DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF
TECHNOLOGY
MADRAS
CHENNAI-600 036

Rich Graph Structures: Algorithms and Applications

A thesis

Submitted by

Tarun Kumar

For the award of the degree

Of

DOCTOR OF PHILOSOPHY

April, 2023

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF
TECHNOLOGY
MADRAS
CHENNAI-600 036

Rich Graph Structures: Algorithms and Applications

A thesis

Submitted by

Tarun Kumar

For the award of the degree

Of

DOCTOR OF PHILOSOPHY

April, 2023

© 2023 Indian Institute of Technology Madras

THESIS CERTIFICATE

This is to undertake that the thesis titled, *Rich Graph Structures: Algorithms and Applications* submitted by me to the Indian Institute of Technology Madras, for the award of Ph.D is a bonafide record of the research work done by me under the supervision of Prof. Balaraman Ravindran and Prof. Manikandan Narayanan. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Place: Chennai 600 036
Date: 24 April 2023

Tarun Kumar
Research Scholar

Prof. Balaraman Ravindran
Research Guide

Prof. Manikandan Narayanan
Research Co-Guide

ACKNOWLEDGEMENTS

First and foremost, I am extremely grateful to my supervisors Prof. Balaraman Ravindran and Prof. Manikandan Narayanan. Their continuous support, invaluable advice, and everlasting patience have made this work possible. I have enjoyed listening to them during lectures, working with them as a teaching assistant, and learning from them during research meetings. Prof. Balaraman Ravindran has always encouraged me with a perfect blend of insight and humor. I have benefited greatly from his wealth of knowledge and how he presents complex ideas in a simple form. The guidance of Prof. Manikandan Narayanan from the initial step of research enabled me to develop an understanding of the subject. I am incredibly grateful that he took me on as a student and continued to have faith in me over the years. I am proud of, and thankful for, my time working under both my supervisors.

I want to thank my DC members - Prof. N.S. Narayanaswamy, Dr. Meghana Nasre, Prof. Rupesh Nasre and Dr. Venkatesh Ramaiyan. Their suggestions have helped in prioritizing my research ideas. During my Ph.D., I was fortunate to collaborate with Prof. Srinivasan Parthasarathy from The Ohio State University. I am thankful to him for teaching me the art of designing experiments. In the later part of my Ph.D., I got a chance to collaborate with Dr. Ramanathan Sethuraman from Intel research. I am thankful to him for helping me understand the industry value of my research. I extend my gratitude to Prof. Gitakrishnan Ramadurai for helping me know the implication of my research on other domains via the HIP seminar series. I would also like to express my gratitude to all my collaborators from IIT Madras. I want to thank Harini A. for

taking the time to explain the basics of hypergraphs over numerous phone calls. It was fun collaborating with Sankaran and Darwin on my work related to hypergraphs. I want to extend my gratitude to Dr. Sanga Mitra for contributing to my work on finding node centrality by adding biological interpretation of the results. I have learned a lot from the group discussions with RBCDSAI as well as BIRDS members. I would also like to acknowledge that I have learned immensely from my co-authors of works, not in this thesis - Aditya, Mohit, Tamizhini and Naganand.

I want to thank Deepak, Nikita, Naganand, Jeshuren, Aparna, Shreya and Anasua for their collaboration in organizing the GCLR workshop series at AAAI 2021, 2022, and 2023. I would also like to thank the advisors of this workshop series Prof. Kristian Kersting, Prof. Sarika Jalan, Prof. Partha Talukdar, Prof. Sriraam Natarajan, Prof. Ginestra Bianconi, Dr. Philip S. Chodrow, Prof. Srinivasan Parthasarathy and Prof. Balaraman Ravindran for their constant guidance and support.

I have made many friends at IIT Madras who made my stay wonderful and ensured my sanity while chasing deadlines. My labmates (ordered by their proximity to my lab seat) - Nikita, Shweta, Pritha and Nitesh (now Prithesh), Deepak Maurya, Priyesh, Ananya, Pranshu, Sidharth Nishthala, Sugyani, Kanika, Stanley and Shikha were always there to proofread my manuscripts, arrange food during deadlines, and bring on exciting discussions. Special mention of Maurya for (still) maintaining a list of pending treats. I also thank my hostel friends - Yashwant, Abhishek, Chiranth, Hari Ganesh, Gautam Shaw, Saikat, Vijay Atkan, Jhalu and Arshdeep. Outside IITM, I want to thank my friends Amit Sharma, Jatinder Bhola, Satish Verma, Jatin Gheek, Sukhwinder, Ashish

Chalana, Swasti, Arif, Jaymin, Manisha and Kajol for being part of my life and supporting me during different phases.

I believe that a Ph.D. scholar's family puts enormous effort during the course. My family deserves endless gratitude: my late father for making me humble and teaching me perseverance - much-required characteristics for pursuing a Ph.D., my mother - for her unfailing love, and my sisters - Dr. Anupam Gumber and Monika, for keeping my morale up throughout this journey.

I am thankful to RBCDSAI, RISE lab, DON lab and BIRDS group for providing me with the best work environment. I am also thankful to IIT Madras for providing me HTRA fellowship and Intel India for providing an Intel Ph.D. fellowship. I am grateful to Prof. Noshir Contractor for providing alums travel grant to attend International Conference for Complex Networks in Portugal. I am grateful to the CSE department office, Ramya, and RBCDSAI staff, especially Mr. Sridhar, for all the help in administrative tasks.

ABSTRACT

KEYWORDS: complex networks, multilayer networks, hypergraphs, inter tissue communication, node centrality, PageRank, hypergraph clustering

Network science provides a framework to model systems involving interacting components. These systems are observed in various domains such as genes interacting in tissues, railway junctions connected in transportation systems, multiple computers connected on the internet, humans involved in social relationships, researchers collaborating on projects, etc. There is a growing body of literature on graphs that represents such richer forms of interactions, and our thesis adds novel measures, algorithms, and network science applications, to this line of research, specifically pertaining to multilayer graphs and hypergraphs. In this thesis, we analyze these two rich graph structures, viz. multilayer networks and hypergraphs, and discuss their applications in modeling multi-tissue datasets and collaboration networks, respectively. In particular, we look at the problems of finding node centrality in multilayer networks and clustering and hyperedge prediction in hypergraphs.

We begin with providing the mathematical framework required to define multilayer networks and hypergraphs. We start with exploring different forms of multilayer networks based on coupling schemes and zoom into the most general form of multilayer network based on cross-coupling. We systematically study centrality methods for multilayer networks and discuss their applicability in different scenarios. Existing centrality measures for multilayer networks fail to distinguish between within-layer and across-layer edges' impact. Identifying across-layer central nodes is crucial in several systems. For example, discovering genes responsible for inter-tissue communication in a multi-tissue system can help answer interesting biological questions. We introduce MultiCens, a novel centrality framework that can distinguish within- vs. across-layer connectivity to quantify the "influence" of any gene in a tissue on a query set of genes of interest in another tissue. MultiCens enjoys theoretical guarantees on convergence and decomposability, and performs well on synthetic benchmarks. On human multi-tissue datasets, MultiCens predicts known and novel genes linked to hormones. Our findings on multilayer networks establish the necessary foundation for several other methods, such as clustering, where the effect of within-layer and across-layer edges must be separated.

In the next part of this thesis, we focus on another network structure, hypergraphs, and explore two problems, hypergraph clustering and hyperedge prediction. Hypergraph clustering is the problem of finding densely connected components (set of nodes) in a hypergraph. Hypergraph clustering is analogous to graph clustering, where modularity maximization-based clustering methods have been known to work well. In this work, we provide a generalization of the modularity maximization framework for clustering on hypergraphs. We also propose an iterative technique that provides refinement over the obtained clusters, as shown by our extensive set of experiments. The second problem in the space of hypergraphs we focus on is hyperedge prediction. This problem has immense applications in multiple domains, such as predicting new collaborations in social networks, discovering new chemical reactions in metabolic networks, etc. Despite its significant importance, the problem of hyperedge prediction hasn't received adequate attention, mainly because of its inherent complexity. We propose HPRA (Hyperedge Prediction using Resource Allocation) to predict new hyperedges with a reasonable computation time. The proposed method is tested to predict missing hyperedges as well as future hyperedges using past data, where it outperforms the state-of-the-art methods.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS **i**

ABSTRACT **iv**

LIST OF TABLES **xi**

LIST OF FIGURES **xiii**

ABBREVIATIONS **xiv**

1 Introduction **1**

1.1 Objectives and Scope 6

1.2 Contributions 9

1.3 Outline 11

I Fundamentals **12**

2 Preliminaries **13**

2.1 Basic Notations 13

2.1.1 Graph 13

2.1.2 Multilayer Networks 13

2.1.3 Hypergraphs 15

2.2 PageRank Centrality 17

2.3 Modularity 18

2.4 Link Prediction using Similarity-based Algorithms 19

2.5 Resource Allocation (RA) 20

2.6 Summary 20

3 Review of Multilayer Network Representations and Algorithms 21

3.1 Introduction 21

3.2 Inter-layer Coupling Methods 23

3.2.1 Diagonal Coupling (Adjacent Layers) 23

3.2.2 Diagonal Coupling-based Multilayer Networks 27

3.2.3 Cross Coupling-based Multilayer Networks 30

3.3 Summary 32

II Multilayer Networks: Centrality and its Applications 33

4 MultiCens: Multilayer Network Centrality framework 34

4.1 Introduction 35

4.2 Existing Methods 37

4.2.1 Degree-based Centrality and S_{sec} 37

4.2.2 Versatility 39

4.2.3 Other Works 40

4.3 Proposed Methods 40

4.3.1 Local Centrality 42

4.3.2 Global Centrality 43

4.3.3 Layer-specific Centrality 44

4.3.4 Query-set Centrality 45

4.3.5 Complexity analysis of the proposed centrality measures . . . 52

4.4 Summary 53

5 Applications of MultiCens framework to analyze Biological Networks 54

5.1 Introduction 54

5.2 Datasets and Network Generation 57

5.2.1 Synthetic Multilayer Networks 57

5.2.2 Real-world Multilayer Networks 58

5.2.3 Hormone-related Network Construction 59

5.2.4 Multi-brain-region Data - Preprocessing and Correction . . . 64

5.3 Results 66

5.3.1 Results with Synthetic Multilayer Networks 66

5.3.2 Results with Human Multi-tissue Data 68

5.3.3 Centrality of random node sets to assess statistical significance 75

5.3.4 Novel Findings: lncRNAs and their Support from Literature . 77

5.3.5 MultiCens detects Changes in Brain Networks between Alzheimer Disease and Control Populations 79

5.4 Summary 86

III Hypergraphs: Theoretical insights and implications for effective clustering and hyperedge prediction 87

