or Twitter friendship, and investigating face-to-face interactions. Similarly, edges in web graphs can be directly determined by checking if hyperlinks exist between web pages^{3;15;17;80;161}. However, there are cases where the relationships between nodes cannot be observed directly². Instead, we may only have node activity data that reflect the properties of nodes from various aspects. In these cases, we need to estimate the underlying network structure from nodal data. Such a

¹This chapter is a slightly modified version of a published article⁷⁶.

problem exists in many areas, including the construction of financial, biological, and climate networks^{19;20;22-31}. In these areas, measurements of pairwise relationships are not always feasible^{2;3;159}. Instead, we can conduct various experiments to measure node activities under different conditions¹⁶².

This chapter explores whether we can use neural networks to learn node representations from noisy and heterogeneous data and tries to use an entropy-based method to extract network structures. Specifically, we investigate the problems of feature learning and network construction for gene co-expression data. Different high throughput technologies, including microarray and RNA-sequencing, allow simultaneous evaluation of thousands of gene expression data. Usually, the data can be organized into a matrix that consists of rows representing N genes (nodes) and columns representing M experimental conditions. To construct a network from such expression data, we need to consider three problems. (1) The expression data are often distributed over various ranges, as they are measured using different experimental technologies. This can result in values that are dispersed from zero to tens of thousands, and which do not follow any specific distribution. (2) Missing values are frequently present in the dataset. This can occur because some experiments only test a subset of genes for specific purposes or because some experimental data for certain genes (nodes) are not available. (3) The levels of noise are not constant, as environmental factors such as humidity, temperature, and light intensity can potentially influence the accuracy of the devices and the measured node activity data. To address these challenges, the method for network construction is expected to be slightly or not affected by missing values and noise.

There are diverse approaches aiming at constructing networks from nodal data. A significant volume of works uses the correlation coefficient to measure the degree to which a pair of nodes is related, and edges are selected by thresholding the correlation coefficients^{31;163;164}. In Chapter. 3, we demonstrated that Pearson Correlation Coefficient is very effective in network construction. However, the drawbacks of the correlation methods are that: 1) the expression data are required to follow a (quasi-) normal distribution, 2) the correlation coefficients are significantly affected by outliers, and 3) the correlation coefficient does not have any bearings on the Euclidean distance of two vectors^{163;164}. Mutual Information (MI) and its variants

are also used to construct gene co-expression networks. The MI models do not require the data to follow the normal distribution. Still, the MI models are even more complex, since we are expected to find the joint probability distribution for every pair of genes¹⁶⁵. Some researchers have proposed using rescaling and normalization methodologies to obtain quasi-normal distributed data from the raw node activity data¹⁶⁶. According to¹¹⁷, the number of experimental conditions significantly influences the correlation coefficients under the null hypothesis that two nodes are not correlated. Theoretical analysis shows that correlations based on ten conditions tend to be higher than those computed with 50 conditions. Missing values lead to node pairs with a different number of paired elements, meaning that the node pairs with fewer paired elements are more likely to have high correlation coefficients. Therefore, some works use imputation, interpolation or sliding threshold to solve the problem (2)^{74;167}. The complex data processing procedures pose a severe challenge to the principle of parsimony when we further study the resulting network structure¹⁶⁸.

Edge selection is another issue we need to consider when constructing networks from node activity data. Both correlation and MI methods return coefficients between -1 and 1. Many researchers construct unweighted networks by applying a threshold to select edges of the network corresponding to node pairs with the highest coefficients. However, choosing a threshold is always tricky since a high threshold could generate singleton nodes, while a low threshold generates networks with many weakly connected node pairs^{83;166}. Though the problem can be solved by fixing the minimum number of neighbors of each node, the choice of the threshold influences the node degree distribution, meaning that nodes' roles in the resulting network are related to the choice of thresholds. As an alternative, we propose an entropy-based network construction method, which has better performance in maintaining nodes' roles (e.g., hubs and leaf nodes) and avoiding isolating nodes.

This chapter presents a novel approach for constructing networks from noisy node activity data, which involves a neural network-based method to extract node representations and an entropy-based approach to extract network structure. Inspired by the application of neural networks in natural language processing (NLP)^{169–174}, we propose generating node sequences from node activity data to simulate sentences in documents. The neural network model can

embed node sequences into vectors of identical dimensions, which allows us to study node features and construct networks. The main contributions of the chapter are as follows: Firstly, it offers a simple and direct data processing scheme that generates random node sequences from M conditions, without the need for any specific data distribution, re-scaling or normalization. Moreover, the M conditions are processed separately, minimizing negative impacts from missing data and outliers. Secondly, the node sequences are trained with a neural network model that builds on the hypothesis that nodes with similar properties tend to have similar neighbors, resulting in similar nodes having similarly trained node vectors¹⁷⁴. Thirdly, an entropy-based method is proposed to construct the corresponding network where selected edges can recover node roles. Finally, the proposed approach is validated experimentally using both synthetic and real data^{105;175}. Overall, this approach presents a promising solution for constructing networks from noisy node activity data, which can provide insights into the structure and function of complex systems.

5.2 Feature vectors from node activity data

This section introduces the key components of our proposed approach, including the definition of the context set, the node sequence generation process, and the entropy-based method for network construction.

In language, words with similar meanings tend to appear in similar contexts¹⁶⁹. Similarly, nodes with similar properties in different conditions tend to appear in similar neighborhoods. By leveraging natural language processing (NLP) techniques, we can learn node representations from node sequences that capture the local structural and dynamic features of the nodes. The measurements of nodes in different conditions represent different properties, similar to words in various topics that may have different meanings. Building on these observations, we design a scalable node sequence generation strategy to process the M conditions separately.

5.2.1 Generate context sets from node activity data

Suppose we have N nodes that are measured in M conditions. Let v_i ($i \leq N$) denote a node, and let $v_i(\omega)$ denote its value in the ω th condition. To generate node sequences, we define the context set of node v_i in the ω th condition as follows:

$$C_\omega(v_i) = \{v_j : |v_j(\omega) - v_i(\omega)| \leq \delta_i^\omega\}. \quad (5.1)$$

where the tolerance δ_i^ω can be a parameter such that $\delta_i^\omega = \beta_\omega v_i(\omega)$. By employing the parameter β_ω , we can tune the size of the context set per the error levels of different technologies. In practice, when a missing value is present in the ω th condition for node v_i , we skip the generation of $C_\omega(v_i)$ and do not predict the missing values from other conditions. This approach helps to minimize the impact of missing data and outliers on the node sequences.

Formally, the context set of node v_i is composed of nodes with measurements falling in the range $[v_i(\omega) - \delta_i^\omega, v_i(\omega) + \delta_i^\omega]$. Therefore, the number of elements of the intersection set $C_\omega(v_i) \cap C_\omega(v_k)$ is related to the measurements of the two nodes v_i and v_k . For example, assume the measurements of the three nodes v_x , v_y , and v_z in the ω th condition are respectively $v_x(\omega) = 1000$, $v_y(\omega) = 990$, and $v_z(\omega) = 950$. It is clear that $v_y(\omega)$ is closer to $v_x(\omega)$ than $v_z(\omega)$, i.e.,

$$|v_x(\omega) - v_y(\omega)| < |v_x(\omega) - v_z(\omega)|. \quad (5.2)$$

Therefore, we have the following inequality:

$$|C_\omega(v_x) \cap C_\omega(v_y)| \geq |C_\omega(v_x) \cap C_\omega(v_z)|, \quad (5.3)$$

where $|\cdot|$ denotes the cardinality of the intersection set. The context set $C_\omega(v_y)$ recapitulates more elements of $C_\omega(v_x)$ than the context set $C_\omega(v_z)$. For any node $v_j \in C_\omega(v_x)$, we have

the probability

$$P(v_j \in C_\omega(v_y)) \geq P(v_j \in C_\omega(v_z)). \quad (5.4)$$

Furthermore, we assume $G_\omega(v_i)$ is a set consisting of nodes whose context set contains node v_i , such that

$$G_\omega(v_i) = \{v_j : v_i \in C_\omega(v_j)\}. \quad (5.5)$$

Based on the same example above, we can say that there are more context sets containing simultaneously v_x and v_y than containing simultaneously v_x and v_z . Therefore, we have

$$|G_\omega(v_x) \cap G_\omega(v_y)| \geq |G_\omega(v_x) \cap G_\omega(v_z)|, \quad (5.6)$$

meaning that the nodes with closer values are more likely to be present in the same context sets. Similarly, for any node $v_j \in G_\omega(v_x)$, we have the probability

$$P(v_j \in G_\omega(v_y)) \geq P(v_j \in G_\omega(v_z)). \quad (5.7)$$

In the generation of node sequences, we always sample the subsequent node from the context set of the current node. For example, given a node sequence l , suppose the i th node is v_x , i.e., $l_i = v_x$. Then, we have a node sequence

$$\{\dots, l_{i-1} \in G_\omega(v_x), l_i = v_x, l_{i+1} \in C_\omega(v_x), \dots\}. \quad (5.8)$$

Based on Eqs. 5.4 and 5.7, l_{i+1} is more likely to be in $C_\omega(v_y)$ than in $C_\omega(v_z)$, and l_{i-1} is more likely to be in $G_\omega(v_y)$ than in $G_\omega(v_z)$. This implies that the context nodes of v_x tend to be the context nodes of v_y rather than of v_z , since $v_y(\omega)$ is closer to $v_x(\omega)$ than $v_z(\omega)$. Thus, in the generated node sequences, nodes with closer measurements tend to appear in similar contexts.

5.2.2 Generate random node sequences

The simplest approach to generating node sequences from context sets is to randomly sample the next node from the context set of the current node. This approach corresponds to a first-order Markov chain¹⁷⁶. Specifically, assuming that the i th node in a node sequence is denoted as l_i , the next node $l_{i+1} \in C_\omega(l_i)$ is selected with a probability given by:

$$p(l_{i+1} \mid l_i) = \frac{1}{|C_\omega(l_i)|}. \quad (5.9)$$

Under this assumption, the nodes in the context set $C_\omega(l_i)$ have an equal probability of being chosen as the subsequent node.

Alternatively, we can generate biased random node sequences. Suppose we are currently at node l_i , having just traversed node l_{i-1} . In this case, the probability of sampling the next node l_{i+1} can be biased by the previous node l_{i-1} . To introduce this bias, we use a parameter ρ , and the unnormalized probability of the next node is defined as follows:

$$p(l_{i+1} \mid l_i, l_{i-1}) = \begin{cases} 1 & \text{if } l_{i+1} \in C_\omega(l_{i-1}) \cup \{l_{i-1}\} \\ \rho & \text{else,} \end{cases} \quad (5.10)$$

where $l_{i+1} \in C_\omega(l_i)$. The sampling strategy resembles a second-order Markov chain¹⁷⁶, where the probability of adding the next node is influenced not only by the current node but also by the previous node. The parameter ρ is introduced to bias the probability of the next node given the previous node. A low value of ρ increases the likelihood of sampling an element from $C_\omega(l_{i-1})$, while a high value of ρ increases the probability of exploring a node further away from l_{i-1} . Higher values of ρ allow sampling a node in $C_\omega(l_i)$ but not in $C_\omega(l_{i-1})$. When $\rho = 1$, Eq.5.10 becomes equivalent to Eq.5.9.

In the ω th condition, we generate K random node sequences starting from each node. Repeating the process for all the M conditions, we obtain a corpus T containing $KNM - KZ$ node sequences, where Z represents the number of missing values.

The goal of generating random node sequences is to feed the corpus T to a three-layer

neural network to obtain node vectors^{172–174;177;178}.

5.2.3 The skip-gram model

The objective of generating random node sequences is to provide the corpus T as input to neural networks for training node vectors. We adopt a simple three-layer skip-gram model, illustrated in Fig. 5.1, as our neural network architecture^{172–174;177;178}. The model consists of an input layer, a hidden layer, and an output layer. In this work, we aim to obtain d -dimensional vectors for each of the N nodes.

In our assumption, nodes with similar values tend to appear in similar contexts. Given a neighborhood H consisting of $2c$ nodes, we denote $P(v_x | H)$ as the conditional probability of node v_x neighboring the $2c$ nodes in H . Based on Bayes' theorem, we can write:

$$P(v_x | H) = \frac{P(H | v_x)P(v_x)}{P(H)}, \quad (5.11)$$

where $P(H)$ and $P(v_x)$ are respectively the probability of H and v_x , and $P(H)$ and $P(v_x)$ can be regarded as constants. Then, we have

$$P(v_x | H) \propto P(H | v_x). \quad (5.12)$$

Now, let us consider one of the node sequences from the corpus. Let l_i denote the i th node of the sequence, and $H = l_{i-c}, \dots, l_{i-1}, l_{i+1}, \dots, l_{i+c}$. That is, we have an outcome H given l_i . Since the goal is to determine $f(l_i)$, we can replace v_x in Eq. 5.12 with $f(l_i)$, and assuming the $2c$ nodes are independent¹⁷⁹, we have

$$P(f(l_i) | H) \propto P(H | f(l_i)) = \prod_{-c \leq j \leq c, j \neq 0} P(l_{i+j} | f(l_i)), \quad (5.13)$$

where $P(l_{i+j} | f(l_i))$ is the occurring probability of node l_{i+j} given the vector $f(l_i)$. To

determine $f(l_i)$, we have the optimization problem after taking the log form of Eq. 5.13:

$$E = -\min_f \sum_{-c \leq j \leq c, j \neq 0} \log P(l_{i+j} | f(l_i)). \quad (5.14)$$

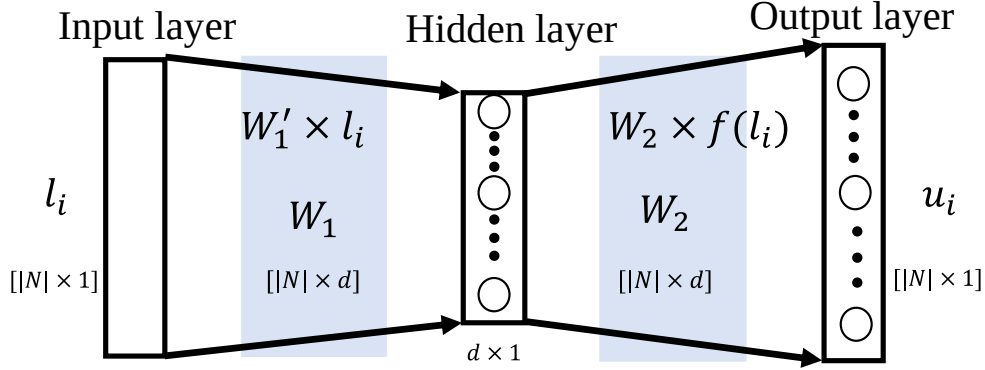


Figure 5.1: The three-layer neural network model (The skip-gram model). Each input node is associated with an N -dimensional one-hot vector¹⁷², which is mapped to the node vector $f(v_i)$ (the hidden layer) of dimension d by matrix W_1 . The hidden layer is mapped to the output vector by matrix W_2 . The elements of W_1 and W_2 are initialized with random values, which are expected to be optimized by backpropagation¹⁷³.

The model projects the node vector $f(l_i)$ to an N -dimensional output vector u_i , as shown in Fig. 5.1. Each dimension of u_i is associated with a node in the corpus, and we use the softmax function^{173;174;180} to convert the entries of u_i into probabilities, which form a probability distribution. For example, the probability of the r th entry of u_i given $f(l_i)$ is

$$P(v_r | f(l_i)) = \frac{\exp(u_i^r)}{\sum_{r' \in N} \exp(u_i^{r'})}, \quad (5.15)$$

where u_i^r is r th entry of u_i and $P(v_r | f(l_i))$ is the probability of node v_r to be the context of l_i . According to Eq. 5.15, nodes in H have higher probabilities of being the context of node l_i . Combining Eq. 5.14 and Eq. 5.15, we have the loss function^{181;182}

$$E = -\min_f \sum_{-c \leq j \leq c, j \neq 0} u_{i+j} + \log \left(\sum_{r' \in N} \exp(u_i^{r'}) \right). \quad (5.16)$$

which is applied to every node in the sequence. Eq. 5.16 is optimized by using the stochastic gradient descent approach^{169;173;174}, which backpropagates¹⁸⁰ errors to update the elements of the matrices W_1 and W_2 in Fig. 5.1.

The method we have introduced falls in the category of semi-supervised learning, in which we learn node vectors from node activity data. The node vectors can be used to extract networks or detect nodes with similar properties.

5.3 Network structure from trained node vectors

After training the neural network model, we obtain N vectors for the N nodes. With these node vectors, we can predict relationships between the nodes, visualize the global structures of the nodes, and construct a corresponding network.

A common approach to selecting edges is to globally threshold the cosine similarities to filter out weak links and obtain a backbone of the underlying network. This method, known as globally thresholding edges (GTE), is widely used in determining gene co-expression networks¹⁶⁴. However, the drawback of GTE is that some nodes may become isolated from the network if the threshold is too high. Although we can force isolated nodes to connect to other nodes, the degree distribution of the resulting network is still affected by the choice of threshold, meaning that the roles of nodes in the network may be sensitive to this choice. To avoid these issues, we propose a Rényi entropy-based method (REM) to extract a network from the trained node vectors^{175;183–187}.