6 Hypergraph: Methods for Clustering and Hyperedge Prediction 88

6.1 Hypergraph Modularity 89

6.1.1 Fundamental Observations: 94

6.1.2 Connection to Random Walks: 95

6.2 Iterative Hyperedge Reweighting 96

6.2.1 Hypergraph Resource Allocation (HRA) 101

6.2.2 Node-Hyperedge Attachment Score (NHAS) 103

6.2.3 HPRA: Hyperedge Prediction using Resource Allocation . . . 103

6.3 Summary 105

7 Applications: Hypergraph clustering and Hyperedge prediction 106

7.1 Hypergraph Clustering 106

7.1.1 Evaluation on Ground Truth 108

7.1.2 Settings for IRMM 110

7.1.3 Compared Methods 110

7.1.4 Datasets 111

7.1.5 Experiments 112

7.1.6 Results and Analysis 114

7.1.7 Complexity analysis of the proposed method 120

7.2 Hyperedge Prediction 123

7.3 Experiments 124

7.3.1 Baselines 124

7.3.2 Datasets 125

7.3.3 Evaluation of HPRA 128

7.3.4 HPRA with a Candidate Hyperedge Set (HPRA-CHS) 129

7.3.5 Future Hyperedge Prediction with HPRA 130

7.3.6 Results and Discussions 131

7.3.7 Computational complexity of HPRA 132

8 Conclusion 134

LIST OF TABLES

Table	Title	Page
3.1	Centrality measures for monoplex networks	22
5.1	Top predicted lncRNAs for Insulin in Pancreas	77
5.2	Top predicted lncRNAs for Insulin in Skeletal Muscle	77
5.3	Top predicted lncRNAs for Somatotropin in Pituitary Gland	78
5.4	Top predicted lncRNAs for Somatotropin in Liver	78
5.5	Top predicted lncRNAs for Progesterone in Uterus	78
5.6	Top predicted lncRNAs for Progesterone in Ovaries	78
5.7	Top predicted lncRNAs for Norepinephrine in Adrenal Glands	79
5.8	Top predicted lncRNAs for Norepinephrine in Small Intestine	79
7.1	Dataset Description	111
7.2	Rand Index, Purity and Average F1 scores against ground truth	113
7.3	Rand Index, Purity and Average F1 scores against ground truth	115
7.4	CPU times (in seconds) for the hypergraph clustering methods on all datasets	121
7.5	Datasets to evaluate the HPRA method	125
7.6	Average F1 Scores of HPRA and baselines	126
7.7	AUC results of HPRA and aforementioned baselines	127
7.8	Precision results of HPRA and aforementioned baselines	127
7.9	Temporal ACM Cocitation Dataset Description.	130
7.10	AUC, Precision and Average F1 score results	130

LIST OF FIGURES

1.1 Example multilayer network 2

1.2 An example hypergraph 3

2.1 A multilayer network depicting the cross inter-layer coupling 14

2.2 Supra-adjacency matrix representation of a multilayer network 15

2.3 Incidence matrix corresponding to the hypergarph 16

3.1 Supra adjacency matrix representation of multilayer network 24

3.2 A multilayer network depicting the diagonal inter-layer coupling of adjacent layers 25

3.3 Supra adjacency matrix representation of multilayer network 28

3.4 A multilayer network depicting the diagonal inter-layer coupling . . . 29

4.1 Workflow of our proposed MultiCens measures 42

5.1 Synthetic multilayer network construction and evaluation 67

5.2 MultiCens on human multilayer networks: ground-truth validation . . 69

5.3 OMIM-based disease set enrichment analysis 70

5.4 Literature support for the gene rankings obtained using the proposed centrality score 74

5.5 Comparison of gene rankings of ground truth sets with a random gene set 76

5.6 MultiCens on multi-brain-region networks in disease 81

5.7 Ranks and centrality scores of the genes participating in enriched biological process resulting from LC-GC delta rank 83

5.8 Ranks and centrality scores of the genes, participating in enriched pathways resulting from GC-QC delta rank 84

5.9 MultiCens on multi-brain-region networks in disease (PIG-based query-set) 85

6.1 Reweighting for different hyperedge cuts 98

6.2 Example illustrating resource transfer directly between nodes in a hypergraph. 101

6.3 An Illustration of HPRA algorithm 103

7.1 F1 scores for varying number of clusters 116

7.2 Effect of iterative hyperedge reweighting 118

7.3 CPU time (in secs) on synthetic hypergraphs 123

7.4 An illustration of the temporal dataset 130

7.5 AUC scores vs α 133

ABBREVIATIONS

ICN	Inter-organ Communication Network
WGCNA	Weighted correlation network analysis
GSEA	Gene Set Enrichment Analysis
FDR	False Discovery Rate
FP	Frontal Pole
STG	Superior Temporal Gyrus
CTL	samples in Control
AD	Alzhiemer’s Disease
GO-BP	Gene Ontology based Biological Process
IRMM	Iteratively Reweighted Modularity Maximization
HRA	Hypergraph Resource Allocation
NHAS	Node-Hyperedge Attachment Score

CHAPTER 1

Introduction

Many real-world systems can be characterized by several components that interact in multiple ways and determine the overall system behavior. Such systems are observed in various domains such as genes interacting in the tissues, railway junctions connected in transportation systems, multiple computers connected on the internet, humans involved in social relationships, researchers collaborating for projects, etc. Traditionally, graph-based modeling has been used to represent such systems. Though graph-based modeling is backed by well-established graph theory, it fails to retain the peculiar artifacts of the underlying complex systems. For instance, a set of genes in the human body can have different interaction patterns in different tissues, resulting in a multilayer structure where a dedicated layer represents each tissue. A graph-based model will either overlook the peculiarities of individual tissues and merge all connections in the same layer or ignore the multilayer tissue structure of the system and model a giant network of interacting genes. A multilayer network where each layer represents a tissue provides a natural representation of the multi-tissue system. Here, each layer (tissue) comprises the same set of nodes (genes), but their connectivity (edge) pattern in each layer may differ.

Similarly, in bibliographic systems, multiple researchers can collaborate on the same project resulting in a super-dyadic relation (involving more than two agents). Modeling it as a graph will limit us to capture only pairwise relations and breaking the super-dyadic relations into pairwise interactions. Hypergraphs provide a natural representation for such systems where the researchers are represented by nodes and hyperedges connect nodes that form collaborations. In this thesis, we explore multilayer networks and hypergraphs. In addition to multilayer networks and hypergraphs, there are several other complex network structures, such as knowledge graphs, heterogeneous graphs, multipartite graphs, etc., which are not covered in this thesis.

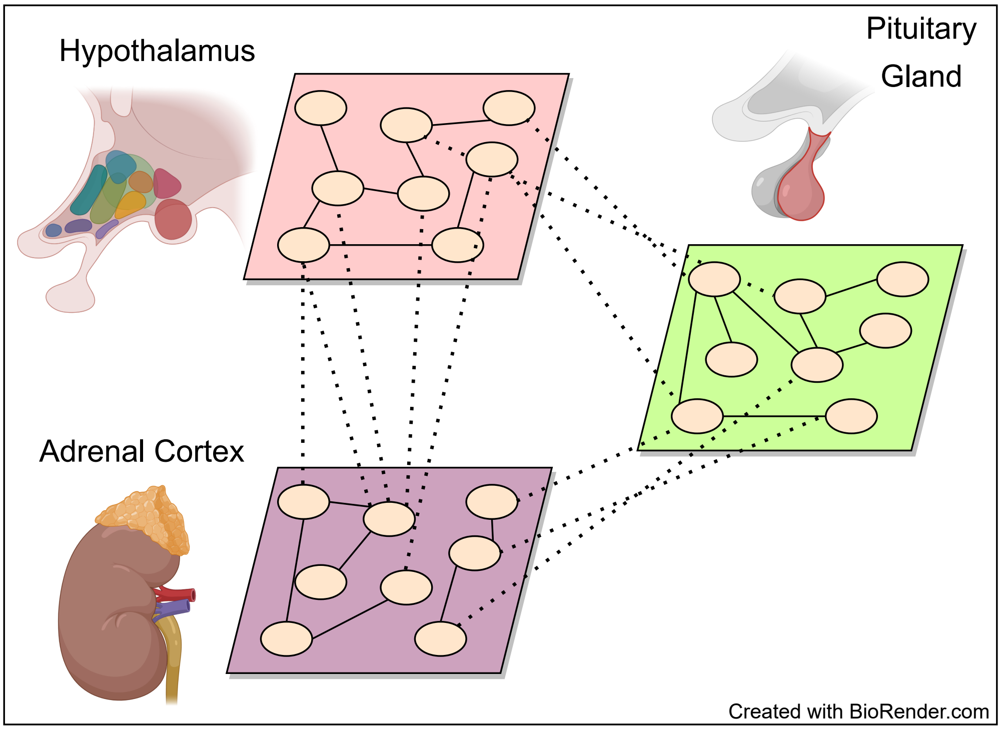


Figure 1.1: **Example multilayer network**: A multi-tissue system comprising of Hypothalamus, Pituitary Gland and Adrenal Cortex, also known as HPA axis. Each tissue corresponds to a layer in the multilayer network and each layer contains the same set of genes which are connected based on their co-expression scores. Solid edges represent within-tissue connections and inter-tissue connections are represented by dotted edges.

Figure 1.1 shows a multilayer network representation of a multi-tissue system. There are three tissues encoded by three layers, where each layer comprises the same set of genes encoded by nodes. Unlike graphs, multilayer networks can distinguish between within-layer and cross-layer connections, which makes this representation suitable to

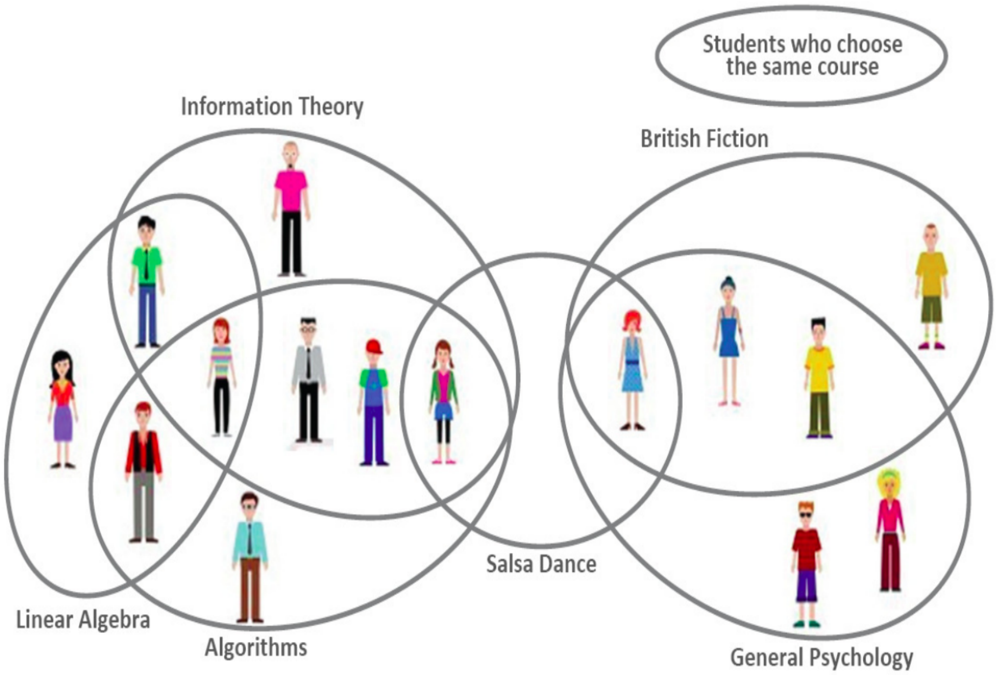


Figure 1.2: **An example hypergraph**: An example hypergraph representing a collaboration network. Here, students represent nodes in the hypergraph and a set of nodes are connected if they are part of the same study group.

answer several interesting questions, as we will discuss in the remaining part of this section. Figure 1.2[1] shows an example collaboration hypergraph where students are represented by nodes and hyperedges connect students from the same study group. Unlike graphs, a hypergraph can capture the underlying system's super-dyadic (of cardinality greater than two) relations.

In this work, we start with investigating the existing network science methods and measures for traditional graphs and ways to extend them to multilayer networks and hypergraphs.We begin by reviewing the different coupling techniques of multilayer networks and dive into the most generalized inter-layer cross-coupling scheme. We focus on finding node centrality (also called network centrality) in multilayer networks. Network centrality assesses the “importance or centrality" of each node in a network

---

[1] Image source: http://www.math.iisc.ernet.in/~ifcam/new_avenue/Slides/Seminars/Ravindran.pdf

by quantifying how well-connected or well-knit each node is to other nodes in the network. Network centrality has numerous applications, including finding influential people in a social network (Awangga *et al.* (2018)), detecting essential proteins or disease genes in biological networks (del Rio *et al.* (2009); Mistry *et al.* (2017)), etc. Our goal is to discover the genes responsible for multi-tissue communication. Traditionally, centrality algorithms have been used to study gene-gene interactions, but most of them are limited to single-layer graphs (Jeong *et al.* (2001)) and hence are suitable for single tissue systems. Recently, the community has seen progress in analyzing multi-tissue interaction data, but these studies fail to distinguish between inter and intra-tissue communication links. A central open question in extending centrality measures to multi-tissue networks is the assessment of the local effect of a gene in its layer vs. the global effects of the gene in the overall network or in specific target tissues or target gene sets. In this study, we propose several PageRank-like iterative centrality measures that offer these local vs. global effects of genes by decomposing the overall multilayer centrality into relative contributions from intra-tissue vs. inter-tissue edges. We show that these measures have desirable theoretical properties like decomposability and derive the necessary conditions for convergence.

In the later part of the thesis, we focus on hypergraph clustering and hyperedge prediction. Learning on hypergraphs has been garnering increased attention with potential applications in network analysis, VLSI design, and computer vision, among others. Hypergraph clustering is gaining attention because of its enormous applications such as component placement in VLSI (Karypis *et al.* (1999); Shamir (2008)), group discovery in bibliographic systems (Sharma *et al.* (2014)), image segmentation in computer vision (Ducournau and Bretto (2014)), etc. For the problem of clustering on graphs, modularity maximization has been known to work well in the pairwise setting. Our primary contribution in this work is to provide a generalization of the modularity maximization framework for clustering on hypergraphs. In doing so, we introduce a null model for graphs generated by hypergraph reduction and prove its equivalence to the configuration model for undirected graphs. The proposed graph reduction technique preserves the node degree sequence from the original hypergraph. The modularity function can be defined on a thus reduced graph, which can be maximized using any standard modularity maximization method, such as the Louvain method. We additionally propose an iterative technique that provides refinement over the obtained clusters. We demonstrate both the efficacy and efficiency of our methods on several real-world datasets.

In hyperedge prediction, our goal is to predict either missing or future hyperedges in a given hypergraph. This problem has immense applications in multiple domains, such as predicting new collaborations in social networks (Yoon *et al.* (2020)), discovering new chemical reactions in metabolic networks (Zhang *et al.* (2018a)), etc. Despite its significant importance, hyperedge prediction has not received adequate attention, mainly because of its inherent complexity. In a graph with n nodes, the number of potential edges is $\mathcal{O}(n^2)$, whereas in a hypergraph, the number of potential hyperedges is $\mathcal{O}(2^n)$ and there is a need for non-trivial ways to explore it. Existing hyperedge prediction methods restrict this search space by either fixing the hyperedge degree to a specific number k (*k-uniform hypergraph*) or by exploring only a set of potential hyperedges (known as candidate hyperedge set). As many real-world hypergraphs are not restricted to be k-uniform, and it is not always feasible to access the collection of potential hyperedges, existing methods fail to predict hyperedges. We propose *HPRA - Hyperedge Prediction using Resource Allocation*, the first-of-its-kind algorithm, which can predict hyperedges of any cardinality without using any candidate hyperedge set in a reasonable time where existing algorithms fail. *HPRA* is a similarity-based method working on the principles of the resource allocation process. In addition to recovering missing hyperedges, we demonstrate that *HPRA* can predict future hyperedges in a wide range of hypergraphs. Our extensive set of experiments shows that HPRA achieves statistically significant improvements over state-of-the-art methods.

1.1 OBJECTIVES AND SCOPE

In this thesis, we work with two rich graph structures - multilayer networks and hypergraphs with the following key objectives.

Centrality in Multilayer Networks

This problem focuses on finding node importance in multilayer networks. In particular, we are interested in defining node centrality in the context of a target layer or target set of nodes in a target layer. Such a measure has immediate applications in biological systems. Knowing the genes with target effect on another set of genes can reveal interesting insights such as gene-hormone relations, potential drug targets, dispersion of abnormalities from one brain region to another, etc. The existing multilayer network centrality measures either rely entirely on the inter-layer edges or treat the inter-layer and intra-layer edges equally. These methods fail to capture long-hop effects targeted to a specific layer or a specific set of genes. In this work, our objective is to define a set of centrality measures that can distinguish between the local effect of a node in its layer vs. the global effects of the node in the overall network or in a specific target layer or target node-set. We also want our centrality measure(s) to have theoretical guarantees such as convergence, decomposability, etc. We apply our proposed centrality measures to predict genes involved in inter-tissue communication in multi-tissue systems. Our centrality measures can readily be applied to answer several additional exciting questions in systems biology and can potentially be applied to other settings, such as studying multimodal transportation systems' congestion behavior.

Hypergraph clustering

This problem can be formally stated as finding densely connected components in a network. One definition of a densely connected component is if a set of nodes have more than the expected number of edges among them. The modularity of a network is a measure that denotes the difference between the observed number of edges and the expected number of edges for a given clustering assignment (Newman (2006)). Maximizing modularity over a given network to find the optimal clustering is a hard problem (Brandes *et al.* (2006)). Louvain algorithm is a heuristic-based modularity maximization approach that is known to work well on networks with pair-wise interactions. In order to apply the Louvain algorithm to hypergraphs, we need to define modularity for hypergraphs. In this work, our objective is to define modularity for hypergraphs, propose a reduction mechanism that projects a hypergraph to a weighted graph and prove that maximizing modularity over this reduced graph and the original hypergraph are equivalent. We also aim to analyze the way hyperedges get cut during clustering and intend to obtain the cluster assignments where hyperedges get cut in a balanced way.

Hyperedge Prediction

This problem focuses on predicting hyperedges in a given hypergraph. In a graph with n nodes, the number of potential edges is $\mathcal{O}(n^2)$, whereas in a hypergraph, the number of potential hyperedges is $\mathcal{O}(2^n)$ and there is a need for non-trivial ways to explore it in an efficient manner. In order to avoid searching through this vast space of hyperedges, current methods restrain the original problem in the following two ways. One class of algorithms assumes the hypergraphs to be k-uniform, where each hyperedge can have exactly k nodes. However, many real-world systems are not confined to have interactions involving only k components. Thus, these algorithms are not suitable for many real-world applications. The second class of algorithms requires a candidate set of hyperedges from which the potential hyperedges are chosen. In the absence of domain knowledge, the candidate set can have $\mathcal{O}(2^n)$ possible hyperedges, which makes this problem intractable. More often than not, domain knowledge is not readily available, making these methods limited in their applicability. In this work, our objective is to propose an algorithm to predict hyperedges of any cardinality without relying on a candidate hyperedge set. The problem of hyperedge prediction can arise either because of missing hyperedges or to predict potential future interactions. Our objective is to test the proposed method under both settings. The problem of hyperedge prediction can arise either because of missing hyperedges in past data or to predict future interactions.

1.2 CONTRIBUTIONS

In this thesis, we work with two rich graph structures - multilayer networks and hypergraphs and focus on the following research problems.

Multilayer Network Centrality

We model multi-tissue systems as multilayer networks and propose a set of centrality measures that can capture effects of genes within a tissue, across the tissue, targeted to a specific tissue or set of genes in a particular tissue. Our work is among very few approaches to model multi-tissue system as a multilayer network to study inter-tissue communication. To systematically identify inter-tissue mediators, we present a novel computational approach MultiCens (Multilayer/Multi-tissue network Centrality measures). Unlike single-layer network methods, MultiCens can distinguish within- vs. across-layer connectivity to quantify the "influence" of any gene in a tissue on a query set of genes of interest in another tissue. MultiCens enjoys theoretical guarantees on convergence and decomposability, and performs well on synthetic benchmarks. On human multi-tissue datasets, MultiCens predicts known and novel genes linked to hormones. MultiCens further reveals shifts in gene network architecture among four brain regions in Alzheimer's disease. MultiCens-prioritized hypotheses from these two diverse applications, and potential future ones like "Multi-tissue-expanded Gene Ontology" analysis, can enable whole-body yet molecular-level systems investigations in humans.

Hypergraph Clustering

We define a null model for graphs generated by hypergraph reduction that preserves the hypergraph node degree sequence. Using this null model and the hypergraph reduction mechanism, we define a modularity function that can be used in conjunction with the popular Louvain method to find clusters in a hypergraph. We propose a generic iterative refinement procedure to enforce balanced hyperedge-cuts and eventually improve the cluster quality in hypergraphs. This refinement is done by reweighting hyperedges and operating natively on the hypergraph structure. We perform extensive experiments with the proposed algorithm, titled Iteratively Reweighted Modularity Maximization (IRMM), on a wide range of real-world datasets and demonstrate both its efficacy and efficiency over state-of-the-art methods. We empirically establish that the hypergraph-based methods perform better than their graph-based counterparts. We also examine the scalability of the hypergraph modularity maximization algorithm using synthetically generated hypergraphs.

Hyperedge Prediction

We propose a computationally efficient hyperedge prediction model, HPRA (Hyperedge Prediction using Resource Allocation), which can predict novel hyperedges without using any candidate hyperedge set. Since many of the existing works use a candidate set while predicting hyperedges, we propose a variant of HPRA that can work under this setting as well. We show that HPRA can recover missing hyperedges in a given hypergraph as well as predict future hyperedges in a temporal hypergraph. Our comprehensive set of experiments demonstrates that HPRA significantly outperforms the state-of-the-art methods in terms of widely used metrics such as the area under the precision-recall curve (AUC), F1 score and precision.

1.3 OUTLINE

The current chapter provides the introduction to the complex networks area along with the problems that we study. This thesis is comprised of three parts. The first part provides fundamental concepts required to define the problems. This part begins with Chapter 2, where we introduce the notations used in this thesis and discuss the basic concepts of modularity maximization, link prediction, resource allocation, and node centrality. In Chapter 3, we will dive into multilayer networks and discuss different coupling schemes with a particular focus on cross-coupled multilayer networks. In the second part, we begin with defining the centrality measures and discussing their theoretical aspects in Chapter 4 and continue with their applications in biological systems in Chapter 5. The third part is dedicated to hypergraphs, and we discuss the proposed frameworks for hypergraph clustering and hyperedge prediction in Chapter 6. In Chapter 7, we present the empirical validation of the proposed methods.

Part I

Fundamentals

CHAPTER 2

Preliminaries

This section defines the basic notations and fundamental concepts required to formulate our research problem and derive its solution.

2.1 BASIC NOTATIONS

In this section, we define the basic notations used in the thesis to represent multilayer networks and hypergraphs. We begin by providing definition of a graph.

2.1.1 Graph

A graph $G = (V, E)$, where V is the set of n nodes (or vertices) and E is the set of m edges is defined by its adjacency matrix as:

$$\mathbf{A}(i,j) = \begin{cases} w_{ij} & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where w_{ij} denotes the weight of an edge between nodes v_i and v_j by a non-zero value.