Once we have the node vectors, we can compute the cosine similarity to quantify the connection strength for each pair of nodes. Here, we define $S^0(v_i)$ as the initial neighbor set of node v_i . $S^0(v_i)$ is composed of nodes that are positively similar to v_i , i.e., $S^0(v_i) = \{v_j : s(v_i, v_j) > 0\}$, where $s(v_i, v_j)$ is the cosine similarity between v_i and v_j . However, the network constructed from $S^0(v_i)$, $\forall i < N$ is not helpful in real applications because most node pairs are weakly connected.

Inspired by the application of entropy in ecology, we regard the nodes in the set $S^0(v_i)$ are the states of the system v_i . Then, we associate each state with a probability, which is

computed from the similarity values, such that:

$$\bar{s}_i^0(v_j) = \frac{s(v_i, v_j)}{\sum_{v_j \in S^0(v_i)} s(v_i, v_j)}. \quad (5.17)$$

In information theory, entropy depicts the diversity and randomness of a system²¹. The Rényi entropy for node v_i with order α is

$$H_\alpha^1(v_i) = \frac{1}{1-\alpha} \ln \sum_{v_j \in S^0(v_i)} (\bar{s}_i^0(v_j))^\alpha, \quad (5.18)$$

where $\alpha > 0$.

Note that the Rényi entropy converges to the Shannon entropy as $\alpha \rightarrow 1$, i.e., $H_\alpha^1(v_i) = \sum_{v_j \in S^0(v_i)} \bar{s}_i^0(v_j) \log \bar{s}_i^0(v_j)$. For any α , the entropy $H_\alpha^1(v_i)$ ranges from zero to $\ln |S^0(v_i)|$. In the case of a certain event, i.e., $\exists v_j \in S^0(v_i)$, where $\bar{s}_i^0(v_j) = 1$ and $H_\alpha(v_i) = 0$. Conversely, the entropy $H_\alpha^1(v_i) = \ln |S^0(v_i)|$ when $\bar{s}_i^0(v_j)$ follows a uniform distribution. The diversity index $D_\alpha^1(v_i)$ is

$$D_\alpha^1(v_i) = \exp(H_\alpha^1(v_i)) = \left(\sum_{v_j \in S^0(v_i)} (\bar{s}_i^0(v_j))^\alpha \right)^{1/(1-\alpha)}, \quad (5.19)$$

which is also known as the Hill numbers¹⁸³. It is unsurprising that $\bar{s}_i^0(v_j)$ is not uniformly distributed, and $H_\alpha^1(v_i) \in [0, \ln |S^0(v_i)|]$. In ecology, the diversity index quantifies the abundance of species in a community. The diversity index approaches the total number of species when the species are equally abundant and approaches one if there is a dominant species. In Eq. 5.19, the order α influences the sensitivity of the diversity index. Increasing α strengthens the weights of the most abundant species. That is, higher α allows us to select the more abundant species, while lower α will detect more species. Therefore, we can use α to control the number of neighbors of node v_i .

We pick the top $\lfloor D_\alpha^1(v_i) \rfloor$ nodes with the highest similarities from $S^0(v_i)$ to be the effective number of neighbors of node v_i . These selected neighbors form a new neighbor set, $S^1(v_i)$,

which are more strongly connected to v_i than the nodes in $S^0(v_i)$. The resulting network constructed from $S^1(v_i), \forall i \leq N$ is denser than that constructed from $S^0(v_i), \forall i \leq N$. We can then iteratively apply Eqs. 5.17, 5.18, and 5.19 to obtain a network with desired edge density. Let $\lfloor D_\alpha^k(v_i) \rfloor$ be the diversity index at the k th iteration, then we can express the process as:

$$S^k(v_i) = \{v_j \in S^{k-1}(v_i) : |\{v_z \in S^{k-1}(v_i) : s(v_i, v_j) < s(v_i, v_z)\}| < \lfloor D_\alpha^k(v_i) \rfloor\}, \quad (5.20)$$

where $k \geq 1$. In each iteration, top $\lfloor D_\alpha^k(v_i) \rfloor$ nodes with the highest similarity values are selected as the neighbors of v_i . By repeating this process, the REM effectively filters out weak links for v_i , leaving only the most meaningful neighbors in the set $S^k(v_i)$. As a result, the network constructed from $S^k(v_i)$ becomes progressively denser and more informative about the connectivity patterns of the nodes.

In real-world networks, nodes can be classified into two categories: leaf nodes and hubs. Leaf nodes are those connected to a small number of others, while hubs have many neighbors. Considering the property of entropy^{184;187}, the size of the resulting neighbor set $S^k(v_i)$ is relatively small if the similarity value distribution of $S^0(v_i)$ is right-skewed. On the contrary, the size of $S^k(v_i)$ is much larger if the similarity value distribution of $S^0(v_i)$ is left-skewed^{175;183;185}. This indicates that the role of node v_i in the network is closely related to the distribution of similarity values.

5.4 Results of Feature learning

The method we have presented falls in the category of semi-supervised learning. In this section, we use both synthetic and real data to evaluate the performance of the proposed approach in recovering global and local structures in terms of feature learning and network reconstruction.

5.4.1 Results on synthetic data

In this part, we used two case studies with $N_1 = 5000$ and $N_2 = 5500$ nodes to evaluate the performance of the proposed approach in recovering a global structure.

The preparation of synthetic data

In two case studies, nodes are measured in six different conditions ($M_1, M_2, \dots, M_6$) and distributed across five communities ($G_1, G_2, \dots, G_5$). In the first case study, there are five communities of equal size, with each group consisting of 1,000 nodes. In contrast, in the second case study, the communities have different sizes, with $G_1 = 1000$, $G_2 = 1500$, $G_3 = 500$, $G_4 = 750$, and $G_5 = 1750$ nodes, which were randomly chosen. Note that the sixth condition is a perturbation. In each condition, nodes in the same community are assigned random values from one of the intervals: $A = [1, 100]$, $B = [101, 200]$, $C = [201, 300]$, $D = [301, 400]$, $E = [401, 500]$, and $R = [1, 500]$. We created four datasets for each case study, as shown in Tables 5.1, 5.2, 5.3, and 5.4. In Table. 5.1 (*Data.1*), G_1 and G_2 are adjacent but not overlapped. In Tables 5.2 (*Data.2*), 5.3 (*Data.3*), and 5.4 (*Data.4*), nodes from G_1 and G_2 are assigned values from two, three, and four overlapping intervals, respectively, as indicated in bold font. As the number of overlapping intervals increases, the relative distance between G_1 and G_2 is expected to decrease.

Table 5.1: *The synthetic dataset 1 (Data.1)*

group	Conditions					
	M_1	M_2	M_3	M_4	M_5	M_6
G_1	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>R</i>
G_2	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>A</i>	<i>R</i>
G_3	<i>C</i>	<i>D</i>	<i>E</i>	<i>A</i>	<i>B</i>	<i>R</i>
G_4	<i>D</i>	<i>E</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>R</i>
G_5	<i>E</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>R</i>

Table 5.2: *The synthetic dataset 2 (Data.2)*

group	Conditions					
	M_1	M_2	M_3	M_4	M_5	M_6
G_1	A	B	C	D	E	R
G_2	A	B	D	E	D	R
G_3	B	C	E	A	C	R
G_4	C	D	A	B	B	R
G_5	D	E	B	C	A	R

Table 5.3: *The synthetic dataset 3 (Data.3)*

group	Conditions					
	M_1	M_2	M_3	M_4	M_5	M_6
G_1	A	B	C	D	E	R
G_2	A	B	C	E	D	R
G_3	B	C	D	A	C	R
G_4	C	D	E	B	A	R
G_5	D	E	B	C	B	R

Table 5.4: *The synthetic dataset 4 (Data.4)*

group	Conditions					
	M_1	M_2	M_3	M_4	M_5	M_6
G_1	A	B	C	D	E	R
G_2	A	B	C	D	D	R
G_3	B	C	D	E	C	R
G_4	C	D	E	A	B	R
G_5	D	E	B	C	A	R

Experimental results on synthetic data

Based on the approach introduced in Section 5.2, we generated context sets for our experiments with a tolerance of $\delta_i^\omega = \beta_\omega v_i(\omega)$, where β_ω was set to 0.1. In the experiments, we generated $K = 10$ random node sequences of length $l = 80$ starting from each node in each of the six conditions. Consequently, the corpus T_1 and T_2 consist of 300000 and 330000 node sequences, respectively. In the neural network, we set the node vector dimension to $d = 128$.

To evaluate the training results qualitatively, we mapped the trained node vectors to a 2D plane using the Principle component analysis (PCA)^{188;189}. In Fig. 5.2(a) and 5.3(a),

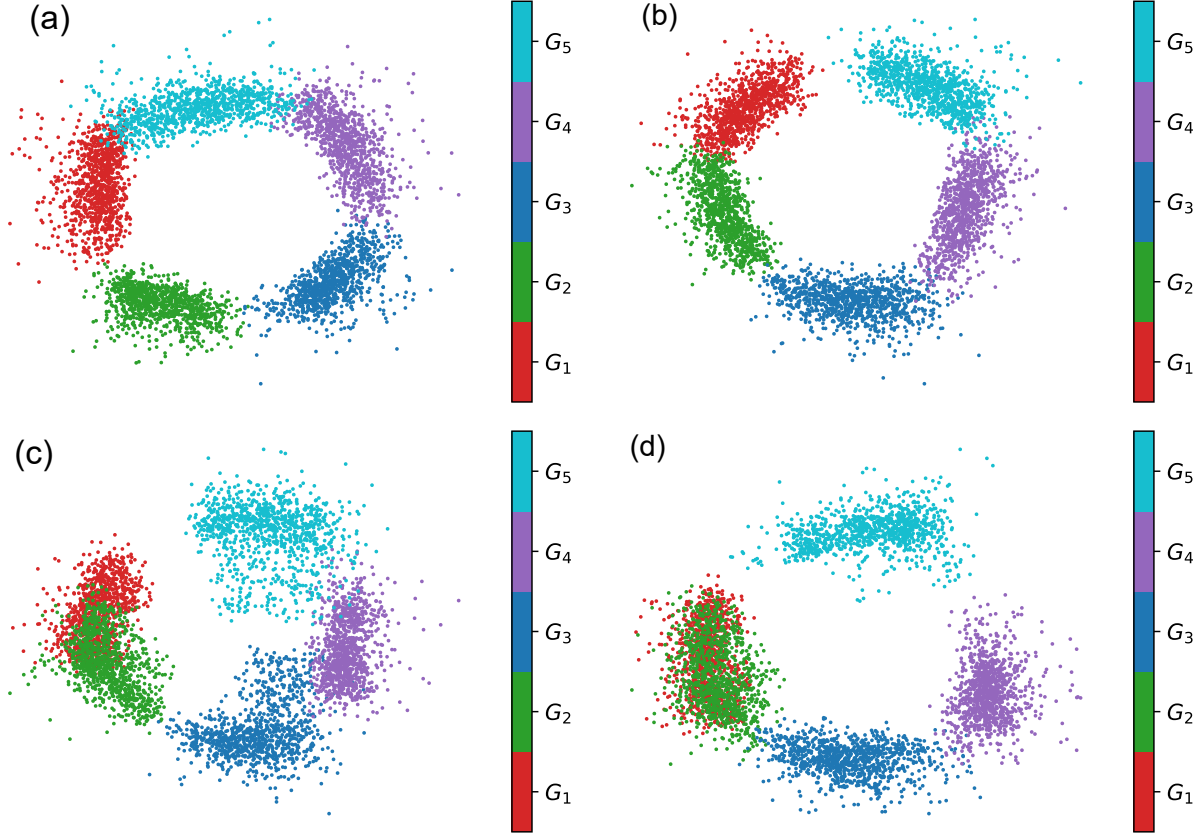


Figure 5.2: The node vectors trained from the first case study are visualized via PCA. The five communities have an equal number of nodes. Panels (a) to (d) are respectively the training results of *Data.1* to *Data.4*.

the nodes from the same communities are mapped to the same areas, meaning that the proposed method can recover the global structure of the dataset. As previously mentioned, nodes in G_1 and G_2 are assigned to values from two, three, and four overlapped sub-intervals from *Data.2* to *Data.4*. Therefore, the distance between G_1 and G_2 is assumed to be decreasing for *Data.2*, *Data.3*, and *Data.4*. In Fig. 5.2(b) and 5.3(b), we observed the relative distance between G_1 and G_2 was closer than that in Fig. 5.2(a) and 5.3(a). Similarly, the relative distance between G_1 and G_2 was even closer in Fig. 5.2(c) and 5.3(c), and the two communities were almost merged in Fig. 5.2(d) and 5.3(d). The results for the two case studies (eight datasets in total) demonstrated that the node vectors could reflect the relative distances of the node communities, which are affected by the number of overlapped sub-intervals.

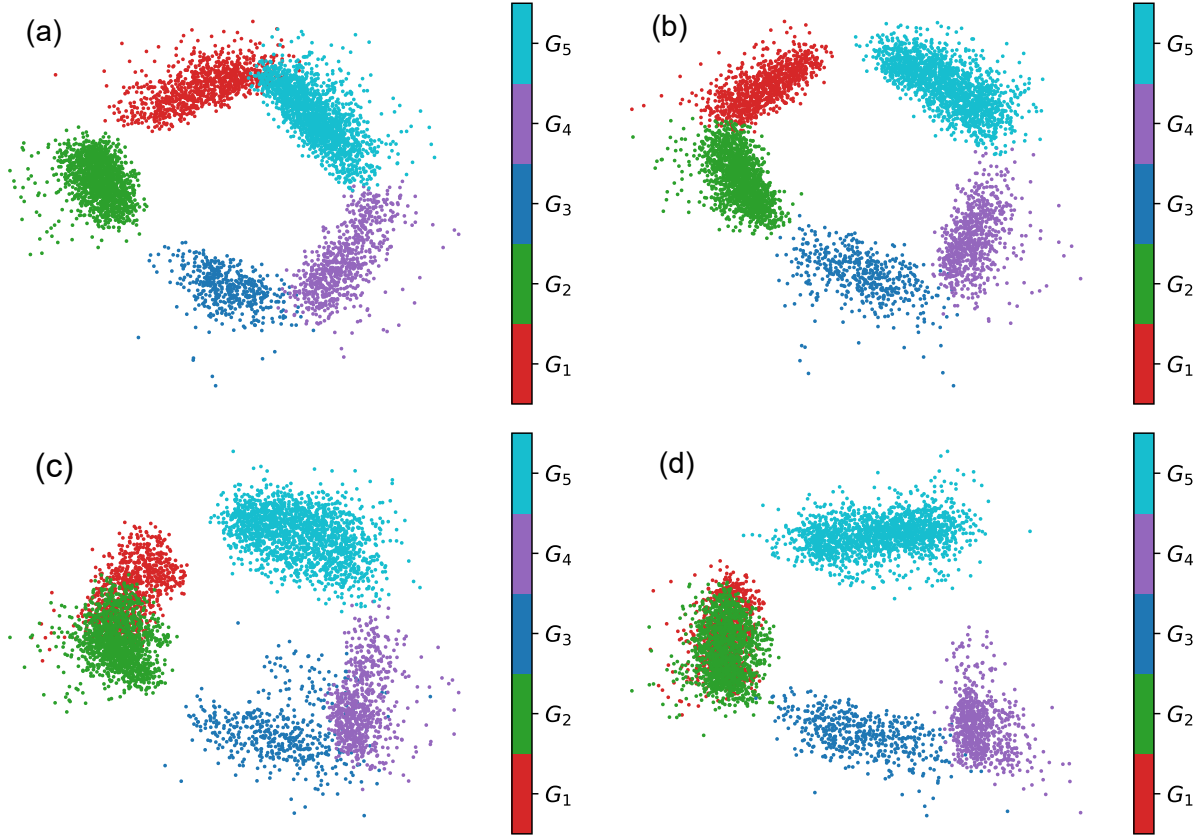


Figure 5.3: The node vectors trained from the second case study are visualized via the PCA. The five communities have 1000, 1500, 500, 750, and 1250 nodes. Panels (a) to (d) are, respectively, the training results of Data.1 to Data.4.

Table 5.5: The relative distance between G_1 and G_2 and classification accuracy

Data.	First case study		Second case study	
	Distance	Accuracy(%)	Distance	Accuracy(%)
1	2.18	99.98	2.14	100
2	1.82	99.96	1.86	99.98
3	1.52	99.96	1.49	99.90
4	1.18	79.84	1.16	77.92

To quantify our results, we computed the distance between G_1 and G_2 . We achieved this by calculating the cosine distance (1-cosine similarity) between every pair of nodes. The distance between G_1 and G_2 was computed as the sum of all possible node pairs between the two communities. For instance, the cosine distance between G_1 and G_2 can be expressed as

follows:

$$Dis(G_1, G_2) = \sum_{v_i \in G_1, v_j \in G_2} 1 - s(v_i, v_j). \quad (5.21)$$

Then, we calculate the relative distance between G_1 and G_2 as:

$$RelaDis(G_1, G_2) = \frac{Dis(G_1, G_2) * Dis(G_1, G_2)}{Dis(G_1, G_1) * Dis(G_2, G_2)}. \quad (5.22)$$

In addition, we applied the simple K -means clustering method¹⁹⁰ to classify the trained node vectors into five distinct communities. We then compared the resulting classification with the ground-truth communities.