2.1.2 Multilayer Networks

A multilayer network is represented by $G = (V, \mathbb{L}, E)$, where V represent the set of n nodes which is the same across all layers, $\mathbb{L}$ is the set of L number of layers and E represents the set of inter and intra layer edges. The set of nodes in layer α is represented by $V = \{v_1^\alpha, v_2^\alpha, \ldots, v_n^\alpha\}$. The total number of nodes in the multilayer network is

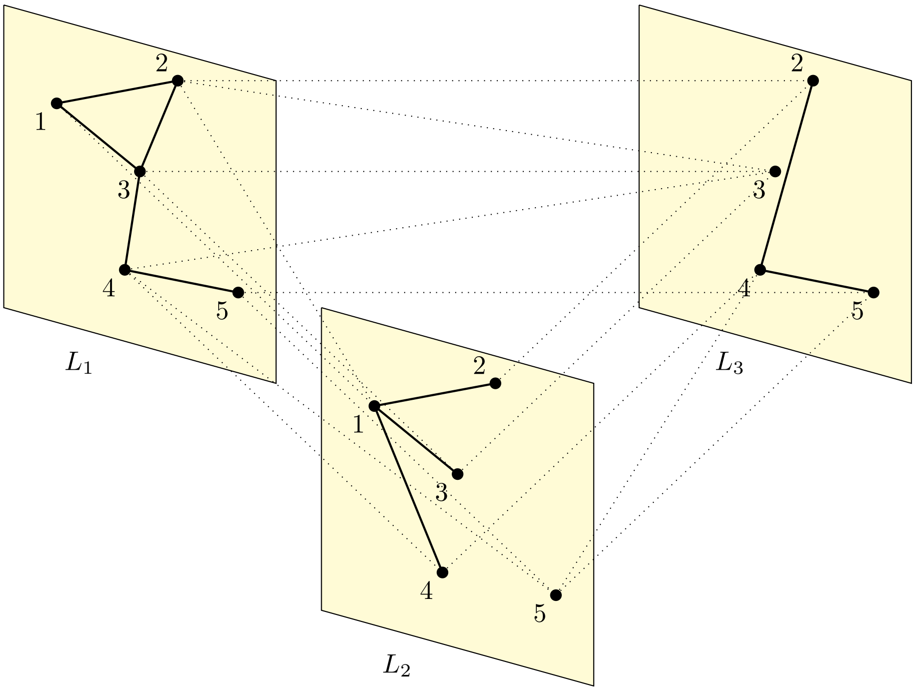


Figure 2.1: A multilayer network depicting the cross inter-layer coupling. A node is allowed to have connections with nodes within the same layer and nodes from other layers. The node-set in the layers need not be the same among all layers, as node 1 is missing in L_3. In order to keep the notation consistent, such nodes are added without any connections.

$N = n \times L$. Following the convention used in (Gomez *et al.* (2013)), we represent the multilayer network by a supra-adjacency matrix M of dimension $N \times N$ as,

$$\mathbf{M}(i_\alpha, j_\beta) = \begin{cases} w(i_\alpha, j_\beta) & \text{if } (v_i^\alpha, v_j^\beta) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

where $w(i_\alpha, j_\beta)$ denotes the weight of edge between node i in layer α and node j in layer β.

Supra-adjacency matrix is a special kind of matrix with an inherent block structure with diagonal blocks dedicated to intra-layer edges and non-diagonal blocks dedicated to inter-layer edges. For the multilayer network shown in Figure 2.1, the supra-adjacency matrix is shown in Figure 2.2.

In Figure 2.2, the presence of intra-layer edges is indicated by 🟦 and inter-layer edges are indicated by 🟥. Since all the edges (inter-layer and intra-layer) are undirected for this example, so the supra-adjacency matrix is symmetric.

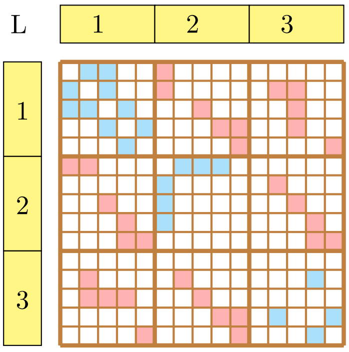


Figure 2.2: Supra-adjacency matrix representation of the multilayer network shown in Fig. 2.1.

2.1.3 Hypergraphs

A hypergraph is represented by a tuple $G = (V, E)$; where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes (or vertices) and $E = e_1, e_2, \ldots, e_m$ is the set of m hyperedges. Each hyperedge e contains an unordered subset of V and has a positive weight $w(e)$ associated with it. While in a traditional graph, an edge connects two nodes, a hyperedge can connect an arbitrary number of nodes. Degree of node v is defined as $d(v) = \sum_{e \in E, v \in e} w(e)$ and $N(v)$ is a set containing the one-hop neighbors of node v (nodes of hyperedges, v is part of). For a hyperedge e, its degree is defined as $\delta(e) = |e|$. Incidence matrix H of a hypergraph is defined as

$$\mathbf{H}(v,e) = \begin{cases} 1 & \text{if } v \in e \\ \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

Presence of 1 in the incidence matrix represents participation of the corresponding node in that particular hyperedge. $D_v \in \mathbb{R}^{n\times n}$, $D_e \in \mathbb{R}^{m\times m}$ and $W \in \mathbb{R}^{m\times m}$ are the diagonal matrices containing node degrees, hyperedge degrees and hyperedge weights at the diagonals and zero otherwise, respectively. Figure 2.3 shows an example hypergraph and an incidence matrix associated with it. Each hyperedge corresponds to one column, and a row represents a vertex in the incidence matrix.

	e1	**e2**	**e3**	**e4**	**e5**
v1	1	0	0	0	0
v2	1	0	0	0	0
v3	1	1	0	0	0
v4	0	1	1	0	0
v5	0	0	1	0	0
v6	0	0	1	1	0
v7	0	0	0	1	1
v8	0	0	0	0	1
v9	0	0	0	0	1
v10	0	0	0	0	1

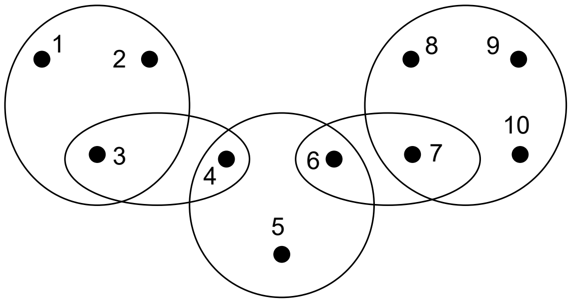


Figure 2.3: An incidence matrix (left) corresponding to the hypergarph (right).

The adjacency matrix of the graph associated to a hypergraph G is defined as(Hadley *et al.* (1992)): $A = HWH^T - D_v$.

2.2 PAGERANK CENTRALITY

PageRank, initially proposed for a network of webpages (Page *et al.* (1999)), assigns an importance score to each node in the network. The PageRank algorithm is based on the principle of aggregating node importance from its neighbors; hence a node reachable by several other central nodes gets high centrality. Another interpretation of PageRank centrality is based on random web surfer model, which works as follows. On a given web, a random walk starts from any webpage, and at each step it does one of the following,

1. It jumps to a page linked by a hyperlink with probability p.
2. With probability $(1-p)$, it jumps to a random webpage in the network and restarts the walk.

In such a model, the PageRank centrality of a node/webpage is given by the fraction of times a webpage is visited after infinite random walks (Chung and Zhao (2010)). Formally, for a network G with adjacency matrix A, the PageRank centrality x is given by the following expression.

$$x = pAx + \frac{(1-p)}{n}\vec{1}$$

where p is a scalar known as damping factor that denotes the transition probability of the random walker to a neighborhood node. $\vec{1}$ is known as the personalization vector that contains all ones.

2.3 MODULARITY

When clustering graphs, it is desirable to cut as few edges (or edges with lesser weights in the case of weighted graphs) within a cluster as possible. Modularity is a metric of clustering quality that measures whether the number of within-cluster edges is greater than its expected value. In (Newman (2006)) the modularity function is defined as:

$$\begin{aligned} Q &= \frac{1}{2m}\sum_{ij}[A_{ij} - P_{ij}]\delta(g_i, g_j) \\ &= \frac{1}{2m}\sum_{ij} B_{ij}\delta(g_i, g_j) \end{aligned} \tag{2.4}$$

Here, $\delta(.)$ is the Kronecker delta function, and g_i, g_j are the clusters to which vertices i and j belong. The $\frac{1}{2m}$ is based on a constant (number of edges) for a given graph and can be dropped as it does not affect the maximization of Q. $B_{ij} = A_{ij} - P_{ij}$ is called the modularity matrix. A_{ij} denotes the actual, and P_{ij} denotes the expected number of edges between node i and node j, given by a *null model*. For graphs, the *configuration model* (Newman (2010)) is used as a *null model*, where edges are drawn at random while keeping the node-degree preserved. For two nodes i and j, with (weighted) degrees k_i and k_j respectively, the expected number of edges between them is hence given by:

$$P_{ij} = \frac{k_i k_j}{\sum_{j \in V} k_j}$$

Since the total number of edges in a given network is fixed, maximizing the number of within-cluster edges is the same as minimizing the number of between-cluster edges.

This suggests that clustering can be achieved by *modularity maximization*. Kindly note that in this thesis, we focussed on modularity as defined by Newman et al. (Newman (2006)). There are other definitions of modularity (Courtney and Bianconi (2016)) that we do not consider in this work.

2.4 LINK PREDICTION USING SIMILARITY-BASED ALGORITHMS

Several kinds of real-world networks such as social networks, web networks are known to exhibit the property of *homophily*, which states that similar nodes are more likely to connect in the future than dissimilar nodes (McPherson *et al.* (2001); Sarkar *et al.* (2011)). In accordance with this, similarity-based algorithms are broadly used for edge prediction in graphs. In a typical similarity-based algorithm, a similarity score is defined for node pairs of a graph. All the possible edges are ranked based on the similarity score, and the top-ranked edges are chosen as the potential edges. One group of existing works use the node attributes to define the similarity score for node-pairs (Lin (1998)), but are restricted to attributed graphs. Without such restriction, another group of methods defines similarity scores solely based on the network structure and are termed structural similarity scores. Popular structural similarity scores are *common neighbors* (Newman (2001)), *resource allocation* (Zhou *et al.* (2009)) and *katz index* (Katz (1953)). Among these similarity scores, *resource allocation* is shown to work well in a wide variety of graphs Lü and Zhou (2011). We elaborate on *resource allocation* similarity score in the remaining part of this section.

2.5 RESOURCE ALLOCATION (RA)

Motivated by the resource allocation process in networks (Georgiadis *et al.* (2006)), *RA* score (Zhou *et al.* (2009)) for a node-pair (x, y), which are *not directly connected* is defined as:

$$RA_{xy} = \sum_{z \in N(x) \cap N(y)} \frac{1}{d(z)}$$

To illustrate resource allocation in a simple way, assume node x has a resource amount of $d(x)$ units allocated to it. Node x transfers its resource to node y through common neighbors, who act as transmitters in the following way; Node x uniformly distributes its resource to all its neighbors, resulting in each neighbor of x getting a unit of resource. Following this, neighbors of x uniformly transfer their unit resources to their neighbors. The resource that node y receives from node x is defined as the resource allocation score for pair (x, y). A higher amount of resource transferred between two nodes signifies a higher similarity among those nodes.

2.6 SUMMARY

This chapter introduced the mathematical framework required to formulate our research problem and derive its solution. We presented mathematical notations to describe graphs, multilayer networks and hypergraphs and skimmed through some methods and processes defined for graphs. In the upcoming chapters, we will extend some of these approaches to multilayer networks and hypergraphs. In the next chapter, we begin by plunging into the anatomy of multilayer networks.