Table 5.5 shows the relative distances between G_1 and G_2 , as well as the classification results. As expected, the cosine distance between G_1 and G_2 decreases from *Data.1* to *Data.4*, which is consistent with the visualizations presented in Fig. 5.2 and Fig. 5.3. Notably, the distance is close to one for *Data.4*, indicating that the two communities have almost merged. The classification results also agree with the visualization. The classification results also align with the visualizations, with a classification accuracy of over 99% for *Data.1*, *Data.2*, and *Data.3*. However, the accuracy drops significantly for *Data.4* due to the high degree of overlap between the two communities, resulting in nodes being incorrectly classified. Overall, the node vectors successfully recover the mesoscopic structure of the nodes. For the following experiments, we will focus only on the first case study for further analysis.

Table 5.6: *The influence of missing values on the distance between G_1 and G_2 and classification accuracy*

Data.	10% missing values		20% missing values	
	Distance	Accuracy(%)	Distance	Accuracy(%)
1	2.19	100	2.16	99.92
2	1.77	99.92	1.76	99.72
3	1.49	99.64	1.47	98.76
4	1.15	79.68	1.14	79.96

To investigate the impact of missing values, we generated incomplete datasets by ran-

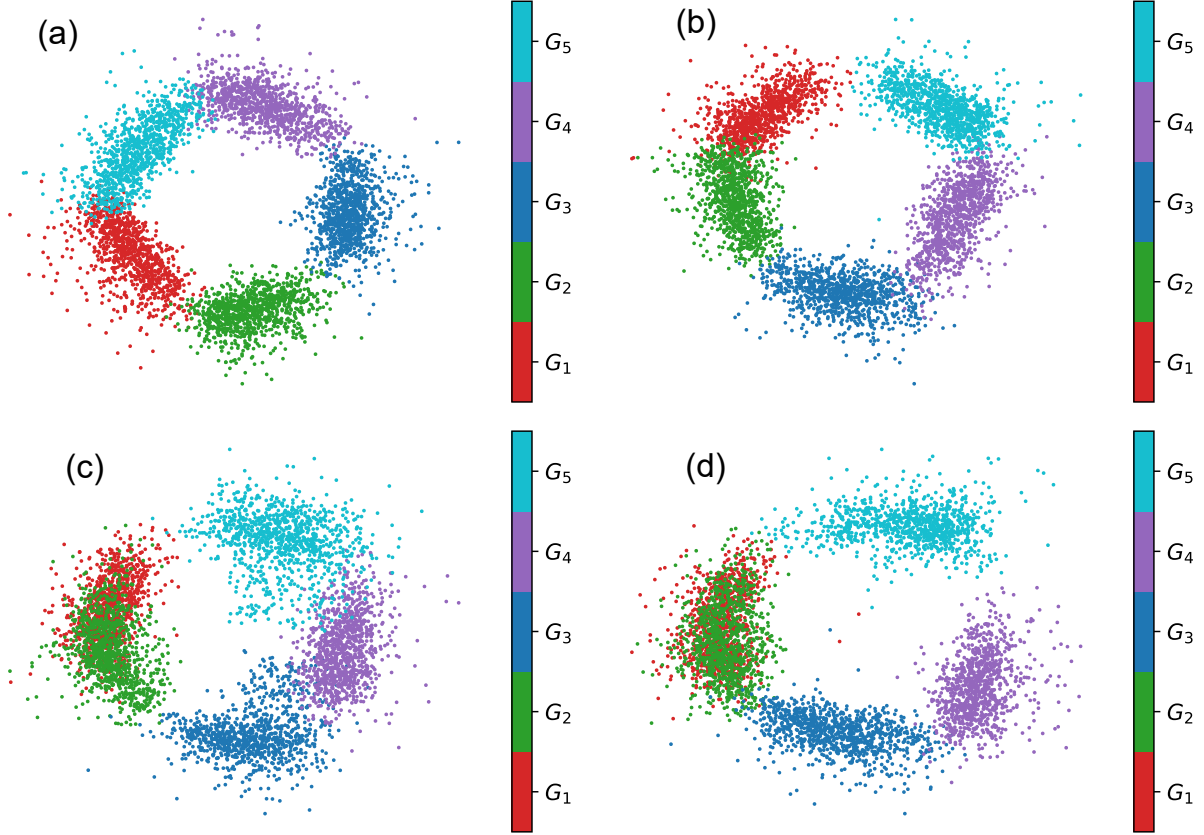


Figure 5.4: Visualization of node vectors trained from the first example with 20% missing values. Panels (a) to (d) are, respectively, the training results of Data.1 to Data.4.

domly removing 10% and 20% of values from each condition. Using the same parameters as before, we trained the neural network and present the results in Table 5.6. Interestingly, we observed that the relative distances between G_1 and G_2 , as well as the classification accuracies, were not significantly affected by the missing values. In addition, Fig. 5.4 illustrates that the global structure of the nodes can still be recovered even when 20% of the data is randomly removed. These findings suggest that the proposed method is robust to missing values.

The training results exhibit robustness to the selection of training parameters. To regulate the size of the context set δ_ω^i in the generation of node sequences, we assigned distinct values to β_ω . Subsequently, we trained the node sequences utilizing a neural network model with $\rho = 1$. Moreover, we calculated the relative distances between G_1 and G_2 , along with the classification accuracy. Based on the data in Table 5.7, it can be inferred that the relative

Table 5.7: The relative distance between G_1 and G_2 and classification accuracy w.r.t. the variation of $\beta_\omega \in \{0.05, 0.1, 0.15, 0.2\}$

Data.	Distance				Accuracy(%)			
	0.05	0.1	0.15	0.2	0.05	0.1	0.15	0.2
1	2.08	2.18	2.22	2.16	100	99.98	99.86	99.86
2	1.78	1.82	1.82	1.76	99.98	99.96	99.86	99.88
3	1.49	1.52	1.48	1.47	99.92	99.96	99.06	99.84
4	1.15	1.18	1.16	1.14	80.72	79.84	80.86	81.34

distances between the same datasets are consistent, and the classification accuracies are not significantly influenced by β_ω . This implies that the global structure remains intact, despite varying β_ω . Hence, the selection of β_ω has a limited impact on the embedded node vectors.

To compare the proposed approach with the commonly used correlation approach^{117;164}, we generated four networks from synthetic data and trained them with semi-supervised learning algorithms to obtain node vectors. Prior to training, we normalized the values of each condition using the z-score method:

$$\bar{v}_i(\omega) = \frac{v_i(\omega) - \mu_\omega}{\sigma_\omega} \quad (5.23)$$

where μ_ω is the mean of all the values in the ω th condition, σ_ω is the standard deviation, and $\bar{v}_i(\omega)$ is the normalized expression value.

The Pearson correlation coefficient (PCC) of any two nodes is:

$$r_{(v_x, v_y)} = \frac{\sum_{i=1}^M (\bar{v}_x(\omega) - \bar{v}_x)(\bar{v}_y(\omega) - \bar{v}_y)}{\sqrt{\sum_{i=1}^M (\bar{v}_x(\omega) - \bar{v}_x)^2} \sqrt{\sum_{i=1}^M (\bar{v}_y(\omega) - \bar{v}_y)^2}} \quad (5.24)$$

where $r_{(v_x, v_y)}$ is the PCC between node v_x and v_y , and $\bar{v}_x$ is the mean of node v_x across the M conditions. The PCC measures how much the two genes are related^{164;166}. In this experiment, we did not consider the missing value problem, which could substantially influence the correlation coefficients according to the results in¹¹⁷. The edges are selected by thresholding correlation coefficients, such that $\text{PCC} \geq 0.95$ ^{127;166}. All four networks have edge densities of over 5%, as shown in Table 5.8.

Table 5.8: *The relative distance between G_1 and G_2 and the classification accuracy of the PCC networks*

Data.	Edge density	Distance	Accuracy(%)
1	6.30%	6.13	100
2	5.81%	3.64	81.30
3	5.21%	2.66	80.82
4	5.73%	1.17	64.26

Various methodologies are employed to derive node vectors from network structures for studying node properties^{88;169–171}. In this study, we utilized the node2vec approach introduced in¹⁶⁹ to obtain node vectors from the constructed networks. This approach has demonstrated remarkable performance in network reconstruction. Furthermore, we computed the relative distances between G_1 and G_2 from the trained node vectors, and the results are presented in Table 5.8. It was observed that the relative distances between the two communities for the first three networks were considerably higher than in our proposed approach (Table 5.5). Although G_1 and G_2 were assumed to overlap in the synthetic datasets partially, this characteristic was not captured in the trained node vectors, as demonstrated in the visualization of Fig. 5.5. The PCC method is susceptible to errors introduced in data normalization, network construction, and feature learning, which may consequently affect the accuracy of trained node vectors. In contrast, our proposed approach trains the node vectors directly from the raw data, allowing for a more accurate representation of the underlying network structure.

We investigate the impact of ρ on the relative distance between G_1 and G_2 , and the results are presented in Tables 5.9 and 5.10. It was observed that the distance between the two communities slightly increased when a low value of ρ was employed, as a small ρ promotes the addition of nodes that also exist in the context set of the previous node. Consequently, far-away nodes tend to become closer, resulting in a decreased distance between the two communities. Nonetheless, the results were not significantly influenced by ρ , since the relative distances between the two communities remained at the same level for the same dataset. Therefore, we suggest using $\rho = 1$ in most scenarios.

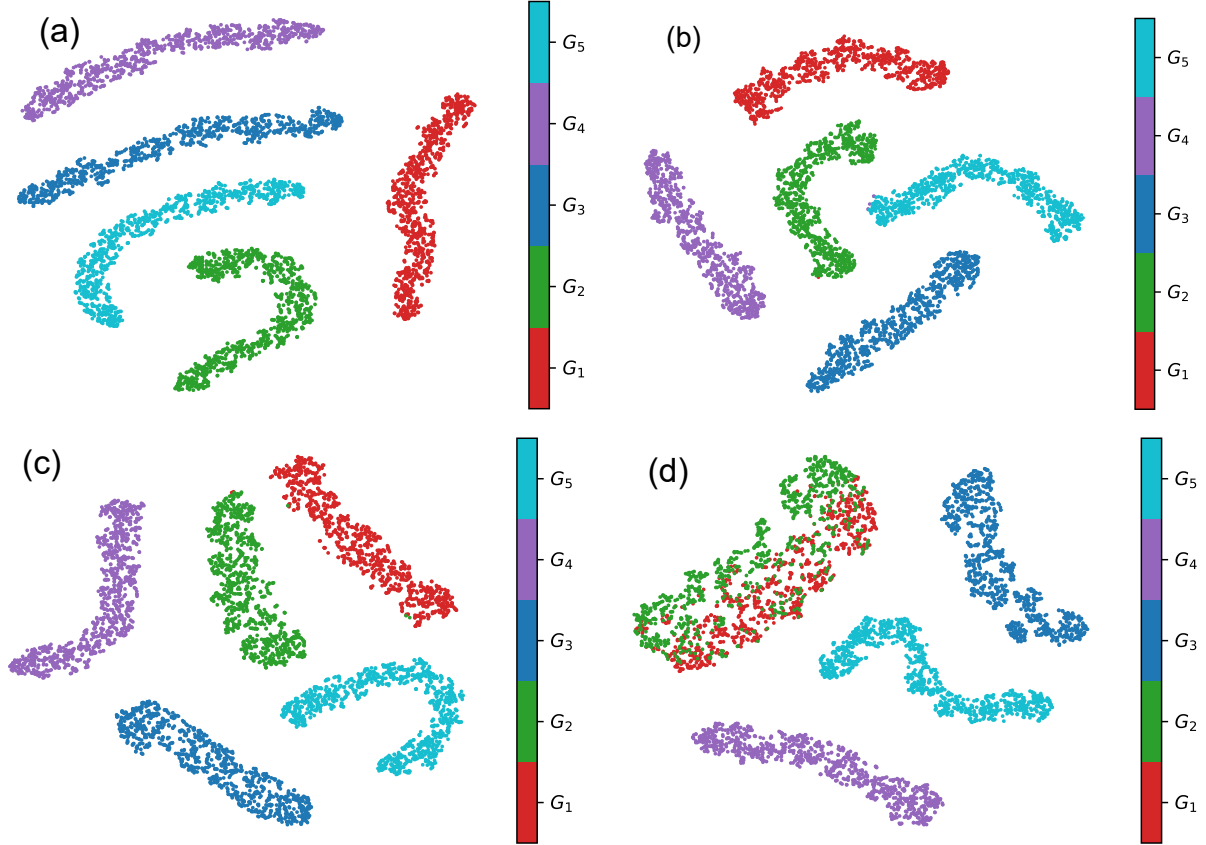


Figure 5.5: The visualization of node vectors trained from the Pearson correlation network. Panels (a) to (d) are, respectively, the visualizations of Data.1 to Data.4.

Table 5.9: The relative distance between G_1 and G_2 w.r.t. ρ

Data.	1/10	1/5	1/3	3	5	10
1	2.23	2.22	2.19	2.18	2.17	2.07
2	1.84	1.83	1.82	1.80	1.78	1.67
3	1.53	1.52	1.52	1.48	1.42	1.32
4	1.22	1.21	1.20	1.17	1.16	1.13

5.4.2 Results on real data

Data description

We utilized two real *An. gambiae* gene expression datasets^{2,74} to demonstrate that the learned node vectors could capture the local structure of the nodes. The first dataset comprises 10,433 *An. gambiae* genes that were measured in time series after desiccation stress,

Table 5.10: *The prediction accuracy (%) w.r.t. ρ*

Data.	1/10	1/5	1/3	3	5	10
1	98.24	98.24	98.72	98.80	98.90	98.82
2	98.26	97.22	98.68	98.00	97.30	98.12
3	97.52	97.48	97.72	98.22	97.58	97.32
4	78.14	78.58	78.62	79.72	79.32	79.78

with five conditions⁴⁵. The five measurements (conditions) of each gene were almost at the same level, and the distributions of the *coefficient of variation* (*CV*) and means of the 10,433 genes are displayed in Fig. 5.6(a) and (c). The second dataset measured gene expression values after mating, with four measurements (also in time series)⁵⁰. The distributions of the *CV* and means are presented in Fig. 5.6(b) and (d).

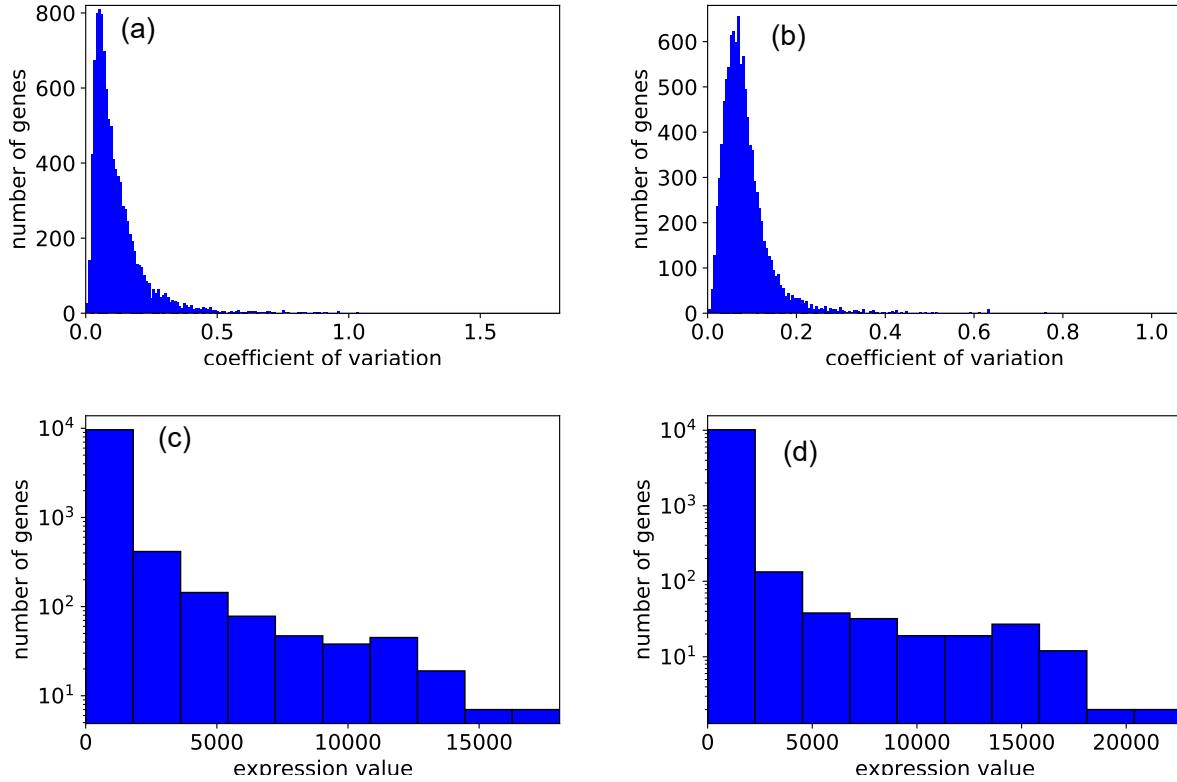


Figure 5.6: *The properties of the two real datasets. Panels (a) (first dataset) and (b) (second dataset) are the distributions of the CV of the expression values. Panels (c) (first dataset) and (d) (second dataset) are the distributions of means of the expression values.*

In Fig. 5.6(a) and (b), we observe that the CVs of most genes are at a low level.