CHAPTER 3

Review of Multilayer Network Representations and Algorithms

3.1 INTRODUCTION

In this chapter, we explore multilayer networks in the context of different coupling schemes and understand the applicability of various centrality measures to these networks. Centrality can be defined in multiple ways depending on the type of network (directed/undirected, size) or application domain. For example, PageRank is an appropriate centrality measure for ranking the web pages in response to a search query (Gleich (2015)), whereas betweenness centrality is employed in designing the packet routing strategies in computer networks (Dolev *et al.* (2010); Holme (2003)). A thorough study of the centrality measures for monoplex (i.e., single layer) networks can be found at (Newman (2018)). Let $A \in \mathbb{R}^{N \times N}$ be the adjacency matrix of a monoplex network with N nodes, where A_{ij} is the strength of the ij^{th} connection. Let $x \in R^N$ be the centrality vector of a network where x_i represents the centrality of node i. For a given adjacency matrix A, Table 3.1 represents different ways of calculating centrality vector x. These centrality measures are traditionally defined for single-layered graphs. With the advent of data collection methods and superior data processing techniques, we often have access to multiple views of the data. For example, a set of authors can have multiple types of relationships, such as co-authorship, citation, co-citation, etc. Networks that change their structure with time (temporal networks) can also give rise to multiple data views. Such systems are better modeled by multilayer networks. Extending existing centrality measures to multilayer networks is a non-trivial task.

Centrality measure	Centrality of node i	Description
Eigenvector centrality	$x_i = \lambda^{-1} \sum_j A_{ij} x_j$	λ is the leading eigenvalue
PageRank centrality	$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j}$	α is a scalar and k_j is the degree of node j
HITS centrality (authority)	$x_i = \alpha \sum_j A_{ij} y_j$	α is a scalar and y_j is the hub centrality of node j
HITS centrality (hub)	$y_i = \alpha \sum_j A_{ij} x_j$	α is a scalar and x_j is the authority centrality of node j
Betweenness centrality	$x_i = \sum_{st} \frac{n_{st}^i}{g_{st}}$	n_{st}^i is the # of shortest paths between s and t that go through i. g_{st} is the total # of shortest paths between nodes s and t

Table 3.1: Centrality measures for monoplex networks. For detailed description of these measures and many others, please refer Newman (2018)

Finding centrality in multilayer networks has immense applications, such as the study of the emergence of congestion in transport flows (Solé-Ribalta *et al.* (2016*b*)), ranking in evolving networks (Liao *et al.* (2017)), and analyzing different life stages in the species (Shinde and Jalan (2015)). In multilayer networks, the local neighborhood of a node can comprise nodes from the same layer as well as nodes from other layers. To define a centrality measure for multilayer networks, one has to come up with a way to handle the multilayer neighborhood of a node. Recently there have been several attempts at defining centrality measures for multilayer networks. Most of these methods differ in the way they handle inter-layer coupling. For instance, the multiple layers can be merged to form a monoplex network, or at the other extreme, the multilayer network itself can be treated as a giant monoplex network. The coupling methods have their own implications when combined with the centrality measures. In the upcoming sections, we will discuss how existing works extend centrality measures to multilayer networks.

3.2 INTER-LAYER COUPLING METHODS

There are different possible approaches to handle multiple layers and inter-layer edges of multilayer networks. The first approach is to ignore the layered structure of the network and treat nodes from all the layers like a giant network. One can use existing centrality measures on this entire network. This approach fails to distinguish between intra-layer and inter-layer edges and is hence not appropriate for analyzing multilayer networks. Another approach is to calculate the centrality of node i in each layer separately and get the vector $c(i) = \{x^{[1]}(i), \dots x^{[L]}(i)\}$. Node centrality x_i can be identified by finding the mean of $c(i)$. In addition to finding the mean, there are several other possible ways such as finding a convex combination, finding the weighted average (Battiston *et al.* (2017)), or normalizing the eigenvector relative to the largest eigenvalue (Solá *et al.* (2013)), etc. This approach seems to be straightforward; however, it ignores the inter-layer coupling of the network. Setting aside these two naive methods, we discuss the following coupling schemes in the remaining part of this chapter.

3.2.1 Diagonal Coupling (Adjacent Layers)

Diagonal coupling refers to the condition where interlayer edges are only allowed between identical nodes in a pair of layers. For the purpose of this section, we will assume that the layers are ordered, and interlayer edges are only permitted between two adjacent layers. This type of network is particularly useful for modeling temporal systems, where changes in the underlying system at different time points lead to alterations in the network structure. As a result, multiple layers of the network are generated, with each layer corresponding to a specific timestamp. Finding centrality in temporal networks has many applications such as dynamic network analysis (Liu *et al.* (2018)), finding temporal node centrality (Kim and Anderson (2012)), finding joint and marginal centrality (Yin *et al.* (2018)), etc. Network structure at different timestamps can be interpreted as multiple layers of a larger network (Gallotti and Barthelemy (2015); Hristova *et al.* (2016)). Working with the layers independently to define temporal network centrality measures may lead to undesired results like sudden fluctuations in the centrality scores as seen in unsteady university rankings in a multilayer academic network (Sorz *et al.* (2015)). Temporal networks have the special property of having the inter-layer coupling only between adjacent layers. Which means that the network can have inter-layer edges only between layers $\{\alpha, \alpha \pm 1\}$. This

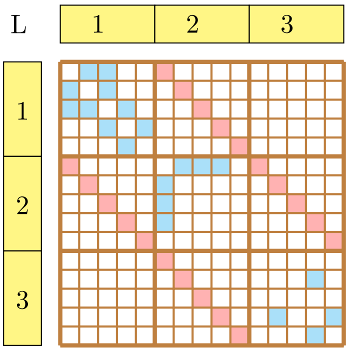


Figure 3.1: Supra adjacency matrix representation of multilayer network shown in Fig. 3.2

particular kind of coupling leads to diagonal blocks in non-diagonal positions (adjacent to diagonal blocks) of the supra-adjacency matrix as shown in Fig. 3.1.

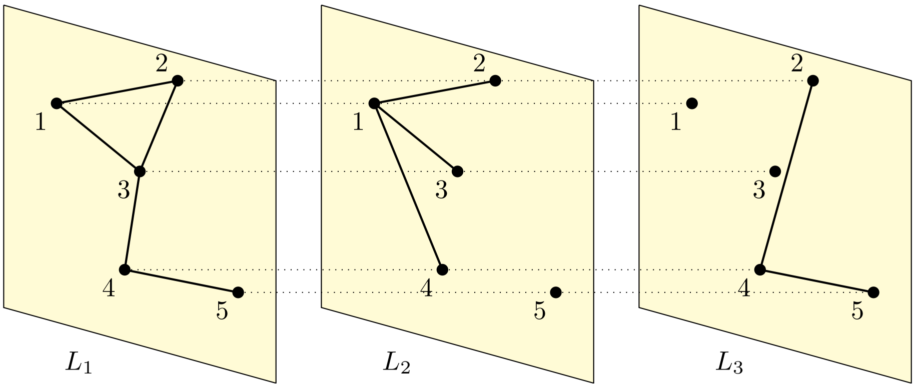


Figure 3.2: A multilayer network depicting the diagonal inter-layer coupling of adjacent layers, for example, time-series points.

To extend the existing centrality measures to temporal networks, one obvious way is to use the $N \times N$ supra-adjacency matrix with different attention to the inter-layer edges. Here, the magnitude of the attention can be represented by assigning a corresponding weight to the inter-layer connections. There can be several criteria to identify appropriate weight for inter-layer edges. In general, the supra-adjacency matrix can be written as,

$$A = \begin{bmatrix} A^{[1]} & \omega(I) & 0 & \dots \\ \omega(I) & A^{[2]} & \omega(I) & \ddots \\ 0 & \omega(I) & A^{[3]} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

where $\omega \geq 0$ is known as the layer coupling coefficient. Traditional centrality measures can directly be applied on A, which will lead to a centrality vector of size N. The centrality vector can be interpreted to return the node centrality at each time stamp

(Bazzi *et al.* (2016)). Clearly, this approach ignores the block diagonal structure of the matrix and the centrality measure doesn't distinguish between the inter-layer and intra-layer edges (De Domenico *et al.* (2013*b*)). This issue can be circumvented by changing the representation of either the inter-layer edges or the intra-layer edges. We discuss both of these methods below:

Inter-layer Coupling of Centrality Matrices

The idea of this approach is to find the centrality matrix for each layer and directly couple it to the centrality matrix of its adjacent temporal layers. Let $C^{[\alpha]}$ denotes the centrality matrix for the temporal network at layer α. Let $\epsilon = \frac{1}{\omega}$. Then the *supra-centrality* matrix (Taylor *et al.* (2017) can be represented as,

$$\mathbb{C} = \begin{bmatrix} \epsilon C^{[1]} & I & 0 & \dots \\ I & \epsilon C^{[2]} & I & \ddots \\ 0 & I & \epsilon C^{[3]} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

The above formulation works with the assumption that $C^{[\alpha]}$ is non-negative and irreducible for every $\alpha \in L$. Similarly, $\mathbb{C}$ is also non-negative and irreducible for any $\epsilon > 0$, which leads to the Perron– Frobenius theorem for non-negative matrices (Meyer (2000) and ensures uniqueness of the largest eigenvalue and non-negativity of the corresponding eigenvector (Yin *et al.* (2018)). Thus, the $\mathbb{C}$ matrix can be viewed as an adjacency matrix to find centrality using standard methods.

Incorporating Inter-layer Similarity

Let $C^{[\alpha,\alpha+1]} = diag(c_1^{[\alpha,\alpha+1]}, c_2^{[\alpha,\alpha+1]}, \ldots, c_N^{[\alpha,\alpha+1]})$ be the $N \times N$ dimensional inter-layer similarity matrix for layers α and $\alpha + 1$. The supra-adjacency matrix can then be formulated (Yin *et al.* (2018) as:

$$\mathbb{A} = \begin{bmatrix} A^{[1]} & C^{[1,2]} & 0 & \cdots \\ C^{[2,1]} & A^{[2]} & C^{[2,3]} & \ddots \\ 0 & C^{[2,3]} & A^{[3]} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

where $c_i^{\alpha,\alpha+1}$ denotes the similarity between the same physical node at two adjacent layers. The supra-adjacency matrix $\mathbb{A}$ can now be used to find the centrality. There are many similarity measures to compute $c_i^{\alpha,\alpha+1}$ such as Adamic-Adar Index (Adamic and Adar (2003)), Jaccard Index, Salton Index (Hamers *et al.* (1989)), Resource Allocation Index (Zhou *et al.* (2009)), etc. For a detailed experimental study on this method, kindly refer to (Yin *et al.* (2018)).

3.2.2 Diagonal Coupling-based Multilayer Networks

In time-independent multilayer networks, the inter-layer coupling is not limited to adjacent layers. For example, a multilayer network can represent different relationships among authors such as citation, co-authorship, co-citation, etc. In such networks, one can observe inter-layer coupling among all pairs of layers as shown in Figure 3.3. Without loss of generality, one can assume the same set of vertices but a possibly different set of edges in different layers. Finding centrality in such networks has multiple applications such as node ranking (Sideris *et al.* (2018)), finding the most versatile nodes (De Domenico *et al.* (2015)), etc. For the same reason, multilayer centrality is also referred to as versatility.

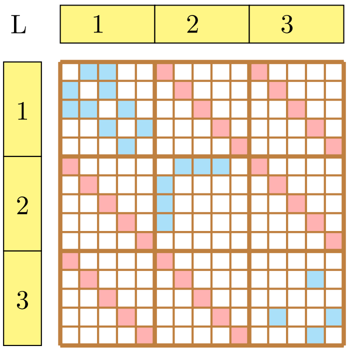


Figure 3.3: Supra adjacency matrix representation of multilayer network as shown in Fig. 3.4

Network structure in layer α may get influenced by the nodes from other layers α'. The centrality measure for multilayer networks must take this influence into account. This influence among layers can be captured by a matrix $W \in \mathbb{R}^{L\times L}$, where $w_{\alpha\beta}$ denotes the influence of layer α on layer β. Once the $w_{\alpha\beta}$ is fixed, we can define the local multilayer eigenvector-like centrality c_α as a leading eigenvector of the following matrix:

$$A_\alpha = \sum_\beta w_{\alpha\beta} A^{[\beta,\beta]}$$

Now, we can directly use this matrix to find centrality with traditional methods.

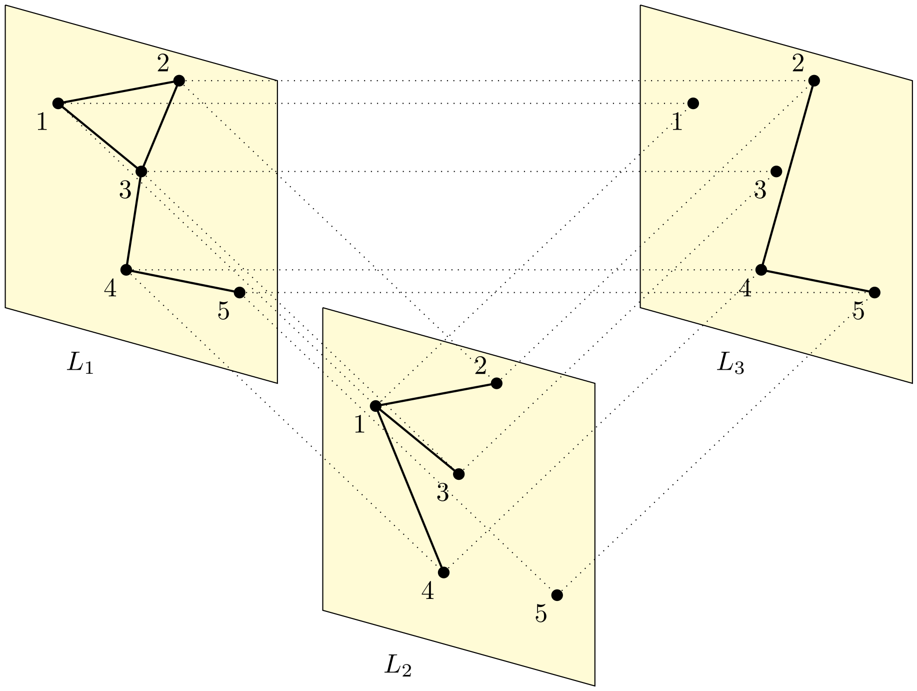


Figure 3.4: A multilayer network depicting the diagonal inter-layer coupling. Every node in a layer is connected to its counterpart in all other layers.

In some cases, the centrality of a node v in a layer not only depends on the other connected nodes in the same layer α but also on the nodes from other layers. In such networks, it becomes essential to consider the influence of the nodes across the layers. To find the centrality of a node in a particular layer, the following modified adjacency matrix can be used (Solá *et al.* (2013)).

$$A^{\otimes} = \begin{bmatrix} w_{11}A^{[1,1]} & w_{12}A^{[2,2]} & \dots & w_{1L}A^{[L,L]} \\ w_{21}A^{[1,1]} & w_{22}A^{[2,2]} & \dots & w_{2L}A^{[L,L]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{L1}A^{[1,1]} & w_{L2}A^{[2,2]} & \dots & w_{LL}A^{[L,L]} \end{bmatrix} \in \mathbb{R}^{NL \times NL}$$

Where $A^{\otimes}$ is the Khatri-Rao product of the following matrices:

$$W = \begin{bmatrix} w_{11} & \dots & w_{1L} \\ \vdots & \ddots & \vdots \\ w_{L1} & \dots & w_{LL} \end{bmatrix} \text{and}(A^{[1,1]}A^{[2,2]}\dots A^{[L,L]})$$

$A^{\otimes}$ can be viewed as a giant adjacency matrix and can be used to find centrality by the measures defined on monoplex networks. To read up on the existence and uniqueness of the leading eigenvector of $A^{\otimes}$, please refer to (Solá *et al.* (2013)).

3.2.3 Cross Coupling-based Multilayer Networks

This is the most general case of coupling among all the methods. A node-layer pair (i, α) can be influenced by the nodes in the same layer $((j, \alpha) : i \neq j)$ as well as nodes of any other layer $((j, \beta) : \alpha \neq \beta)$ as shown in Fig. 2.1. Note that the coupling techniques discussed in earlier sections are the special cases of this particular coupling, which makes it essential to extend (or reformulate) the centrality measures for these networks. We do so by adopting as is the framework of random walk on weighted (monoplex) graphs, and only changing the graph on which this framework is applied (viz., applying this random walk framework as is to a large weighted graph constructed out of our multilayer network as described below).

In order to define the centrality measures first, we introduce the random walk operator for a multilayer network $G = (V, \mathbb{L}, E)$. A weighted multilayer network can have weights associated to the edges. Let $w_{ij}(\alpha, \beta)$ be the weight of the edge between (i, α) and (j, β). Let $s_{i\alpha} = \sum_{j,\beta} w_{ij}(\alpha, \beta)$ be the node strength of (i, α). We can write the transition probability from (i, α) to (j, β) as

$$\mathbf{T}(i_\alpha, j_\beta) = \frac{w_{ij(\alpha,\beta)}}{max(s_{i\alpha}, \epsilon)},$$

where $\epsilon > 0$ is a constant. Note that $\mathbf{M}(i_\alpha, j_\beta)$ can also be viewen as a 4-dimensional tensor (also known as tensor). At time t, let $p_{i\alpha}(t)$ be the probability of finding the random walker at (i, α). Then,

$$p_{j\beta}(t+1) = \sum_{(i,\alpha)\in V_M} T^{i\alpha}_{j\beta} p_{i\alpha}(t).$$

The steady state solution for the random walk can be given by the leading *eigentensor* (Solé-Ribalta *et al.* (2016*a*)),

$$\mathbf{T}(i_\alpha, j_\beta)\Pi_{i\alpha} = \lambda \Pi_{j\beta}.$$

Intuitively a random walker should visit the nodes with high strength more frequently than those with lesser strength. This is also evident from the formulation as, $\pi_{i\alpha} \propto s_{i\alpha}$. We can use this formulation to define different centrality measures on multilayer networks. Following the discussion from (De Domenico *et al.* (2015)), in PageRank, a walker can move from one node to its neighbor with probability p, and it can teleport to any other node with probability $(1-p)$. Considering the uniform probability of getting a node picked while teleportation, the transition matrix can be given by,

$$\mathbf{R}(i_\alpha, j_\beta) = p\mathbf{T}(i_\alpha, j_\beta) + \frac{(1-p)}{N}\mathbf{U}(i_\alpha, j_\beta),$$

where $\mathrm{U}(i_\alpha, j_\beta)$ assigns a uniform probability of transition between any pair of nodes. PageRank centrality for multilayer networks is given by the solution of this master equation. In addition to using the supra-adjacency list and supra-adjacency matrix representation of multilayer networks, there are other popular approaches such as representing the multilayer network as a collection of nodes and multi links (Iacovacci *et al.* (2016)). Since cross-coupling-based multilayer networks represent the most generalized form of multilayer networks, we explore it in detail in the next part of this thesis, with a specific focus on deriving new centrality measures.

3.3 SUMMARY

In this chapter, we discussed different types of inter-layer coupling in multilayer networks, their use cases, and their implications on centrality measures. We show that cross-layer-based coupling is the most generalized form of multilayer network. In the upcoming chapters, we will discuss the applicability of centrality measures to this form of multilayer network and its applications in understanding biological networks.

Part II

Multilayer Networks: Centrality and its Applications

CHAPTER 4

MultiCens: Multilayer Network Centrality framework

Network centrality assesses the "importance or centrality" of each node in a network by quantifying how well-connected or well-knit each node is to other nodes in the network. Network centrality has numerous applications, including finding influential people in a social network, studying the emergence of network structure, and detecting essential proteins or disease genes in biological networks. Traditional centrality measures are limited to single-layered graphs where nodes can be involved in only one type of interaction. With advances in data collection, we may often have access to more than one view of a complex system. For instance, research interactions among a set of authors can be viewed via co-authorship, co-citation, and other types of relations; state variables of a time-evolving system may exhibit varied behaviors and interaction patterns in different time points; and high-throughput technologies in biology permit us to measure the activity of the same set of genes in different tissues, yielding multiple layers of the biological system. A central open question in extending centrality measures to multilayer networks is the assessment of the local effect of a node in its layer vs. the global effects of the node in the overall network or in specific target layers or target node sets. In this study, we propose several PageRank-like iterative centrality measures that offer these local vs. global effects of nodes by decomposing the overall multilayer centrality into relative contributions from intra-layer vs. inter-layer edges. We show that these measures have desirable theoretical properties like decomposability and derive the necessary conditions for convergence. For validation, we use our proposed centrality measures with several multilayer networks from human multi-tissue datasets and discover genes involved in tissue-tissue communication.

4.1 INTRODUCTION

Multilayer networks have found immense applications in ecological systems (Pilosof *et al.* (2017)), biological systems (Halu *et al.* (2019)), transport systems (Aleta *et al.* (2017)), social network analysis (Türker and Sulak (2018)), etc. In this work, we focus on centrality in multilayer networks which has found applications in the study of the emergence of congestion in transport flows (Solé-Ribalta *et al.* (2016*b*)), ranking in evolving networks (Liao *et al.* (2017)), and analyzing different life stages in the species (Shinde and Jalan (2015)). In multilayer networks, the neighborhood of a node can comprise nodes from the same layer as well as nodes from other layers. Based on the neighborhood, centrality measures such as PageRank can be used to find the centrality of nodes. The working of the PageRank can also be viewed as a random web-surfer model that explores the neighborhood of nodes using a random walk operator. These random walks are of infinite length and have a restart probability associated with them. In this work, we propose several PageRank-like centrality measures that differ in their ways of exploring within-layer or across-layer neighborhoods of the random walk operator. The different strategies proposed by us explore the multilayer structure of the network intelligently and help us answer several interesting enigmas of biology. Our main contribution is in developing a gene importance or centrality measure that quantifies the extent to which each gene in a tissue influences a query set of genes of interest in another tissue, both directly and indirectly via inter/intra-tissue interactions.

In this chapter, we start with the existing methods for multilayer network centrality and discuss their limitations. Then we proceed to the proposed centrality framework and define different centrality measures along with their convergence analysis.

We emphasize another theoretical aspect of the proposed centrality measures, decomposability, which ensures that the overall centrality of nodes in a multilayer network can be decomposed to local vs. global level, a layer-specific level, and further to a node-set level. By having this property, the existing works and upcoming advancements for PageRank centrality can be applied to the proposed set of centrality measures.

In order to discuss the existing methods, assume a multilayer network $G = (V, \mathbb{L}, E)$, where V represents the set of n nodes which is the same across all layers, $\mathbb{L}$ is the set of L number of layers, and E represents the set of inter- and intra- layer edges. The total number of nodes in the multilayer network is $N = n \times L$. Let G be represented by the supra-adjacency matrix M of dimension $N \times N$.