Therefore, we can set the tolerance δ_i^ω as the average CV of all the nodes, such that

$$\overline{CV} = \frac{1}{N} \sum_i \frac{\sigma_i}{m_i}, \quad (5.25)$$

where m_i is the mean value of gene v_i , σ_i is the standard deviation, and $\frac{\sigma_i}{m_i}$ is the CV of gene v_i . The $\overline{CV}$ s of the two data sets are respectively 0.086 and 0.12. Therefore, we set $\beta_\omega = \overline{CV}$.

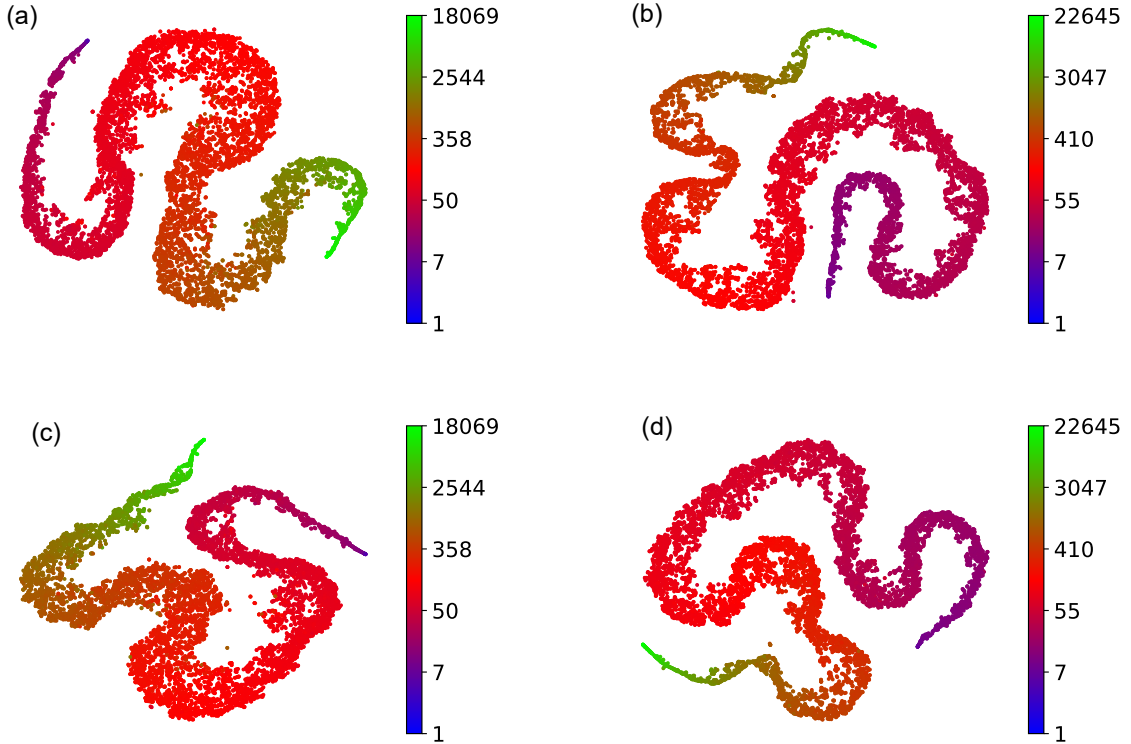


Figure 5.7: Visualization of the gene vectors via t -SNE, the genes are colored by the expression values. Panels (a) (first dataset) and (b) are the visualizations of node vectors trained from raw data. Panels (c) (first dataset) and (d) (second dataset) are the visualizations of node vectors trained from incomplete datasets with 20% values randomly removed from each condition.

Experimental results

The node vectors that have been trained are displayed in Fig. 5.7 using the t-distributed stochastic neighbor embedding (t-SNE) method¹⁹¹. Unlike other methods, t-SNE constructs

a probability distribution over pairs of vectors and does not retain the distances between node pairs, but their probabilities. Therefore, t-SNE has a better performance in preserving local structure. As shown in Fig. 5.7, genes with similar expression values are mapped closer together, even when 20% of values have been removed from each condition.

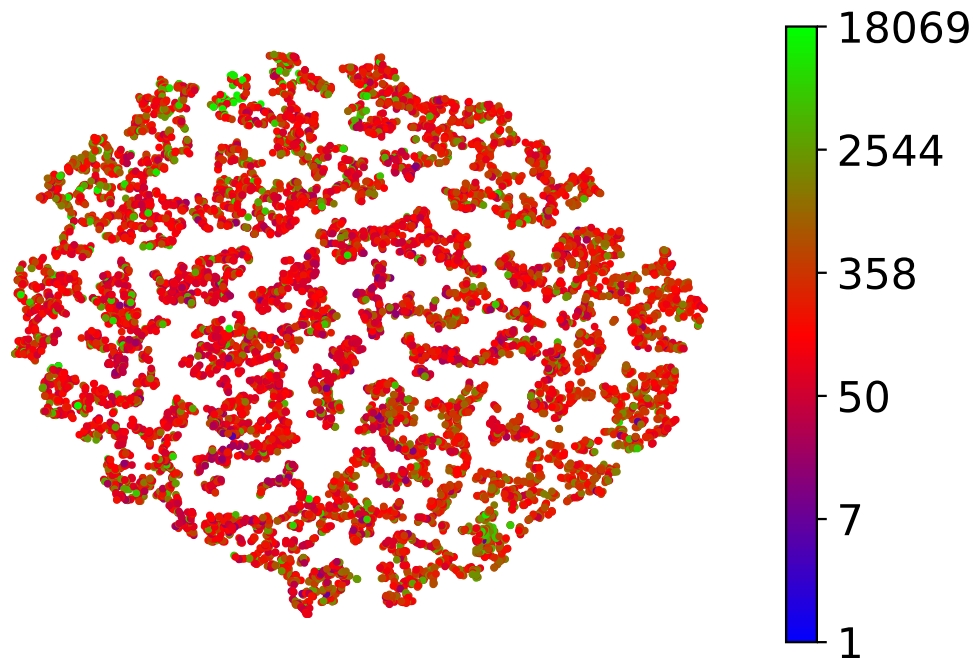


Figure 5.8: *The visualization of node vectors trained from the PCC network.*

We created a PCC network for the first real dataset to provide a comparison. The raw expression values were rescaled using $\log_2^{2;166}$ and normalized using Eq. 5.23 because the distribution of the raw data was heterogeneous, as shown in Fig. 5.6(c). We constructed the PCC network by setting the edge threshold to $\text{PCC} \geq 0.95$ (the resulting network was not sparse) and contained 756,330 edges. We also trained the node2vec model to obtain node vectors in the same manner as before. However, as shown in Fig. 5.8, we observed that nodes were distributed randomly in the 2D plane, indicating that nodes with similar values were not mapped to the same area. For example, the expression values of two genes, AGAP004677 and AGAP012093, were [2764, 2869, 3276, 3690, 3671] and [129, 149, 184,

221, 265], respectively, and it was clear that their expression values were at different levels. However, the PCC between the two genes was 0.983, indicating that the two nodes were highly related. This is because the PCC method does not consider the scale of expression values but only detects the linear dependence of two genes. In contrast, our approach assumes that similar nodes have more shared elements in their context sets.

5.5 Results of network structure extraction

5.5.1 Compare the GTE and REM

Thresholding similarity values is the most commonly used approach in network construction. However, this approach can lead to some nodes being isolated from the network because they may have relatively low similarities to all other nodes. In Fig. 5.9, we applied different thresholds to the cosine similarities computed from the node vectors trained with the synthetic data (Fig. 5.2(a)) and the real data (Fig. 5.6(a)). We observed that the percentage of isolated nodes increased rapidly when the thresholds exceeded 0.8 (for synthetic data) and 0.95 (for real data), respectively. In this chapter, we refer to such a threshold as the critical value. If we use a threshold smaller than the critical value, most nodes will be connected to at least one other node. However, if the threshold is larger than the critical value, we may obtain a network with a large percentage of singleton nodes.

In real applications, we can force isolated nodes to connect with highly similar nodes. However, the neighbors selected through a single threshold are not influenced by the distribution of similarity values. It is not uncommon for hubs to be connected to many other nodes with relatively low similarity values, while leaf nodes may only connect to a small number of nodes with high similarities. In other words, the distribution of similarity values is not taken into account when selecting edges.

The proposed REM method ensures that every node is connected to at least one other node by determining the "threshold" of each node based on the distribution of similarity values per Eq. 5.19. In the experiments, we applied the REM method to the two datasets

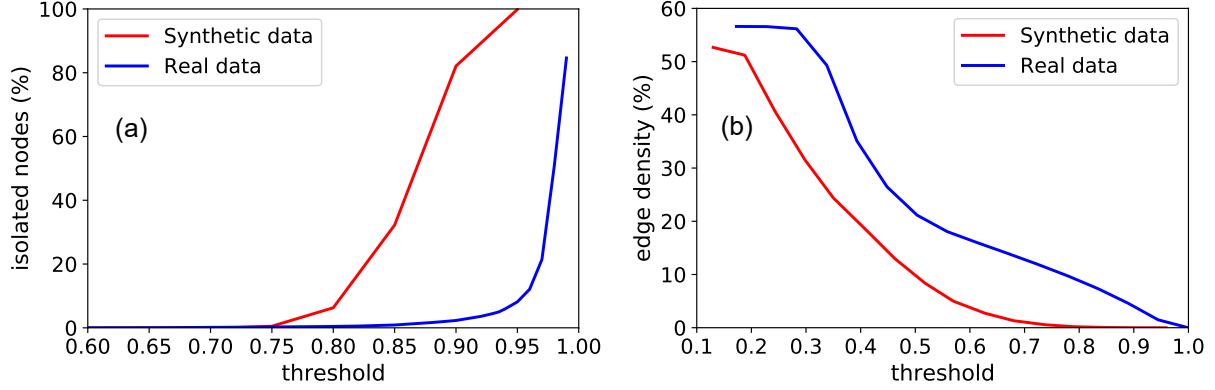


Figure 5.9: *Experimental results when different thresholds are applied. (a) shows the percentages of isolated nodes w.r.t. the thresholds. (b) shows the edge densities of networks when different thresholds are used.*

used in Fig. 5.9, and the results are shown in Fig. 5.10. We observed that the edge density decreases rapidly in the first few iterations, and then decreases gradually. This is because weakly connected edges are removed immediately from the network in the first few cycles, while the remaining edges have relatively high similarities, which are removed at a slower rate. Additionally, the parameter α can be used to control the removal speed and edge density. Higher values of α remove weak links more efficiently, as discussed in Sec. 5.3. In real applications, we can fix α and iteratively update Eq. 5.17 until we obtain a network with the desired edge density.

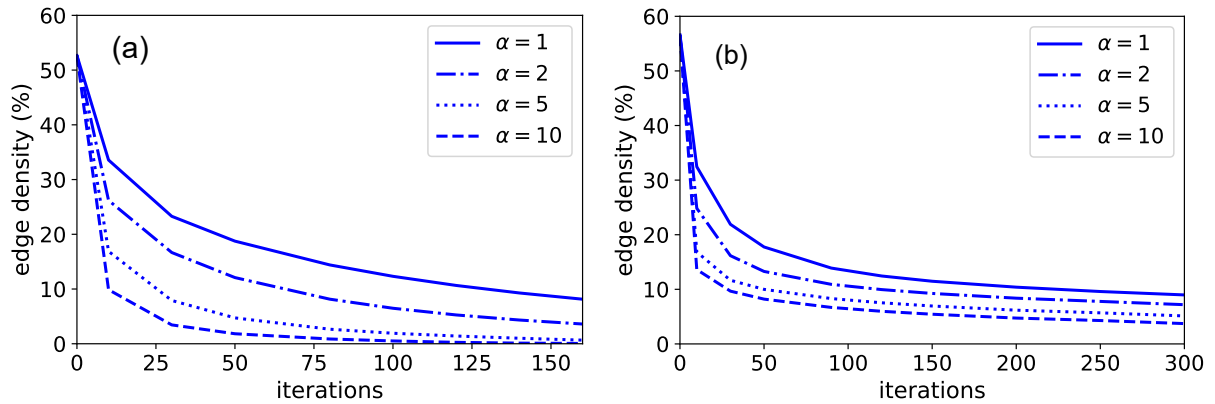


Figure 5.10: *Edge density analysis of the REM. (a) Synthetic data. (b) Real data.*

To further evaluate the performance of the proposed REM, we compared it with the widely used GTE method in network construction. Specifically, we generated four GTE-

Table 5.11: *The properties of networks constructed with the GTE and REM*

Property	<CT ¹ (Syn. ²)		=CT (Syn.)		<CT (Real. ³)		=CT (Real)	
	GTE	REM	GTE	REM	GTE	REM	GTE	REM
Threshold	0.7	-	0.8	-	0.92	-	0.95	-
Isolated nodes	0	0	287	0	372	0	849	0
Edge density	1.03%	0.972%	0.164%	0.160%	2.88%	2.81%	1.18%	1.18%
Ave. degree	51.5	48.6	8.2	8	300.8	293.2	123.2	123.4
Ave. clustering	0.46	0.41	0.38	0.32	0.66	0.55	0.59	0.37

¹ CT denotes Critical threshold

² Syn. denotes the synthetic data used in Fig. 5.9 and 5.10

³ Real. denotes the real data used in Fig. 5.9 and 5.10

based networks with different thresholds for the datasets used in Fig. 5.9 and compared their properties with those of the four REM networks. The results are summarized in Table 5.11. The GTE networks were generated using thresholds that were less than or equal to the critical thresholds. In addition, we ensured that the edge densities of the GTE and REM networks were similar. In Table 5.11, we found that both GTE and REM return networks with similar average degrees when the edge densities are the same. However, the GTE networks always have a higher average clustering coefficient, indicating that nodes in the GTE networks are more likely to cluster together. In Fig. 5.11, we compared the degree distributions of the eight networks. We observed that the degree distributions of the GTE and REM networks almost overlap when the thresholds are less than the critical values (panels (a) and (c)). When the thresholds are at the critical values (panels (b) and (d)), some nodes in the REM networks still have high degrees, which are similar to the *hubs* in many real networks. Furthermore, we observed that all four REM networks had many low-degree nodes, which account for the lower average clustering coefficients in Table 5.11.

In the analysis of the effect of edge density on node roles, we plotted a scatter diagram in Fig. 5.12, where each point represents a node. The horizontal axis shows the node's degree in the densely connected network, while the vertical axis represents the degree in the sparse network. Our observations revealed that the node degree of the GTE networks is significantly impacted by the threshold selection. The highest node degrees decreased from 345 to 73 for synthetic data and from 704 to 370 for real data. In the REM network, the highest node degrees have dropped from 321 to 210 for the synthetic data and 682 to 493 for

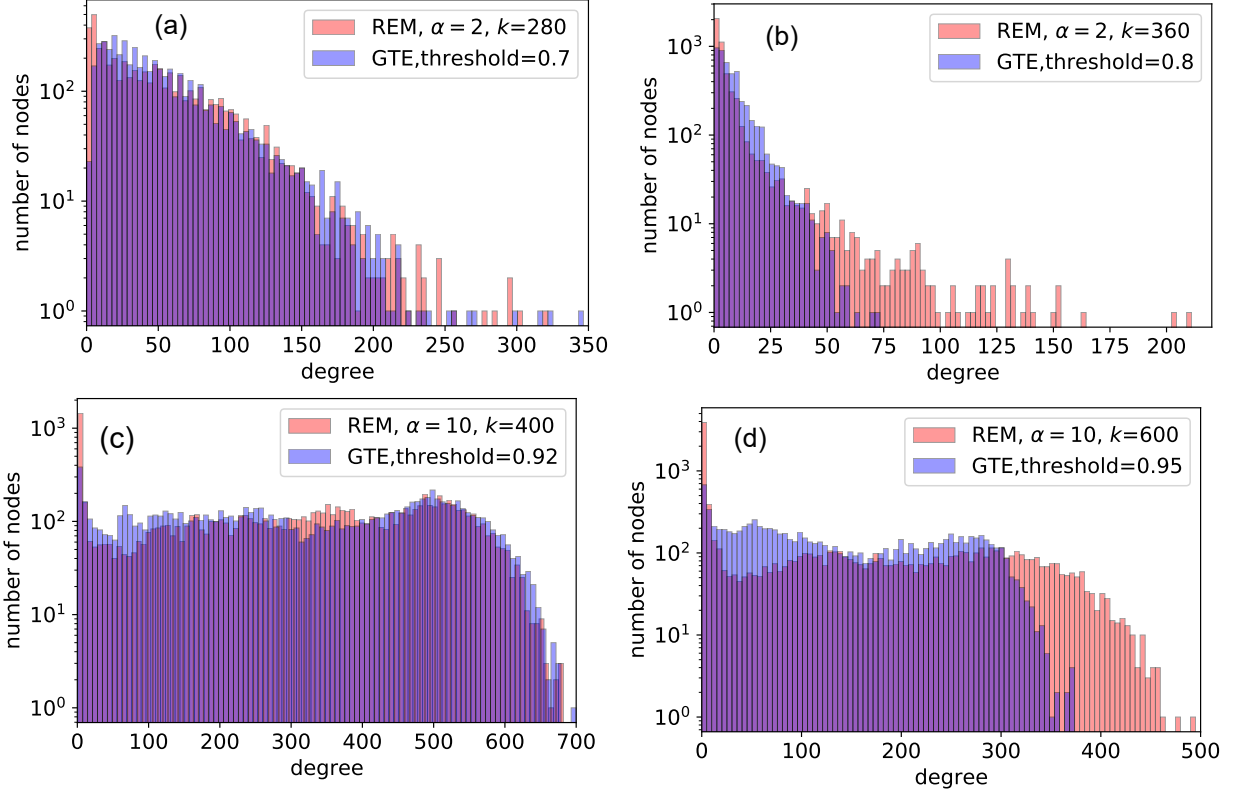


Figure 5.11: *The degree distributions of networks generated from the GTE and REM. (a) and (b) are the degree distributions of the networks constructed from synthetic data. (c) and (d) are the degree distributions of the networks constructed from real data. Panels (a) and (c) show the results of densely connected networks, while (b) and (d) are the results of sparsely connected networks.*

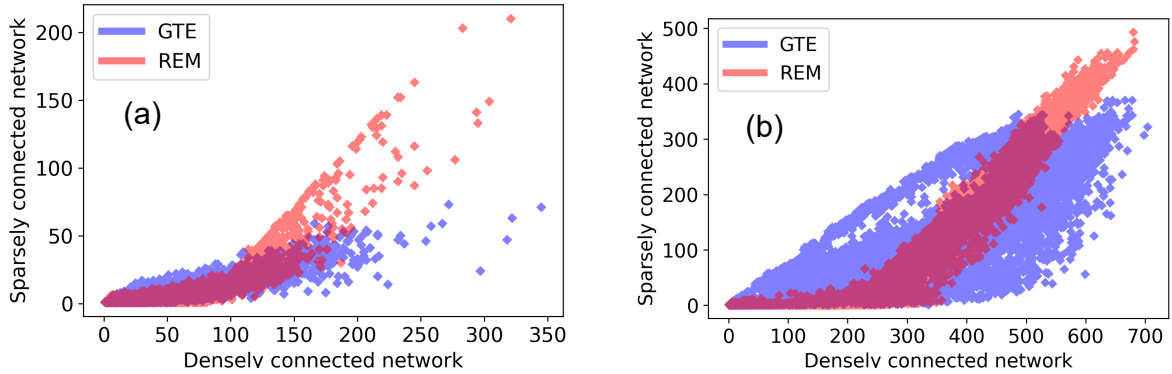


Figure 5.12: *The comparison of node degrees between densely and sparsely connected networks.*

the real data. Notably, the REM approach tends to eliminate edges from low-degree nodes while proportionally removing edges from high-degree nodes, thereby maintaining their roles

without significant influence from edge densities (as shown in Fig. 5.12). In contrast, the resulting GTE networks are heavily influenced by the chosen edge density. As shown in Fig. 5.12(b), the points of the GTE networks are over-dispersed in the diagram.

5.5.2 Study the REM approach with AUC metrics on real data

In this section, we evaluate the proposed approach using two real datasets that contain both network structure and node activity data. It is challenging to establish a quantitative relationship between network structure and node activity data as they describe the nodes' properties from different perspectives. To address this, we adopt a methodology where we first learn node vectors from the node activity data and then calculate their similarity to construct the networks. We compare the constructed networks with the actual network structures, and we use the Area Under the Curve (AUC) metric to evaluate the effectiveness of our REM approach.

The cora dataset

The Cora dataset¹⁹² consists of a sparse citation network containing 2708 nodes and 5278 edges, resulting in an edge density of 0.144%. The nodes in the network represent publications, and the edges represent the citation relationships between the papers. Each node is represented by a binary word vector with 1433 unique words, which indicates the presence or absence of the corresponding word in the publication. The dictionary includes only those words that appear at least ten times in one of the 2708 publications.

The pubmed dataset

The PubMed dataset¹⁹³ contains a sparse citation network comprising 19717 nodes and 44324 edges, resulting in an edge density of 0.0228%. The nodes in the network represent publications, and the edges represent the citation relationships between the papers. Each node is described by a TF/IDF weighted word vector from a dictionary that consists of 500 words.

We applied the proposed approach to the Cora and PubMed datasets to generate 128-dimensional node vectors from the feature datasets. We then computed the similarity between each pair of nodes, and the performance of the approach was evaluated by comparing the constructed networks to the true citation networks. The AUC values for the Cora and PubMed datasets were 0.81 and 0.73, respectively. Although the true relationship between network structure and node activity data is unknown, the results suggest that there is a relationship between the node activity data and the network structure. One of the advantages of our approach is that it enables us to compare and integrate two different types of data, which is valuable for gaining insights into complex systems.

5.6 Summary

This chapter presents a neural network-based approach for learning node vectors from noisy node activity data. The primary advantage of the proposed method is that data are not required to follow any specific distribution. The approach generates *context sets* from raw data for each condition, which allows it to be more flexible and adaptable to different data types. Moreover, this approach can handle missing values that are commonly found in experimental results. Inspired by the application of neural networks in natural language processing, we generate a corpus of node sequences to simulate sentences in documents. The corpus is fed into a three-layer neural network model, which produces node vectors that enable the comparison and identification of nodes with synergistic roles.

The experimental results show that the proposed approach is robust to the choice of parameters and missing values. In addition, we offer an alternative method to select edges for the underlying network. This is accomplished through the REM method, which is based on the Rényi entropy and selects edges based on the distribution of similarity values. By doing so, the proposed approach can construct networks without isolating nodes and can recover the roles of nodes.

In this study, we conducted two experiments to evaluate the effectiveness of the proposed method. Using both synthetic and real data, we demonstrated that the method is capable

of uncovering the global and local structure of nodal data, even when up to 20% of values are randomly removed from the datasets. These results indicate that the proposed method is robust to missing data and can effectively learn nodal representations from noisy input. Moreover, we evaluated the performance of the proposed entropy-based network extraction method. By adjusting the parameter α and the number of iterations, we were able to obtain a network with the desired edge density without isolating any nodes. This suggests that the proposed method can construct networks that preserve the connectivity and roles of nodes, which is essential for accurate analysis and interpretation of the data.

Overall, the proposed approach offers several advantages over existing methods for learning node vectors from noisy data. Its flexibility, robustness, and ability to handle missing values make it a promising technique for analyzing complex datasets with noisy node activity data.

Chapter 6

Conclusion and Future work

6.1 Conclusion

In this dissertation, we have investigated the problems of network construction and feature learning from node interaction and activity data. Specifically, this dissertation conducted the research and illustrated models by studying biological data. Gene co-expression networks are well-known examples of how networks are used to solve practical problems^{31;164}. Data produced by different biological experiments are often rich and multimodal³. Network science provides a variety of tools for biologists to identify how genes interact with each other and co-regulate in biological activities. With this background, this dissertation aims at solving the problems of layer reconstruction and missing link prediction in multilayer networks and network structure extraction from gene expression data. The models designed in chapters 2, 3, 4, and 5 are also applicable to study other data types.

In Chapter. 2, we introduced a MAP model to reconstruct a target layer of a multilayer network. We employed the SimHash algorithm to detect layers that are structurally similar to the (incomplete) target layer. Those structurally similar layers are used to compute the parameters of the conjugate prior, which enables us to make accurate predictions for the target layer. In our experiments, we compared the proposed approach to the widely used MLE method. We found that the proposed approach has better performance when there are

40% or more missing links in the target layer. While MLE is a good choice when there are no structurally similar layers.

In Chapter. 3, we analyzed gene expression data from the *An. gambiae* species, collected from 30 publications. The expression values are normalized using the z-score normalization method, which results in zero mean and unit variance. The Pearson correlation coefficient is calculated for each gene pair based on the normalized expression data. To account for the ubiquitously encountered missing values, we use a fitted sliding threshold to select edges for the correlation network, which is determined by the number of paired elements of gene pairs. Our method demonstrates robustness against random removal of up to 15% of the experimental conditions. The Gene Ontology (GO) term and KEGG pathway enrichment analysis of the constructed network suggest that the *An. gambiae* transcriptome architecture optimizes the integration of essential cellular processes and enables evolutionary flexibility to integrate the expression of novel biological functions.

In Chapter. 4, we presented the *GeCoNet-tool*, an integrated tool for gene co-expression network construction and analysis. The first part of the tool offers users various options for processing gene co-expression data obtained from different technologies. The tool’s output is an edge list, which may optionally include weights associated with each edge. In the second part, users can generate a table containing several network properties, including communities, cores, and some centrality measures.

In Chapter. 5, we proposed a neural network-based method for learning node representations from noisy node activity data. Instead of detecting node correlations, the proposed approach discovers the distance relationships between nodes from heterogeneously distributed node activity data. We created a corpus of node sequences from context sets extracted from raw node activity data. The node sequences are then trained using a neural network model, producing node representations that enable comparison and identification of nodes with synergistic roles. Our experiments showed that the proposed method is resilient to changes in neural network parameters and missing values in the node activity data. To construct the underlying network, we introduced a Rényi entropy-based approach to select edges from the trained node representations for the underlying network. The resulting network construction

method does not isolate nodes and accurately recovers their roles.

6.2 Future Work

In this section, we provide some recommendations for future work based on the limitations of this dissertation.

In the model described in Chapter. 2, it is necessary for the layers used to calculate the conjugate prior to being similar to the target layer. This similarity is either predetermined or detected through feature comparisons. Future work could explore using different properties within the SimHash algorithm to find the most similar layers. Additionally, we evaluated different values for the number of communities (K) and determined the optimal value by balancing convergence time and maximum posterior. Further research could potentially quantify the relationship between the number of communities and estimation accuracy.

The study in Chapter. 3 was limited to the analysis of *An. gambiae* genes. Future research should consider analyzing genes from other species to strengthen the conclusions drawn from this study. Additionally, the sliding threshold used in this study was selected by the rule of thumb. It would be valuable to investigate whether other curves could be used as sliding thresholds in future studies. Moreover, in this study, the Louvain algorithm was used to detect communities in the constructed network. However, the choice of the network community detection algorithm should be based on GO term and KEGG pathway enrichment. Therefore, other community detection algorithms, such as the Girvan-Newman algorithm¹¹¹, should be evaluated. These algorithms perhaps provide further insights into the network structures of genes and their functional relationships.

Chapter. 4 presented an integrated software package for gene co-expression network construction and analysis. Although the package allows users to analyze network structures from various perspectives, we only provide the Fruchterman-Reingold layout¹⁵⁷ for network visualization. In the future, it would be beneficial to incorporate additional network visualization layouts into the package, such as the force atlas algorithm¹³⁷. This would provide users with more options to visualize and interpret the network structures in a way that

best suits their research needs. By incorporating different visualization methods, we can enhance the usability of the software package and make it more accessible to a wider range of researchers.

In Chapter. 5, the experiments demonstrated the potential of the proposed approach in detecting global and local structures from noisy node activity data. One of the primary advantages of the proposed approach is that it eliminates the need for missing value interpolation or prediction. However, in the generation of context sets, elements in the set are determined based on value differences, meaning that we can only detect nodes with close values. In future work, data normalization algorithms (e.g., row normalize) can be used to extend the application of the model in solving real-world problems. By applying normalization techniques, we can ensure that the context sets are generated based on relative rather than absolute values. This would allow the model to handle data with different scales and ranges better, leading to more accurate and reliable results.

Bibliography

- [1] Mark Newman. *Networks*. Oxford university press, 2018.
- [2] Robert M MacCallum, Seth N Redmond, and George K Christophides. An expression map for anopheles gambiae. *BMC genomics*, 12(1):1–16, 2011.
- [3] Mark EJ Newman. Network structure from rich but noisy data. *Nature Physics*, 14(6):542–545, 2018.
- [4] Yuexi Gu, Jian Zu, and Yu Li. A novel evolutionary model for constructing gene coexpression networks with comprehensive features. *BMC bioinformatics*, 20:1–20, 2019.
- [5] Tiago P Peixoto. Network reconstruction and community detection from dynamics. *Physical review letters*, 123(12):128301, 2019.
- [6] Chunlin Yi, Lee W Cohnstaedt, and Caterina M Scoglio. Seir-sei-enkf: A new model for estimating and forecasting dengue outbreak dynamics. *IEEE Access*, 9:156758–156767, 2021.
- [7] Faryad Darabi Sahneh, Aram Vajdi, Heman Shakeri, Futing Fan, and Caterina Scoglio. Gemfsim: A stochastic simulator for the generalized epidemic modeling framework. *Journal of computational science*, 22:36–44, 2017.
- [8] Toshikazu Kuniya. Prediction of the epidemic peak of coronavirus disease in japan, 2020. *Journal of clinical medicine*, 9(3):789, 2020.
- [9] Mingwang Shen, Zhihang Peng, Yanni Xiao, and Lei Zhang. Modeling the epidemic trend of the 2019 novel coronavirus outbreak in china. *The Innovation*, 1(3):100048,

2020. ISSN 2666-6758. doi: <https://doi.org/10.1016/j.xinn.2020.100048>. URL <https://www.sciencedirect.com/science/article/pii/S2666675820300515>.
- [10] Jin-Yan Li, Zhi You, Qiong Wang, Zhi-Jian Zhou, Ye Qiu, Rui Luo, and Xing-Yi Ge. The epidemic of 2019-novel-coronavirus (2019-ncov) pneumonia and insights for emerging infectious diseases in the future. *Microbes and Infection*, 22(2):80–85, 2020. ISSN 1286-4579. doi: <https://doi.org/10.1016/j.micinf.2020.02.002>. URL <https://www.sciencedirect.com/science/article/pii/S1286457920300307>. Special issue on the new coronavirus causing the COVID-19 outbreak.
- [11] Jean-Gabriel Young, Alec Kirkley, and MEJ Newman. Clustering of heterogeneous populations of networks. *Physical Review E*, 105(1):014312, 2022.
- [12] M. E. J. Newman. Estimating network structure from unreliable measurements. *Phys. Rev. E*, 98:062321, Dec 2018. doi: [10.1103/PhysRevE.98.062321](https://doi.org/10.1103/PhysRevE.98.062321). URL <https://link.aps.org/doi/10.1103/PhysRevE.98.062321>.
- [13] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569.
- [14] Ron Bekkerman, Shlomo Zilberstein, and James Allan. Web page clustering using heuristic search in the web graph. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, page 2280–2285, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
- [15] Marc Timme. Revealing network connectivity from response dynamics. *Phys. Rev. Lett.*, 98:224101, May 2007. doi: [10.1103/PhysRevLett.98.224101](https://doi.org/10.1103/PhysRevLett.98.224101). URL <https://link.aps.org/doi/10.1103/PhysRevLett.98.224101>.
- [16] Srinivas Gorur Shandilya and Marc Timme. Inferring network topology from complex

- dynamics. *New Journal of Physics*, 13(1):013004, jan 2011. doi: 10.1088/1367-2630/13/1/013004. URL <https://dx.doi.org/10.1088/1367-2630/13/1/013004>.
- [17] P. Van Mieghem and Qiang Liu. Explicit non-markovian susceptible-infected-susceptible mean-field epidemic threshold for weibull and gamma infections but poisson curings. *Phys. Rev. E*, 100:022317, Aug 2019. doi: 10.1103/PhysRevE.100.022317. URL <https://link.aps.org/doi/10.1103/PhysRevE.100.022317>.
 - [18] S. Boccaletti, G. Bianconi, R. Criado, C.I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014. ISSN 0370-1573. doi: <https://doi.org/10.1016/j.physrep.2014.07.001>. URL <https://www.sciencedirect.com/science/article/pii/S0370157314002105>. The structure and dynamics of multilayer networks.
 - [19] Mary-Ellen Lynall, Danielle S Bassett, Robert Kerwin, Peter J McKenna, Manfred Kitzbichler, Ulrich Muller, and Ed Bullmore. Functional connectivity and brain networks in schizophrenia. *Journal of Neuroscience*, 30(28):9477–9487, 2010.
 - [20] Sebastian Raimondo and Manlio De Domenico. Measuring topological descriptors of complex networks under uncertainty. *Physical Review E*, 103(2):022311, 2021.
 - [21] Caterina De Bacco, Eleanor A Power, Daniel B Larremore, and Cristopher Moore. Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, 95(4):042317, 2017.
 - [22] George Sugihara, Robert May, Hao Ye, Chih-hao Hsieh, Ethan Deyle, Michael Fogarty, and Stephan Munch. Detecting causality in complex ecosystems. *science*, 338(6106):496–500, 2012.
 - [23] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