The supra-adjacency matrix can further be decomposed to represent the network with only intra-layer edges by A and the network with only inter-layer edges by C such that,

$$M = A + C$$

$$\begin{bmatrix} A^{[1,1]} & C^{[1,2]} & C^{[1,3]} & \cdots \\ C^{[2,1]} & A^{[2,2]} & C^{[2,3]} & \ddots \\ C^{[3,1]} & C^{[3,2]} & A^{[3,3]} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} A^{[1,1]} & 0 & 0 & \cdots \\ 0 & A^{[2,2]} & 0 & \ddots \\ 0 & 0 & A^{[3,3]} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} + \begin{bmatrix} 0 & C^{[1,2]} & C^{[1,3]} & \cdots \\ C^{[2,1]} & 0 & C^{[2,3]} & \ddots \\ C^{[3,1]} & C^{[3,2]} & 0 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

Here, A represents adjacency matrices for each layer along the diagonal, and C represents edges between different pairs of tissues at off-diagonal entries. Both A and

C are of dimension $N \times N$, and are composed of $n \times n$ block submatrices $\{A^{[i,i]}\}_{i=1,\ldots,L}$ and $\{C^{[i,j]}\}_{i,j=1,\ldots,L;i\neq j}$ respectively as shown here and in Fig. 4.1b.

In this work, we assume our multilayer network to be *undirected*; thus M, A, C, and $A^{[i,i]}$ for each i are symmetric matrices. In this work, we model multi-tissue data as a multilayer network, where each tissue is represented by a layer and genes are represented by nodes. In the subsequent sections, we discuss the construction of these networks in more detail.

4.2 EXISTING METHODS

The existing methods for finding multilayer centrality either emphasize only the inter-layer degree of the nodes or do not distinguish between within-layer and across-layer connections. Though such methods have revealed exciting properties of the underlying system, our goal is to develop centrality-based methods that can identify different effects of nodes, such as within-layer, across-layer, to a target layer, or a query set of nodes in a target layer. In this section, we discuss the existing methods of finding multilayer network centrality.

4.2.1 Degree-based Centrality and S_{sec}

Degree-based centrality, as the name suggests, counts the degree of a node. In the case of weighted networks, the weights of the edges connected to nodes can be added. In multilayer networks, nodes can have intra-layer connections and inter-layer connections. Thus degree centrality can be defined in two ways based on the type of connections under consideration. Intra-layer degree can be calculated by summing the weights of edges a node has within its own layer.

Definition 4.2.1. *Intra-layer centrality vector of a multilayer network can be computed by the following equation.*

$$deg_{intra} = A\vec{1} \tag{4.1}$$

where $\vec{1}$ *is the vector of all ones.*

Inter-layer degree is a count of the edges that cross the layers. These edges make the backbone of layer-layer communication. The inter-layer degree can be computed using the C matrix as follows.

Definition 4.2.2. *Inter-layer centrality vector of a multilayer network can be computed by the following equation.*

$$deg_{inter} = C\vec{1} \tag{4.2}$$

This inter-layer degree centrality vector is called S_{sec} *score vector when the weight of each edge in the multilayer network is given by* $-\ln($*P-value used to determine the statistical significance of correlation between the two nodes linked by the edge in a given observational dataset).*

The study that proposed this S_{sec} score vector Seldin *et al.* (2018) had used it to find prominent hormone-encoding genes that are strongly connected in a pair of tissues. Recently, degree and connectivity patterns such as shortest paths in multilayer networks are being deployed to complete private data with the help of open datasets Malek *et al.* (2020). Apart from degree-based centrality, there are methods such as PageRank centrality that can capture multi-hop effects in a network. We will now discuss an existing framework that extends PageRank centrality to a multilayer network.

4.2.2 Versatility

Domenico et al., in their seminal paper (De Domenico *et al.* (2015)), described a mathematical framework for centrality computation in multiplex networks. The proposed approach assigns a ranking to the nodes based on their interconnectedness. By setting proper weights of the layers (based on the number of nodes/edges), such a ranking method can reveal versatile nodes in the network. For a user-defined constant $p \in [0, 1)$, and N dimensional vector $\vec{1}$, the *versatility* vector can be defined as follows:

Definition 4.2.3. *Multilayer network PageRank centrality (also known as versatility (De Domenico* et al. *(2015))) x of a supra-adjacency network can be defined by the following equation.*

$$x = pMx + \frac{(1-p)}{N}\vec{1} \tag{4.3}$$

$$x = \Big(I - pM\Big)^{-1}\Big(\frac{(1-p)}{N}\vec{1}\Big)$$

The method itself does not distinguish between the within-layer and cross-layer edges, thus making it unavailing to distinguish the local vs. global effect of nodes. However, the mathematical formulation of a multilayer network described in this work can be extended to define the desired centrality measures, as we will discuss in the upcoming parts.

4.2.3 Other Works

NicheNet Browaeys *et al.* (2020) is a method to predict relations between ligands and their targets between interacting cells. The technique utilizes gene expression data, gene regulatory networks, and prior knowledge of cell signaling. NicheNet predicts genes associated with ligands in a two-layer setting by applying a personalized PageRank centrality algorithm to the overall network. This method is not applied to a multilayer setting where different tissues can have the same set of genes, and connections crossing layer boundaries have different semantics than within-layer connections. In the upcoming parts of this section, we will discuss how the MultiCens centrality framework applies to any number of layers.

4.3 PROPOSED METHODS

In the last section, we discussed methods based on inter-layer degrees and PageRank. Both these methods have shown their usefulness in revealing information about the underlying system. However, the multilayer structure of the network allows us to capture the effect of nodes at multiple levels such as within layer, across the layer, to a target layer, or a query set of genes in a target layer. The existing methods do not capture these effects and thus are limited in their usability. Capturing such effects can have immediate applications in several areas, such as systems biology, where we can identify genes that regulate hormonal communication between two tissues via multiple hops. In order to capture such effects, we propose a set of centrality measures, as shown in Figure 4.1. We begin by defining a multilayer network using a multi-tissue dataset. Figure 4.1(a) shows the multilayer network model representing the Hypothalamus-

Pituitary-Adrenal axis. Here, each layer represents a tissue, and each node represents a gene in the tissue. Nodes are connected to represent gene-gene connections that can be based on high gene co-expression scores, participation in protein complexes, frequent co-appearance in literature, etc. In this work, we call this multilayer network a “multi-tissue" network, where each layer can be a “tissue" network in itself.

Overview of Our Proposed Centrality Measures: MultiCens

We introduce a set of centrality measures, termed MultiCens, to quantify the influence or effect a gene has at different levels of granularity, such as the effect a gene has (i) “locally” within a tissue due to its connections to other genes in the tissue, or (ii) “globally” across all tissues due to within- as well as across-tissue connections, or specifically (iii) to a particular tissue, or (iv) to a query set of genes in a particular tissue. MultiCens measures account for the multilayer, multi-hop network connectivity of the underlying system in a hierarchical fashion, by decomposing the overall centrality (*versatility* pioneered by Domenico et al. De Domenico *et al.* (2015)) of a gene into *local* vs. *global* centrality, and further into *layer-specific* centrality specific to a tissue (referred to interchangeably as layer) or *query-set* centrality specific to a gene set in a tissue (see hierarchical organization in Fig. 4.1). We prove theoretical guarantees on the convergence and decomposability of MultiCens measures in this chapter and demonstrate empirical applications of MultiCens to simulated networks as well as real-world healthy and disease multi-tissue datasets in the next chapter.

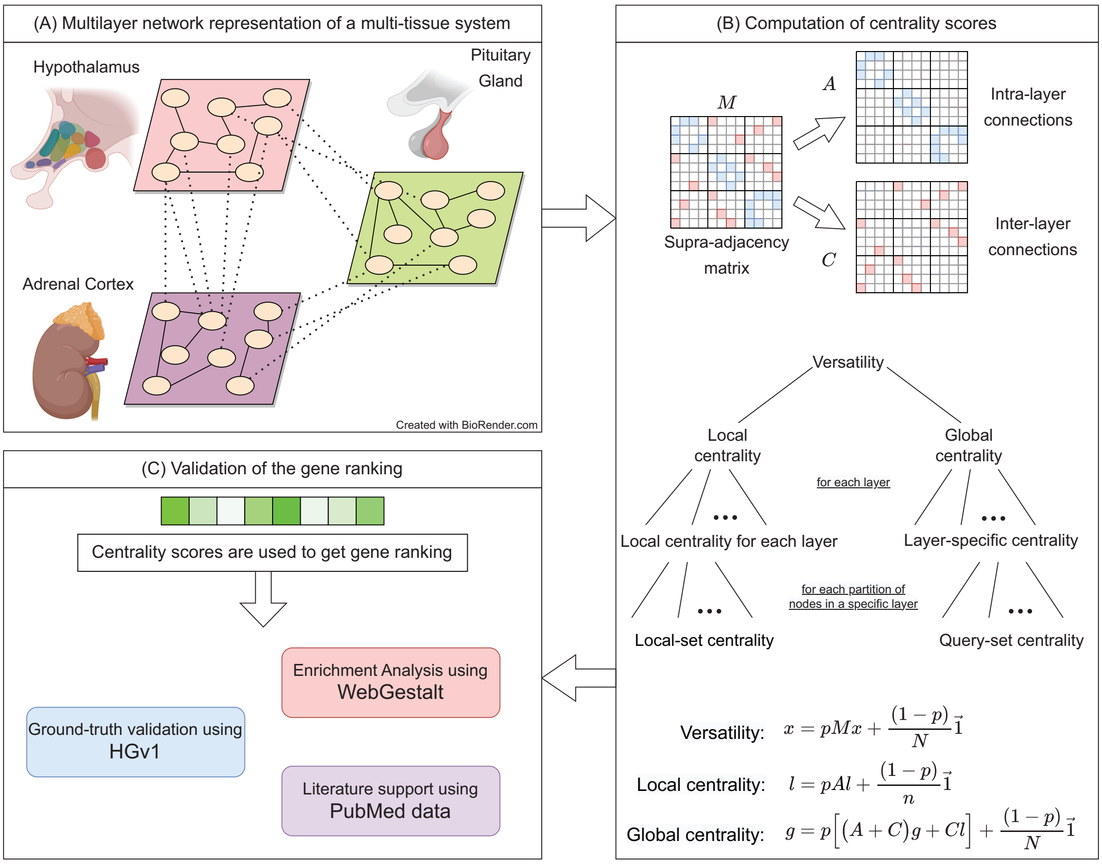

(A) Multilayer network representation of a multi-tissue system
Hypothalamus
Pituitary Gland
Adrenal Cortex
Created with BioRender.com
(B) Computation of centrality scores
M
Supra-adjacency matrix
A
Intra-layer connections
C
Inter-layer connections
Versatility
Local centrality
Global centrality
for each layer
Local centrality for each layer
Layer-specific centrality
for each partition of nodes in a specific layer
Local-set centrality
Query-set centrality
Versatility: $x = pMx + \frac{(1-p)}{N}\vec{1}$
Local centrality: $l = pAl + \frac{(1-p)}{n}\vec{1}$
Global centrality: $g = p\Big[(A+C)g + Cl\Big] + \frac{(1-p)}{N}\vec{1}$
(C) Validation of the gene ranking
Centrality scores are used to get gene ranking
Enrichment Analysis using WebGestalt
Ground-truth validation using HGv1
Literature support using PubMed data

Figure 4.1: **Workflow of our MultiCens measures.** (A) Each layer in the network represents a tissue, and connections represent gene-gene interactions (e.g., inferred from transcriptomic data). (Created with BioRender.com) (B) Supra-adjacency matrix (M) contains within-tissue connections on the diagonal blocks (intra-layer matrix A), and across-tissue connections on the off-diagonal blocks (inter-layer matrix C). The A, C matrices are used to compute different hierarchically-organized centralities as shown (note: the "collectively exhaustive node-sets" mentioned actually partition all the nodes in a layer or the network; see text). The centrality vectors (x, l, and g) have an entry for each gene in every tissue. (C) The centrality scores are used to obtain gene rankings which are further validated using different methods, and interpreted to predict novel mediators of inter-tissue signaling.