- [24] Barbara Benigni, Arsham Ghavasieh, Alessandra Corso, Valeria d’Andrea, and Manlio De Domenico. Persistence of information flow: A multiscale characterization of human brain. *Network Neuroscience*, 5(3):831–850, 2021.
- [25] Jonathan Schiefer, Alexander Niederbühl, Volker Pernice, Carolin Lennartz, Jürgen Hennig, Pierre LeVan, and Stefan Rotter. From correlation to causation: Estimating effective connectivity from zero-lag covariances of brain signals. *PLoS computational biology*, 14(3):e1006056, 2018.
- [26] Jaeseung Jeong, John C Gore, and Bradley S Peterson. Mutual information analysis of the eeg in patients with alzheimer’s disease. *Clinical neurophysiology*, 112(5):827–835, 2001.
- [27] Ali Namaki, Amir H Shirazi, R Raei, and GR Jafari. Network analysis of a financial market based on genuine correlation and threshold method. *Physica A: Statistical Mechanics and its Applications*, 390(21-22):3835–3841, 2011.
- [28] Kazuko Yamasaki, Avi Gozolchiani, and Shlomo Havlin. Climate networks around the globe are significantly affected by el nino. *Physical review letters*, 100(22):228501, 2008.
- [29] Jonathan F Donges, Yong Zou, Norbert Marwan, and Jürgen Kurths. The backbone of the climate network. *EPL (Europhysics Letters)*, 87(4):48007, 2009.
- [30] Jonathan F Donges, Yong Zou, Norbert Marwan, and Jürgen Kurths. Complex networks in climate dynamics. *The European Physical Journal Special Topics*, 174(1):157–179, 2009.
- [31] Steve Horvath and Jun Dong. Geometric interpretation of gene coexpression network analysis. *PLoS computational biology*, 4(8):e1000117, 2008.
- [32] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

- [33] F Mosele, J Remon, J Mateo, CB Westphalen, Fabrice Barlesi, MP Lolkema, N Normanno, A Scarpa, Mark Robson, F Meric-Bernstam, et al. Recommendations for the use of next-generation sequencing (ngs) for patients with metastatic cancers: a report from the esmo precision medicine working group. *Annals of Oncology*, 31(11): 1491–1505, 2020.
- [34] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1), 2005. doi: doi:10.2202/1544-6115.1128. URL <https://doi.org/10.2202/1544-6115.1128>.
- [35] Jonathan A Seaman, Haoues Alout, Jacob I Meyers, Mark D Stenglein, Roch K Dabiré, Saul Lozano-Fuentes, Timothy A Burton, Wojtek S Kuklinski, William C Black, and Brian D Foy. Age and prior blood feeding of anopheles gambiae influences their susceptibility and gene expression patterns to ivermectin-containing blood meals. *BMC genomics*, 16(1):1–18, 2015.
- [36] Anastasios C Koutsos, Claudia Blass, Stephan Meister, Sabine Schmidt, Robert M MacCallum, Marcelo B Soares, Frank H Collins, Vladimir Benes, Evgeny Zdobnov, Fotis C Kafatos, et al. Life cycle transcriptome of the malaria mosquito anopheles gambiae and comparison with the fruitfly drosophila melanogaster. *Proceedings of the National Academy of Sciences*, 104(27):11304–11309, 2007.
- [37] Bryan J Cassone, Karine Mouline, Matthew W Hahn, Bradley J White, Marco Pombi, Frederic Simard, Carlo Costantini, and Nora J Besansky. Differential gene expression in incipient species of anopheles gambiae. *Molecular ecology*, 17(10):2491–2504, 2008.
- [38] Osvaldo Marinotti, Eric Calvo, Quang K Nguyen, Sumudu Dissanayake, JMC Ribeiro, and Anthony A James. Genome-wide analysis of gene expression in adult anopheles gambiae. *Insect molecular biology*, 15(1):1–12, 2006.
- [39] Yury Goltsev, Gustavo L Rezende, Karen Vranizan, Greg Lanzaro, Denise Valle, and

- Michael Levine. Developmental and evolutionary basis for drought tolerance of the anopheles gambiae embryo. *Developmental biology*, 330(2):462–470, 2009.
- [40] Antonio M Mendes, Parfait H Awono-Ambene, Sandrine E Nsango, Anna Cohuet, Didier Fontenille, Fotis C Kafatos, George K Christophides, Isabelle Morlais, and Dina Vlachou. Infection intensity-dependent responses of anopheles gambiae to the african malaria parasite plasmodium falciparum. *Infection and immunity*, 79(11):4708–4715, 2011.
- [41] Bryan J Cassone, Matthew J Molloy, Changde Cheng, John C Tan, Matthew W Hahn, and Nora J Besansky. Divergent transcriptional response to thermal stress by anopheles gambiae larvae carrying alternative arrangements of inversion 2la. *Molecular ecology*, 20(12):2567–2580, 2011.
- [42] Dean A Baker, Tony Nolan, Bettina Fischer, Alex Pinder, Andrea Crisanti, and Steven Russell. A comprehensive gene expression atlas of sex-and tissue-specificity in the malaria vector, anopheles gambiae. *BMC genomics*, 12(1):1–12, 2011.
- [43] Samuel SC Rund, Tim Y Hou, Sarah M Ward, Frank H Collins, and Giles E Duffield. Genome-wide profiling of diel and circadian gene expression in the malaria vector anopheles gambiae. *Proceedings of the National Academy of Sciences*, 108(32):E421–E430, 2011.
- [44] PE Cook and SP Sinkins. Transcriptional profiling of anopheles gambiae mosquitoes for adult age estimation. *Insect molecular biology*, 19(6):745–751, 2010.
- [45] Mei-Hui Wang, Osvaldo Marinotti, Anne Vardo-Zalik, Rajni Boparai, and Guiyun Yan. Genome-wide transcriptional analysis of genes associated with acute desiccation stress in anopheles gambiae. *PLoS one*, 6(10):e26011, 2011.
- [46] Dina Vlachou, Timm Schlegelmilch, George K Christophides, and Fotis C Kafatos. Functional genomic analysis of midgut epithelial responses in anopheles during plasmodium invasion. *Current Biology*, 15(13):1185–1195, 2005.

- [47] Patricia Abrantes, George Dimopoulos, Ana Rita Grosso, Virgílio E Do Rosário, and Henrique Silveira. Chloroquine mediated modulation of anopheles gambiae gene expression. *PLoS One*, 3(7):e2587, 2008.
- [48] M Neira Oviedo, JMC Ribeiro, A Heyland, L VanEkeris, T Moroz, and PJ Linser. The salivary transcriptome of anopheles gambiae (diptera: Culicidae) larvae: A microarray-based analysis. *Insect biochemistry and molecular biology*, 39(5-6):382–394, 2009.
- [49] M Neira Oviedo, L Vanekeris, MDP Corena-Mcleod, and PJ Linser. A microarray-based analysis of transcriptional compartmentalization in the alimentary canal of anopheles gambiae (diptera: Culicidae) larvae. *Insect molecular biology*, 17(1):61–72, 2008.
- [50] David W Rogers, Miranda MA Whitten, Janis Thailayil, Julien Soichot, Elena A Levashina, and Flaminia Catteruccia. Molecular and cellular components of the mating machinery in anopheles gambiae females. *Proceedings of the National Academy of Sciences*, 105(49):19390–19395, 2008.
- [51] Sofia B Pinto, Fabrizio Lombardo, Anastasios C Koutsos, Robert M Waterhouse, Krista McKay, Chunju An, Chandra Ramakrishnan, Fotis C Kafatos, and Kristin Michel. Discovery of plasmodium modulators by genome-wide analysis of circulating hemocytes in anopheles gambiae. *Proceedings of the National Academy of Sciences*, 106(50):21270–21275, 2009.
- [52] Yang O Zhao, Sebastian Kurscheid, Yue Zhang, Lei Liu, Lili Zhang, Kelsey Loeliger, and Erol Fikrig. Enhanced survival of plasmodium-infected mosquitoes during starvation. *PLoS One*, 7(7):e40556, 2012.
- [53] W Robert Shaw, Eleonora Teodori, Sara N Mitchell, Francesco Baldini, Paolo Gabrieli, David W Rogers, and Flaminia Catteruccia. Mating activates the heme peroxidase hpx15 in the sperm storage organ to ensure fertility in anopheles gambiae. *Proceedings of the National Academy of Sciences*, 111(16):5854–5859, 2014.

- [54] Paolo Gabrieli, Evdoxia G Kakani, Sara N Mitchell, Enzo Mameli, Elizabeth J Want, Ainhoa Mariezcurrena Anton, Aurelio Serrao, Francesco Baldini, and Flaminia Catteruccia. Sexual transfer of the steroid hormone 20e induces the postmating switch in *anopheles gambiae*. *Proceedings of the National Academy of Sciences*, 111(46):16353–16358, 2014.
- [55] Rachel M Kwiatkowska, Naomi Platt, Rodolphe Poupardin, Helen Irving, Roch K Dabire, Sara Mitchell, Christopher M Jones, Abdoulaye Diabaté, Hilary Ranson, and Charles S Wondji. Dissecting the mechanisms responsible for the multiple insecticide resistance phenotype in *anopheles gambiae* ss, m form, from vallee du kou, burkina faso. *Gene*, 519(1):98–106, 2013.
- [56] Billy Fossog Tene, Rodolphe Poupardin, Carlo Costantini, Parfait Awono-Ambene, Charles S Wondji, Hilary Ranson, and Christophe Antonio-Nkondjio. Resistance to ddt in an urban setting: common mechanisms implicated in both m and s forms of *anopheles gambiae* in the city of yaoundé cameroon. *PloS one*, 8(4):e61408, 2013.
- [57] Craig S Wilding, David Weetman, EJ Rippon, Keith Steen, HD Mawejje, I Barsukov, and MJ Donnelly. Parallel evolution or purifying selection, not introgression, explains similarity in the pyrethroid detoxification linked *gste4* of *anopheles gambiae* and *an. arabiensis*. *Molecular genetics and genomics*, 290(1):201–215, 2015.
- [58] Kalle Magnusson, Antonio M Mendes, Nikolai Windbichler, Philippos-Aris Papatheos, Tony Nolan, Tania Dottorini, Ermanno Rizzi, George K Christophides, and Andrea Crisanti. Transcription regulation of sex-biased genes during ontogeny in the malaria vector *anopheles gambiae*. *PloS one*, 6(6):e21572, 2011.
- [59] Alison T Isaacs, Henry D Mawejje, Sean Tomlinson, Daniel J Rigden, and Martin J Donnelly. Genome-wide transcriptional analyses in *anopheles* mosquitoes reveal an unexpected association between salivary gland gene expression and insecticide resistance. *BMC genomics*, 19(1):1–12, 2018.

- [60] Laura Vannini, W Augustine Dunn, Tyler W Reed, and Judith H Willis. Changes in transcript abundance for cuticular proteins and other genes three hours after a blood meal in *anopheles gambiae*. *Insect biochemistry and molecular biology*, 44:33–43, 2014.
- [61] Edward A Mead, Meng Li, Zhijian Tu, and Jinsong Zhu. Translational regulation of *anopheles gambiae* mrnas in the midgut during *plasmodium falciparum* infection. *BMC genomics*, 13(1):1–10, 2012.
- [62] Francesco Papa, Nikolai Windbichler, Robert M Waterhouse, Alessia Cagnetti, Rocco D’Amato, Tania Persampieri, Mara KN Lawniczak, Tony Nolan, and Philippos Aris Papathanos. Rapid evolution of female-biased genes among four species of *anopheles malaria* mosquitoes. *Genome research*, 27(9):1536–1548, 2017.
- [63] S Noushin Emami, Bo G Lindberg, Susanna Hua, Sharon R Hill, Raimondas Mozuraitis, Philipp Lehmann, Göran Birgersson, Anna-Karin Borg-Karlson, Rickard Ignell, and Ingrid Faye. A key malaria metabolite modulates vector blood seeking, feeding, and susceptibility to infection. *Science*, 355(6329):1076–1080, 2017.
- [64] Gloria I Giraldo-Calderón, Scott J Emrich, Robert M MacCallum, Gareth Maslen, Emmanuel Dialynas, Pantelis Topalis, Nicholas Ho, Sandra Gesing, VectorBase Consortium, Gregory Madey, et al. Vectorbase: an updated bioinformatics resource for invertebrate vectors and other organisms related with human diseases. *Nucleic acids research*, 43(D1):D707–D713, 2015.
- [65] Beatrice Amos, Cristina Aurrecoechea, Matthieu Barba, Ana Barreto, Evelina Y Basenko, Robert Belnap, Ann S Blevins, Ulrike Böhme, John Brestelli, Brian P Brunk, et al. Veupathdb: the eukaryotic pathogen, vector and host bioinformatics resource center. *Nucleic Acids Research*, 50(D1):D898–D911, 2022.
- [66] Avcl consortium (2014) ncbi bioproject id 238805. broad institute: Umbrella comparative genomics project (subtype:comparative genomics). <https://www.ncbi.nlm.nih.gov/bioproject/238805>.

- [67] Mark S. Handcock and Krista J. Gile. Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1):5 – 25, 2010. doi: 10.1214/08-AOAS221. URL <https://doi.org/10.1214/08-AOAS221>.
- [68] Dean Lusher, Johan Koskinen, and Garry Robins. *Exponential random graph models for social networks: Theory, methods, and applications*. Cambridge University Press, 2013.
- [69] Carter T. Butts. Network inference, error, and informant (in)accuracy: a bayesian approach. *Social Networks*, 25(2):103–140, 2003. ISSN 0378-8733. doi: [https://doi.org/10.1016/S0378-8733\(02\)00038-2](https://doi.org/10.1016/S0378-8733(02)00038-2). URL <https://www.sciencedirect.com/science/article/pii/S0378873302000382>.
- [70] Peter Killworth and H Bernard. Informant accuracy in social network data. *Human Organization*, 35(3):269–286, 1976.
- [71] Peter V Marsden. Network data and measurement. *Annual review of sociology*, pages 435–463, 1990.
- [72] Nicole Bohme Carnegie. Effects of contact network structure on epidemic transmission trees: implications for data required to estimate network structure. *Statistics in medicine*, 37(2):236–248, 2018.
- [73] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10:255–268, 2006.
- [74] Junyao Kuang, Nicolas Buchon, Kristin Michel, and Caterina Scoglio. A global anopheles gambiae gene co-expression network constructed from hundreds of experimental conditions with missing values. *BMC bioinformatics*, 23(1):1–27, 2022.
- [75] Junyao Kuang and Caterina Scoglio. Layer reconstruction and missing link prediction of a multilayer network with maximum a posteriori estimation. *Physical Review E*, 104(2):024301, 2021.

- [76] Junyao Kuang, Caterina Scoglio, and Kristin Michel. Feature learning and network structure from noisy node activity data. *Physical Review E*, 106(6):064301, 2022.
- [77] Qihui Yang, Don Gruenbacher, Jessica L Heier Stamm, Gary L Brase, Scott A DeLoach, David E Amrine, and Caterina Scoglio. Developing an agent-based model to simulate the beef cattle production and transportation in southwest kansas. *Physica A: Statistical Mechanics and its Applications*, 526:120856, 2019.
- [78] Qihui Yang, Don M Gruenbacher, Jessica L Heier Stamm, David E Amrine, Gary L Brase, Scott A DeLoach, and Caterina M Scoglio. Impact of truck contamination and information sharing on foot-and-mouth disease spreading in beef cattle production systems. *Plos one*, 15(10):e0240819, 2020.
- [79] Deisy Morselli Gysi, Ítalo Do Valle, Marinka Zitnik, Asher Ameli, Xiao Gan, Onur Varol, Susan Dina Ghiassian, JJ Patten, Robert A Davey, Joseph Loscalzo, et al. Network medicine framework for identifying drug-repurposing opportunities for covid-19. *Proceedings of the National Academy of Sciences*, 118(19):e2025581118, 2021.
- [80] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics reports*, 544(1):1–122, 2014.
- [81] Mikko Kivelä, Alex Arenas, Marc Barthélemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.
- [82] Roger Guimerà and Marta Sales-Pardo. Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52):22073–22078, 2009.
- [83] Sipko Van Dam, Urmo Vosa, Adriaan van der Graaf, Lude Franke, and Joao Pe-

- dro de Magalhaes. Gene co-expression analysis for functional classification and gene-disease predictions. *Briefings in bioinformatics*, 19(4):575–592, 2018.
- [84] Franziska Liesecke, Johan-Owen De Craene, Sébastien Besseau, Vincent Courdavault, Marc Clastre, Valentin Vergès, Nicolas Papon, Nathalie Giglioli-Guivarc’h, Gaëlle Glévarec, Olivier Pichon, et al. Improved gene co-expression network quality through expression dataset down-sampling and network aggregation. *Scientific reports*, 9(1):1–16, 2019.
- [85] Alfredo Braunstein, Alessandro Ingrosso, and Anna Paola Muntoni. Network reconstruction from infection cascades. *Journal of the Royal Society Interface*, 16(151):20180844, 2019.
- [86] Xiao Han, Zhesi Shen, Wen-Xu Wang, and Zengru Di. Robust reconstruction of complex networks from sparse data. *Physical review letters*, 114(2):028701, 2015.
- [87] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.
- [88] Brian Ball, Brian Karrer, and Mark EJ Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103, 2011.
- [89] Tiago P Peixoto. Reconstructing networks with unknown and heterogeneous errors. *Physical Review X*, 8(4):041011, 2018.
- [90] Mark EJ Newman. Estimating network structure from unreliable measurements. *Physical Review E*, 98(6):062321, 2018.
- [91] Mark EJ Newman and Elizabeth A Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569, 2007.
- [92] Piet Van Mieghem and Qiang Liu. Explicit non-markovian susceptible-infected-susceptible mean-field epidemic threshold for weibull and gamma infections but poisson curings. *Physical Review E*, 100(2):022317, 2019.

- [93] Tiago P Peixoto. Bayesian stochastic blockmodeling. *Advances in network clustering and blockmodeling*, pages 289–332, 2019.
- [94] Federica Parisi, Guido Caldarelli, and Tiziano Squartini. Entropy-based approach to missing-links prediction. *Applied Network Science*, 3(1):1–15, 2018.
- [95] Likang Yin, Haoyang Zheng, Tian Bian, and Yong Deng. An evidential link prediction method and link predictability based on shannon entropy. *Physica A: Statistical Mechanics and its Applications*, 482:699–712, 2017.
- [96] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.
- [97] Tiago P Peixoto. Nonparametric weighted stochastic block models. *Physical Review E*, 97(1):012306, 2018.
- [98] Elizabeth Costenbader and Thomas W Valente. The stability of centrality measures when networks are sampled. *Social networks*, 25(4):283–307, 2003.
- [99] Danai Koutra, Neil Shah, Joshua T Vogelstein, Brian Gallagher, and Christos Faloutsos. Deltacon: Principled massive-graph similarity function with attribution. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(3):1–43, 2016.
- [100] Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. Comparing methods for comparing networks. *Scientific reports*, 9(1):1–19, 2019.
- [101] Panagiotis Papadimitriou, Ali Dasdan, and Hector Garcia-Molina. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications*, 1(1):19–30, 2010.
- [102] MS Charikar. Proceedings of the thirty-fourth annual acm symposium on theory of computing, 2002.

- [103] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [104] Michael C-K Wu, Fatma Deniz, Ryan J Prenger, and Jack L Gallant. The unified maximum a posteriori (map) framework for neuronal system identification. *arXiv preprint arXiv:1811.01043*, 2018.
- [105] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature communications*, 6(1):1–9, 2015.
- [106] Vincent B Chen, Ian W Davis, and David C Richardson. King (kinemage, next generation): a versatile interactive molecular and scientific visualization program. *Protein science*, 18(11):2403–2409, 2009.
- [107] Daniel B Larremore, Aaron Clauset, and Caroline O Buckee. A network approach to analyzing highly recombinant malaria parasite genes. *PLoS computational biology*, 9(10):e1003268, 2013.
- [108] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.
- [109] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.
- [110] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.
- [111] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
- [112] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

- [113] Homin K Lee, Amy K Hsu, Jon Sajdak, Jie Qin, and Paul Pavlidis. Coexpression analysis of human genes across many microarray data sets. *Genome research*, 14(6): 1085–1094, 2004.
- [114] Guillermo de Anda-Jáuregui, Sergio Antonio Alcalá-Corona, Jesús Espinal-Enríquez, and Enrique Hernández-Lemus. Functional and transcriptional connectivity of communities in breast cancer co-expression networks. *Applied Network Science*, 4(1):1–13, 2019.
- [115] K Ovens, B Frank Eames, and Ian McQuillan. The impact of sample size and tissue type on the reproducibility of gene co-expression networks. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–10, 2020.
- [116] Jean-Philippe David, Clare Strode, John Vontas, Dimitra Nikou, Ashley Vaughan, Patricia M Pignatelli, Christos Louis, Janet Hemingway, and Hilary Ranson. The anopheles gambiae detoxification chip: a highly specific microarray to study metabolic-based insecticide resistance in malaria vectors. *Proceedings of the National Academy of Sciences*, 102(11):4080–4084, 2005.
- [117] Antonio Reverter and Eva KF Chan. Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. *Bioinformatics*, 24(21):2491–2497, 2008.
- [118] Riann N Christian, C Strode, H Ranson, Nanette Coetzer, M Coetzee, and Lizette L Koekemoer. Microarray analysis of a pyrethroid resistant african malaria vector, anopheles funestus, from southern africa. *Pesticide biochemistry and physiology*, 99(2):140–147, 2011.
- [119] Rute C Félix, Pie Müller, Vera Ribeiro, Hilary Ranson, and Henrique Silveira. Plasmodium infection alters anopheles gambiae detoxification gene expression. *BMC genomics*, 11(1):1–10, 2010.

- [120] Pie Müller, Martin J Donnelly, and Hilary Ranson. Transcription profiling of a recently colonised pyrethroid resistant anopheles gambiae strain from ghana. *BMC genomics*, 8(1):1–12, 2007.
- [121] Pie Müller, Emma Warr, Bradley J Stevenson, Patricia M Pignatelli, John C Morgan, Andrew Steven, Alexander E Yawson, Sara N Mitchell, Hilary Ranson, Janet Hemingway, et al. Field-caught permethrin-resistant anopheles gambiae overexpress cyp6p3, a p450 that metabolises pyrethroids. *PLoS genetics*, 4(11):e1000286, 2008.
- [122] BL Spillings. Insecticide resistance and bionomics of laboratory and field caught anopheles funestus giles (diptera: Culicidae), 2012.
- [123] Lyric C Bartholomay and Kristin Michel. Mosquito immunobiology: the intersection of vector health and vector competence. *Annual review of entomology*, 63:145–167, 2018.
- [124] Alexander WE Franz, Asher M Kantor, A Lorena Passarelli, and Rollie J Clem. Tissue barriers to arbovirus infection in mosquitoes. *Viruses*, 7(7):3741–3767, 2015.
- [125] Eappen G Abraham and Marcelo Jacobs-Lorena. Mosquito midgut barriers to malaria parasite development. *Insect biochemistry and molecular biology*, 34(7):667–671, 2004.
- [126] Priit Adler, Raivo Kolde, Meelis Kull, Aleksandr Tkachenko, Hedi Peterson, Jüri Reimand, and Jaak Vilo. Mining for coexpression across hundreds of datasets using novel rank aggregation and visualization methods. *Genome biology*, 10(12):1–11, 2009.
- [127] Kestutis Baltakys, Juho Kanninen, and Frank Emmert-Streib. Multilayer aggregation with statistical validation: Application to investor networks. *Scientific reports*, 8(1):1–12, 2018.
- [128] Junyao Kuang and Caterina Scoglio. A principled approach for weighted multilayer network aggregation. *arXiv preprint arXiv:2103.05774*, 2021.

- [129] Kathryn Dempsey and Hesham Ali. Evaluation of essential genes in correlation networks using measures of centrality. In *2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 509–515. IEEE, 2011.
- [130] Francisco J Azuaje. Selecting biologically informative genes in co-expression networks with a centrality score. *Biology direct*, 9(1):1–23, 2014.
- [131] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.
- [132] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.
- [133] Olaf Sporns and Jonathan D Zwi. The small world of the cerebral cortex. *Neuroinformatics*, 2(2):145–162, 2004.
- [134] Shouguo Gao, Zhijie Wu, Xingmin Feng, Sachiko Kajigaya, Xujing Wang, and Neal S Young. Comprehensive network modeling from single cell rna sequencing of human and mouse reveals well conserved transcription regulation of hematopoiesis. *BMC genomics*, 21(11):1–15, 2020.
- [135] Vera Van Noort, Berend Snel, and Martijn A Huynen. The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model. *EMBO reports*, 5(3):280–284, 2004.
- [136] Francisco Aparecido Rodrigues. Network centrality: an introduction. In *A mathematical modeling approach from nonlinear dynamics to complex systems*, pages 177–196. Springer, 2019.
- [137] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PloS one*, 9(6):e98679, 2014.

- [138] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.
- [139] Marius Eidsaa and Eivind Almaas. S-core network decomposition: A generalization of k-core analysis to weighted networks. *Physical Review E*, 88(6):062819, 2013.
- [140] Edgardo Galán-Vásquez and Ernesto Perez-Rueda. Identification of modules with similar gene regulation and metabolic functions based on co-expression data. *Frontiers in molecular biosciences*, 6:139, 2019.
- [141] Chirag Gupta and Andy Pereira. Recent advances in gene function prediction using context-specific coexpression networks in plants. *F1000Research*, 8, 2019.
- [142] Adrian Alexa, Jorg Rahnenfuhrer, et al. topgo: enrichment analysis for gene ontology. *R package version*, 2(0):2010, 2010.
- [143] Bruno Arcà, Fabrizio Lombardo, Claudio J Struchiner, and Jose Ribeiro. Anopheline salivary protein genes and gene families: an evolutionary overview after the whole genome sequence of sixteen anopheles species. *BMC genomics*, 18(1):1–27, 2017.
- [144] Robert M Waterhouse, Evgenia V Kriventseva, Stephan Meister, Zhiyong Xi, Kanwal S Alvarez, Lyric C Bartholomay, Carolina Barillas-Mury, Guowu Bian, Stephanie Blandin, Bruce M Christensen, et al. Evolutionary dynamics of immune-related genes and pathways in disease-vector mosquitoes. *science*, 316(5832):1738–1743, 2007.
- [145] Xiaolong Cao, Mansi Gulati, and Haobo Jiang. Serine protease-related proteins in the malaria mosquito, anopheles gambiae. *Insect biochemistry and molecular biology*, 88: 48–62, 2017.
- [146] R Scott Cornman and Judith H Willis. Extensive gene amplification and concerted evolution within the cpr family of cuticular proteins in mosquitoes. *Insect biochemistry and molecular biology*, 38(6):661–676, 2008.

- [147] John C Doyle, David L Alderson, Lun Li, Steven Low, Matthew Roughan, Stanislav Shalunov, Reiko Tanaka, and Walter Willinger. The “robust yet fragile” nature of the internet. *Proceedings of the National Academy of Sciences*, 102(41):14497–14502, 2005.
- [148] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *nature*, 406(6794):378–382, 2000.
- [149] Pascale Gaudet and Christophe Dessimoz. Gene ontology: pitfalls, biases, and remedies. In *The gene ontology handbook*, pages 189–205. Humana Press, New York, NY, 2017.
- [150] P Topalis, C Tzavlaki, K Vestaki, E Dialynas, DE Sonenshine, R Butler, RV Bruggner, EO Stinson, FH Collins, and C Louis. Anatomical ontologies of mosquitoes and ticks, and their web browsers in vectorbase. *Insect molecular biology*, 17(1):87–89, 2008.
- [151] Evgenia V Kriventseva, Dmitry Kuznetsov, Fredrik Tegenfeldt, Mosè Manni, Renata Dias, Felipe A Simão, and Evgeny M Zdobnov. Orthodb v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. *Nucleic acids research*, 47(D1):D807–D811, 2019.
- [152] C Mencarelli, P Lupetti, and R Dallai. New insights into the cell biology of insect axonemes. *International review of cell and molecular biology*, 268:95–145, 2008.
- [153] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szczesniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):1–19, 2016.
- [154] Phillip Bonacich. Some unique properties of eigenvector centrality. *Social networks*, 29(4):555–564, 2007.
- [155] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.

- [156] Yannick Rochat. Closeness centrality extended to unconnected graphs: The harmonic centrality index. Technical report, 2009.
- [157] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.
- [158] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of modern physics*, 87(3):925, 2015.
- [159] Jean-Gabriel Young, George T Cantwell, and MEJ Newman. Bayesian inference of network structure from unreliable data. *Journal of Complex Networks*, 8(6):cnaa046, 2020.
- [160] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.
- [161] Srinivas Gorur Shandilya and Marc Timme. Inferring network topology from complex dynamics. *New Journal of Physics*, 13(1):013004, 2011.
- [162] István A Kovács, Dániel L Barabási, and Albert-László Barabási. Uncovering the genetic blueprint of the c. elegans nervous system. *Proceedings of the National Academy of Sciences*, 117(52):33570–33577, 2020.
- [163] Michael C Oldham, Steve Horvath, and Daniel H Geschwind. Conservation and evolution of gene coexpression networks in human and chimpanzee brains. *Proceedings of the National Academy of Sciences*, 103(47):17973–17978, 2006.
- [164] Lin Song, Peter Langfelder, and Steve Horvath. Comparison of co-expression measures: mutual information, correlation, and model based indices. *BMC bioinformatics*, 13(1):1–21, 2012.

- [165] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Dalla Favera, and Andrea Califano. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, volume 7, pages 1–15. BioMed Central, 2006.
- [166] Ricardo Ramirez, Yu-Chiao Chiu, Allen Herrera, Milad Mostavi, Joshua Ramirez, Yidong Chen, Yufei Huang, and Yu-Fang Jin. Classification of cancer types using graph convolutional neural networks. *Frontiers in physics*, 8:203, 2020.
- [167] Jiří Kaiser. Dealing with missing values in data. *Journal of Systems Integration (1804-2724)*, 5(1), 2014.
- [168] Birgitta Dresch-Langley, Ole Kristian Ekseth, Jan Fesl, Seiichi Gohshi, Marc Kurz, and Hans-Werner Sehring. Occam’s razor for big data? on detecting quality in large unstructured datasets. *Applied Sciences*, 9(15):3065, 2019.
- [169] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
- [170] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.
- [171] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.
- [172] Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021.

- [173] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.
- [174] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.
- [175] Mi Jin Lee, Eun Lee, Byunghwee Lee, Hawoong Jeong, Deok-Sun Lee, Sang Hoon Lee, et al. Uncovering hidden dependency in weighted networks via information entropy. *Physical Review Research*, 3(4):043136, 2021.
- [176] Ahmad Shamshad, MA Bawadi, WMA Wan Hussin, Taksiah A Majid, and SAM Sanusi. First and second order markov chain models for synthetic generation of wind speed time series. *Energy*, 30(5):693–708, 2005.
- [177] Oren Barkan and Noam Koenigstein. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2016.
- [178] Mihajlo Grbovic and Haibin Cheng. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 311–320, 2018.
- [179] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. corr abs/1802.05365 (2018). *arXiv preprint arXiv:1802.05365*, 1802.
- [180] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [181] DeLiang Wang. Unsupervised learning: foundations of neural computation. *Ai Magazine*, 22(2):101–101, 2001.

- [182] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2):187–212, 2022.
- [183] Mark O Hill. Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2):427–432, 1973.
- [184] Karol Życzkowski. Rényi extrapolation of shannon entropy. *Open Systems & Information Dynamics*, 10(3):297–310, 2003.
- [185] Robert J Hill. Superlative index numbers: not all of them are super. *Journal of Econometrics*, 130(1):25–43, 2006.
- [186] Anne Chao, Nicholas J Gotelli, TC Hsieh, Elizabeth L Sander, KH Ma, Robert K Colwell, and Aaron M Ellison. Rarefaction and extrapolation with hill numbers: a framework for sampling and estimation in species diversity studies. *Ecological monographs*, 84(1):45–67, 2014.
- [187] Lou Jost. Entropy and diversity. *Oikos*, 113(2):363–375, 2006.
- [188] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [189] Peter Filzmoser, Karel Hron, and Clemens Reimann. Principal component analysis for compositional data with outliers. *Environmetrics: The Official Journal of the International Environmetrics Society*, 20(6):621–632, 2009.
- [190] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.
- [191] Laurens Van Der Maaten. Learning a parametric embedding by preserving local structure. In *Artificial intelligence and statistics*, pages 384–391. PMLR, 2009.
- [192] Arnaud Barragao. The cora dataset. URL <https://relational.fit.cvut.cz/dataset/CORA>.

- [193] Jan Motl. The pubmed dataset. URL <https://relational.fit.cvut.cz/dataset/CORA>.
- [194] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, pages 4171–4186, 2019.
- [195] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Appendix A

Tensor factorization

In [21;87;88](#), nodes are factorized by membership vectors. The dimension of membership vectors is interpreted as the number of overlapping communities. In addition, each edge is computed through the tensor product of the membership vectors. The number of communities is obtained by maximizing the likelihood. Vectorization is also used in some natural language processing algorithms [173;174;179;194;195](#) in which words are embedded as vectors to preserve their relations to contexts. However, the dimension of word vectors does not have any semantic meaning, rather the choice of the vector dimension is first affected by the data set size. According to [174;195](#), larger dimensions can improve model accuracy, but the gain diminishes for vectors larger than 200 dimensions. The choice of dimension is also related to the available resources, and it is better to reduce the dimension as long as the choice does not affect the estimation accuracy substantially. In fact, the tensor factorization can be simplified to dot product. We prove this point as follows.

We assume the number of edges between any two nodes is the tensor product of two node vectors and the control matrix, i.e., $E_{ij} = \sum s_{ik}t_{jl}w_{kl}$. w_{kl} (k is not equal to l) is the parameter that controls the edges from any source node i (in community k) to any target

node j (in the community l). Then, we have

$$\begin{aligned}
E_{ij} &= \sum s_{ik} t_{jl} w_{kl} \\
&= s_{ik} t_{jk} w_{kk} + s_{ik} t_{jl} w_{kl} + s_{il} t_{jl} w_{ll} \\
&+ \left(\sum s_{im} t_{jn} w_{mn} - s_{ik} t_{jk} w_{kk} - s_{ik} t_{jl} w_{kl} - s_{il} t_{jl} w_{ll} \right). \tag{A.1}
\end{aligned}$$

In equation A.1, we denote the sum of the first three terms as E_{ij}^{kl} . Then, we can incorporate w_{kk} into s_{ik} , and denote it as $s'_{ik} = s_{ik} w_{kk}$. Similarly, $s'_{il} = s_{il} w_{ll}$. Therefore, we have

$$\begin{aligned}
E_{ij}^{kl} &= s'_{ik} t_{jk} + s'_{ik} t_{jl} \frac{w_{kl}}{w_{kk}} + s'_{il} t_{jl} \\
&= s'_{ik} t_{jk} + \left(s'_{ik} \frac{w_{kl}}{w_{kk}} + s'_{il} \right) t_{jl}. \tag{A.2}
\end{aligned}$$

Since nodes between different communities are loosely connected, we have $w_{kl} < w_{kk}$, and we assume $\epsilon = w_{kl}/w_{kk}$. Then, equation A.2 can be written as

$$\begin{aligned}
E_{ij}^{kl} &= s'_{ik} t_{jk} + (\epsilon s'_{ik} + s'_{il}) t_{jl} \\
&= s'_{ik} t_{jk} + s''_{il} t_{jl}, \tag{A.3}
\end{aligned}$$

where $s''_{il} = \epsilon s'_{ik} + s'_{il}$. For every inter-community edge originating from node i in community k , we can incorporate it into community l by incrementing ϵs_{ik} to s'_{il} . Edges can be factorized through the dot product of node vectors. Therefore, the tensor factorization is equivalent to the dot product.

Appendix B

Comparison between the MAP and MLE

B.1 Estimation based on the Pearson correlation

With the SimHash algorithm, the weighted digest of each layer can be obtained. The similarity between any two layers can be measured by computing the Pearson correlation of the weighted digests. In Fig. B.1, we show the estimation results based on the Pearson correlation coefficient. We observe that the estimation is as consistent as the results based on the Hamming distance. Therefore, the correlation can be an alternative for the Hamming distance.

B.2 The choice of the number of bits

The choice of ϕ affects the resolution and the consistency of the similarity. If the number of tokens is large, it is recommended to employ large ϕ . In our work, the sizes of the two networks are not too large, so we adopt $\phi = 512$ bits. In Fig. B.2, we validate this choice on the FAO network. We set each of the first nine layers of the FAO network as the

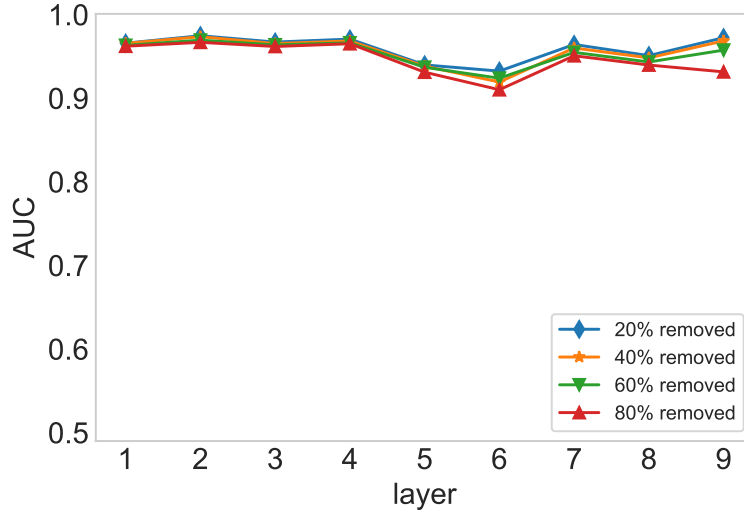


Figure B.1: *The estimation results are based on the Pearson correlation of the weighted digest.*

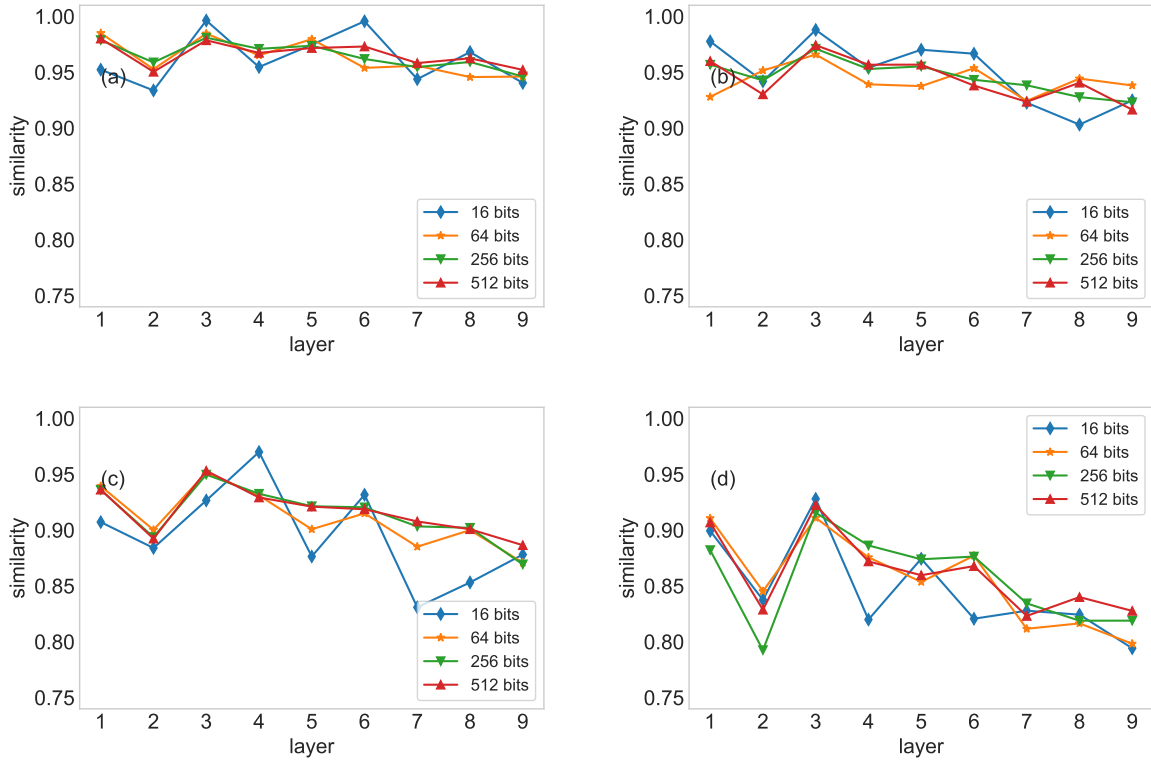


Figure B.2: *Study the choice of the number of digits (based on the Hamming distance). The four panels show the results based on 20% (a), 40% (b), 60% (c), and 80% (d) of edges randomly removed.*

target network (original network), and randomly remove 20%, 40%, 60% and 80% of edges from the target network to generate incomplete networks. We then compare the incomplete networks with the original network. In the experiments, we adopt $\phi = 16, 64, 256$, and 512 digits, respectively. We can observe that the similarities obtained with 256 digits are close to that obtained with 512 digits, while the similarities obtained with 16 and 64 digits vary remarkably.

B.3 The number of similar layers on the estimation accuracy

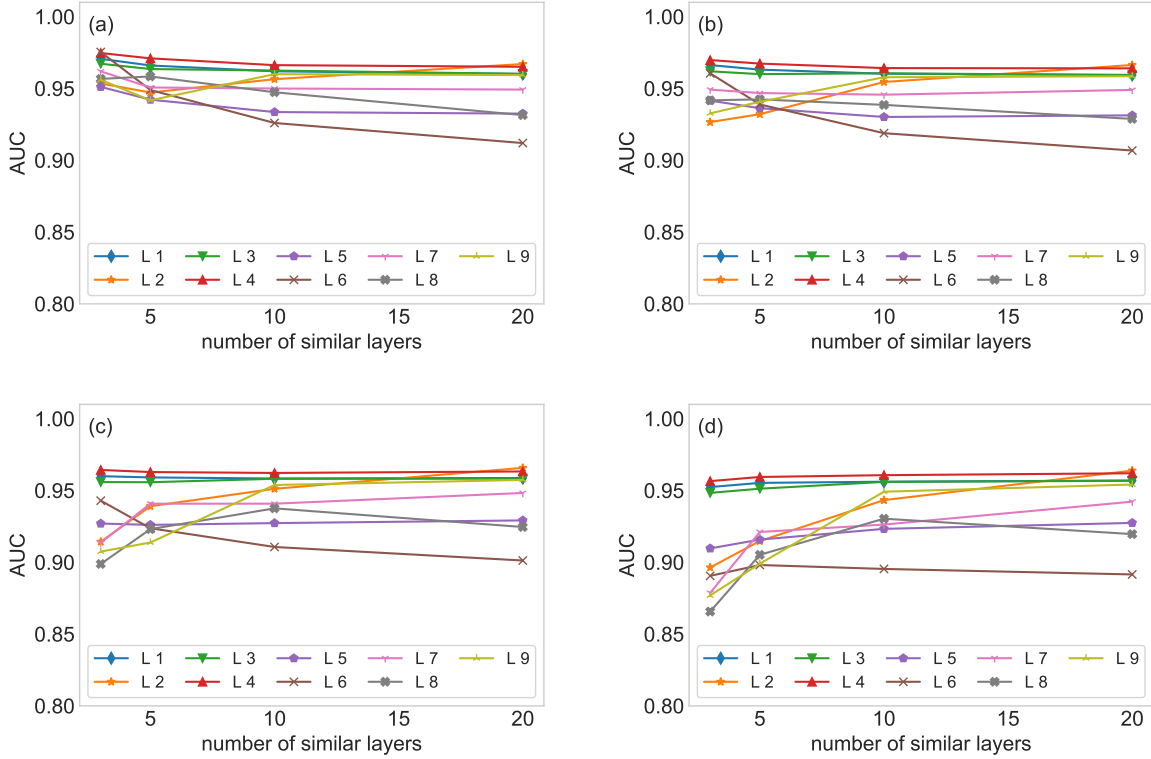


Figure B.3: The number of similar layers on the estimation results. 20% (a), 40% (b), 60% (c), and 80% (d) of edges are randomly removed from the target layer. Different numbers of similar layers are used to compute the parameters of the conjugate prior. The experiments are conducted on the first nine layers of the FAO network. Each AUC is averaged over 10 runs of cross-validations.

If more similar layers are employed to compute the parameters of the conjugate prior, the influence of similar layers on the reconstruction will be promoted. Consequently, the influence of the known part of the target layer will be down-weighted. On the contrary, if we trust the known part of the target layer, we can employ less similar layers to compute the parameters of the conjugate prior. The experimental results are shown in Fig. [B.3](#), top 3, 5, 10, and 20 similar layers are adopted to compute the parameters of the conjugate prior. We observe that the estimation based on 10 and 20 similar layers is robust with respect to the missing links, while the estimation based on three and five similar layers is influenced by the missing links significantly.

Appendix C

Comparison between the MLE algorithm and entropy-based approaches

In section 2.2.2, we compared the MAP algorithm with the MLE algorithm. In fact, the entropy-based approaches are equivalent to the MLE algorithm. In the following, we prove that the MLE algorithm is equivalent to the entropy-based approaches.

Given a network G , we assume the number of nodes is fixed, and the edge weights are variables. Thus, we can define the entropy of a network in terms of edge weights as

$$\begin{aligned} H(G) &= - \sum_G \sum_u p^U(u) \log p^U(u) \\ &= \sum_G \sum_u p^U(u) \log \frac{1}{p^U(u)}, \end{aligned} \tag{C.1}$$

where U is any edge in the network G , u is the weight of edge U . $p^U(u)$ is the precise probability of edge U with weight u . We assume for any edge U , it can take as many as n_U values. If the n_U values follow uniform distribution, then we have $p^U(u) = \frac{1}{n_U}$. In this case, the network has the largest entropy, which means the network is totally uncertain. On the other hand, for any edge U , if there is a value u_s with $p^U(u_s) = 1$, the entropy of the

network will be zero, which indicates all the edge weights in the network are known.

In our problem, the goal is to find a model to reconstruct the target layer. Thus, we need a set of parameters to describe the model. If the model is close to the true network, the entropy of the model is minimized. Hence, the problem is transformed into finding the parameters of a model with minimum entropy, i.e., reducing uncertainty.

Equation C.1 can be written in expectation form

$$H(p) = \sum_G E_{U \sim p^U(u)} \left(\log \frac{1}{p^U(u)} \right). \quad (\text{C.2})$$

Consider the Kullback-Leibler divergence (KL divergence). We assume θ is the parameter set that can describe the model, the probability of edge U with weight u is $q^U(u; \theta)$. Thus, the relative entropy is

$$\begin{aligned} D_{kl}(p||q) &= \sum_G \sum_u p^U(u) \log \frac{p^U(u)}{q^U(u; \theta)} \\ &= \sum_G E_{U \sim p^U(u)} \left(\log \frac{p^U(u)}{q^U(u; \theta)} \right) \\ &= \sum_G E_{U \sim p^U(u)} \left(\log \frac{1}{q^U(u; \theta)} \right) \\ &\quad - \sum_G E_{U \sim p^U(u)} \left(\log \frac{1}{p^U(u)} \right). \end{aligned} \quad (\text{C.3})$$

The second term in the right-hand side is equation C.2 and the first term in the right-hand side is the cross entropy, which is

$$H(p, q) = \sum_G E_{U \sim p^U(u)} \log \frac{1}{q^U(u; \theta)}. \quad (\text{C.4})$$

Recall that our problem is to reconstruct the target layer and equation C.2 is the entropy of the target layer. Equation C.2 is determined by the data (network) solely. Thus, the problem can be simplified to minimise the cross entropy, i.e., minimizing equation C.4.

Then, we consider the MLE method. The MLE algorithm is to find the parameter set θ

that most likely fit a given set of data (network) D . We skip some intermediate steps and use the logarithm form directly as following

$$\begin{aligned}\theta &= \operatorname{argmax}_{\theta} P(G \mid \theta) \\ &= \operatorname{argmax}_{\theta} \sum_G \sum_u \log p^U(u; \theta).\end{aligned}\tag{C.5}$$

$p^U(u; \theta)$ is the model with parameter set θ to describe the true data D . We assume that edge U can take n_U values and the n_U values follow the uniform distribution, then we have $p_{MLE}^U(u) = \frac{1}{n_U}$. Introducing p_{MLE}^U to equation C.5 does not change the results, we have

$$\begin{aligned}\theta &= \operatorname{argmax}_{\theta} \sum_G \sum_u \frac{1}{n_U} \log p^U(u; \theta) \\ &= \operatorname{argmax}_{\theta} \sum_G E_{u \sim p_{MLE}^U(u)} (\log p^U(u; \theta)).\end{aligned}\tag{C.6}$$

Our goal is still to find a set of parameters that can reconstruct the layer. Then, we can generalize this problem by replacing $p_{MLE}^U(u)$ with $p^U(u)$, and replacing $p^U(u; \theta)$ with $q^U(u; \theta)$. The replacements can be regarded as taking real data (network) into equation C.6. We have

$$\begin{aligned}\theta &= \operatorname{argmax}_{\theta} \sum_G E_{u \sim p^U(u)} \log q^U(u; \theta) \\ &= \operatorname{argmax}_{\theta} \sum_G -E_{u \sim p^U(u)} \log q^U(u; \theta) \\ &= \operatorname{argmax}_{\theta} \sum_G E_{u \sim p^U(u)} \log \frac{1}{q^U(u; \theta)}.\end{aligned}\tag{C.7}$$

Thus, the MLE algorithm is equivalent to minimizing the cross entropy.