4.3.1 Local Centrality

A node in a layer can affect other nodes in the same layer as well as different layers. In order to capture the within layer effect of a node, we define the local centrality as follows:

Definition 4.3.1. *Local centrality vector of a multilayer network is given by the following iterative equation.*

$$l = pAl + \frac{(1-p)}{n}\vec{1} \tag{4.4}$$

Local centrality vector for a particular layer i is defined by the following iterative equation.

$$l_{layer_i} = pA^{[i]}l_{layer_i} + \frac{(1-p)}{n}\vec{1^{[i]}} \tag{4.5}$$

where $A^{[i]}$ represents matrix A with all but the i^{th} column-block entries set to 0 (note: i^{th} column-block of A represents adjacency matrix of the layer i), and $\vec{1^{[i]}}$ is a vector with entries for the nodes in layer i set to 1 and 0 otherwise.

It can be noticed that the local centrality of a node is defined by using only within-layer connections; thus, it does not capture any effects beyond the layer where the node is located. The PageRank centrality does not distinguish between within-layer connections and across-layer connections. Since *local centrality* considers the effect of only within-layer connections, the remaining effect is captured by *global centrality*.

4.3.2 Global Centrality

The global centrality of a node is a measure of its influence on all nodes irrespective of their layers. While computing this centrality score, we use both - within and across tissue connections in the following manner.

Definition 4.3.2. *For a given local centrality vector l, global centrality vector in a multi-layer network can be defined by the following iterative equation*

$$g = p\Big[\big(A + C\big)g + Cl\Big] + \frac{(1-p)}{N}\vec{1} \tag{4.6}$$

The *global centrality* of a node can be thought of as seeing an infinite length random walker on that node where at each step, the random walker can do one of the following.

1. With probability p,
   (a) Jump to a neighboring node $v_{n'}$ in the same layer with probability proportional to the weight of the connection.
   (b) Jump to a neighboring node $v_{n'}$ in a different layer with probability proportional to the weight of the connection and the local centrality of $v_{n'}$.
2. Restart the walk from any node in the network with probability $(1 - p)$.

The global centrality of a node results from the effect of all the layers present in the network. For some applications like hormonal communication in a multi-tissue system, the effect on a specific layer (tissue) that either produces or responds to a hormone can be of particular interest. So we decompose the effect of *global centrality* into *layer-specific* centrality.

4.3.3 Layer-specific Centrality

We are interested in finding the effect of node(s) on a specific layer (target layer) in the multilayer network. In doing so, we define the *layer-specific* centrality as follows.

Definition 4.3.3. *For a given local centrality vector for layer i, layer-specific centrality vector in a multilayer network can be defined by the following iterative equation.*

$$g_{layer_i} = p\Big[\big(A + C\big)g_{layer_i} + Cl_{layer_i}\Big] + \frac{(1-p)}{N}\vec{1}^{[i]} \tag{4.7}$$

(note: the Cl_{layer_i} term effectively uses only the ith column-block of C, i.e., the block representing all inter-layer edges that are incident to some node in layer i)

Our proposed centrality framework is highly generic, and the definition of centrality can further be customized to capture the effect of a node on a set of nodes on a specific target layer. We propose another refinement in the *layer-specific* centrality by decomposing it into multiple query-node sets in the specific target layer.

4.3.4 Query-set Centrality

We introduce query-set centrality that can capture the effect of a node on a query-set of nodes present in any specific layer in the multilayer network. We begin by defining local-set centrality, a variant of local centrality focused on a query set of nodes in a specific layer.

Definition 4.3.4. *For a given set of query nodes set_k present in layer i, the local-set centrality vector in a multilayer network can be defined by the following equation.*

$$l^{set_k}_{layer_i} = pA^{[i]}l^{set_k}_{layer_i} + \frac{(1-p)}{n}\vec{1}^{k}_{layer_i}, \tag{4.8}$$

where $\vec{1}^{k}_{layer_i}$ *represents the vector of* $1's$ *at indices corresponding to the nodes in* set_k *in layer i and* 0 *otherwise. Note that query nodes* set_k *is restricted to be in the target layer* i *alone.*

We use this *local-set* centrality to define *query-set* centrality as follows.

Definition 4.3.5. *For a given set of query genes* set_k *in a layer* i*, the query-set centrality in a multilayer network can be defined by the following equation.*

$$g^{set_k}_{layer_i} = p\Big[\big(A + C\big)g^{set_k}_{layer_i} + Cl^{set_k}_{layer_i}\Big] + \frac{(1-p)}{N}\vec{1}^{k}_{layer_i} \tag{4.9}$$

The *query-set centrality* is defined in order to capture the effect of nodes on a query-set of genes in a specific target layer. As shown in Fig. 4.1, our centrality equations are based on the principle of decomposability.

Convergence of MultiCens centrality measures

We now prove the convergence of the proposed centrality measures. The *local centrality* measure is similar to the Pagerank centrality and its convergence follows from the Pagerank centrality convergence itself. Whereas, *global centrality* has additional terms in the equation and we provide a proof for its convergence.

Lemma 4.3.1. *For* $0 \leq p < 1$*, global centrality, as defined by Equation 4.6 always converges.*

Proof. From equation 4.6,

$$
\begin{aligned}
g &= p\Big[(A+C)g + Cl\Big] + \frac{(1-p)}{N}\vec{1} \\
&= p\left[(A+C)\Big(p\big[(A+C)g + Cl\big] + \frac{(1-p)}{N}\vec{1}\Big) + Cl\right] + \frac{(1-p)}{N}\vec{1} \\
&= p\left[p(A+C)^2 g + p(A+C)Cl + (A+C)\frac{(1-p)}{N}\vec{1} + Cl\right] + \frac{(1-p)}{N} \\
&\vdots \\
&= p^k(A+C)^k g + p\sum_k p^k(A+C)^k Cl + \sum_k p^k(A+C)^k\frac{(1-p)}{N}\vec{1} + \frac{(1-p)}{N}\vec{1}
\end{aligned}
$$

The first term on the right side converges as k grows larger. The second and third terms give rise to two geometric series generated by $p(A+C)$. We know that $(A+C)$ is a row stochastic matrix and the product $(p(A+C))$ can have maximum eigenvalue, $|\lambda'| < 1$. A geometric series generated by a matrix with eigenvalues less than 1 always converges. This completes the proof. ☐

Lemma 4.3.2. *For $0 \leq p < 1$, g_{layer_i} defined by Equation 4.7 always converges.*

Proof. Following the steps from Lemma 4.3.1, the layer-specific centrality (Equation 4.7) can be written as:

$$
g_{layer_i} = p^k(A+C)^k g_{layer_i} + \left(p\sum_{k'=0}^{k-1} p^{k'}(A+C)^{k'} Cl_{layer_i}\right) + \left(\sum_{k'=0}^{k-1} p^{k'}(A+C)^{k'}\frac{(1-p)}{N}1^{\vec{[i]}}\right)
$$

The right-hand side of the equation results in multiple geometric series, and all of them converge as the number of iterations increases. This completes the proof. ☐

Lemma 4.3.3. *For* $0 \leq p < 1$, $l^{set_k}_{layer_i}$ *defined by Equation 4.8 always converges.*

Proof. Following the steps from Lemma 4.3.1, we can write *local-set centrality* (Equation 4.8) as:

$$l^{set_k}_{layer_i} = (pA^{[i]})^j l^{set_k}_{layer_i} + \sum_{j'=0}^{j-1} (pA^{[i]})^{j'} \frac{(1-p)}{n} \vec{1}^k_{layer_i}, \text{ where } j \to \infty$$

The right side of the equation is similar to the original PageRank centrality which is known to converge for $0 \leq p < 1$. □

Lemma 4.3.4. *For* $0 \leq p < 1$, $g^{set_k}_{layer_i}$ *defined by Equation 4.9 always converges.*

Proof. Following the steps from Lemma 4.3.1, we can write query-set centrality (Equation 4.9) as:

$$g^{set_k}_{layer_i} = p^d(A+C)^d g^{set_k}_{layer_i} + \left(p \sum_{d'=0}^{d-1} p^{d'} (A+C)^{d'} C l^{set_k}_{layer_i} \right) + \left(\sum_{d'=0}^{d-1} p^{d'} (A+C)^{d'} \frac{(1-p)}{N} \vec{1}^k_{layer_i} \right)$$

The right-hand side of the equation results in multiple geometric series, and all of them converge as the number of iterations increases. This completes the proof. □

Theorem 4.3.5 (Convergence of MultiCens)**.** *For* $0 \leq p < 1$, *all MultiCens centrality measures, including local centrality, global centrality, layer-specific centrality, local-set centrality and query-set centrality as defined by Equations 4.4-4.9 converge.*

Proof. The local centrality measure, defined by Equation 4.4 is similar to the Pagerank centrality and its convergence follows the Pagerank convergence Page *et al.* (1999).

Lemmas 4.3.1-4.3.4 prove the convergence of global centrality, layer-specific centrality, local-set centrality and query-set centrality as defined by Equations 4.6-4.9.

This completes the proof. ☐

Decomposability of MultiCens centrality measures

Our centrality framework exhibits a special theoretical property called decomposability. We define *global centrality* and *local centrality* in a way that they add up to the *versatility* in the multilayer network, which the following proof can verify.

Lemma 4.3.6. *Versatility of a multilayer network can be decomposed into local centrality and global centrality with a scaling factor.*

Proof.

$$l + g = x \tag{4.10}$$

From equation 4.4

$$l = pAl + \frac{(1-p)}{n}\vec{1}$$

from equation 4.6

$$\begin{aligned}
g &= p\Big[\big(A + C\big)g + Cl\Big] + \frac{(1-p)}{N}\vec{1} \\
(l + g) &= p\Big[\big(A + C\big)g + \big(A + C\big)l\Big] + (1-p)(\frac{1}{n} + \frac{1}{N})\vec{1} \\
(l + g) &= p\Big[\big(A + C\big)\big(l + g\big)\Big] + \frac{(L+1)(1-p)}{N}(\vec{1}) \\
(l + g) &= \Big(I - p\big(A + C\big)\Big)^{-1}\Big(\frac{(L+1)(1-p)}{N}\vec{1}\Big) \\
(l + g) &= (L+1)\Big(I - p\big(M\big)\Big)^{-1}\Big(\frac{(1-p)}{N}\vec{1}\Big) \\
(l + g) &= (L+1)x
\end{aligned}$$

Where L is the total number of layers. Since l, g, and x are centrality vectors, they are scale agnostic, so the constant factor $(L+1)$ on the right side of the equation can be ignored. This completes the proof. ☐

We decompose *global centrality* into *layer-specific* centrality and further into *query-set* centrality in a way that instances of each centrality measure add up to their parent centrality measure.

Lemma 4.3.7. *Global centrality of a multilayer network can be decomposed into the layer-specific centrality of all layers, i.e.,*

$$\sum_{i=1}^{L} g_{layer_i} = g \tag{4.11}$$

Proof.

$$\begin{aligned}\sum_{i=1}^{L} g_{layer_i} &= p\Big[(A+C)\sum_{i=1}^{L} g_{layer_i} + C\sum_{i=1}^{L} l_{layer_i}\Big] + \sum_{i=1}^{L}\frac{(1-p)}{N}\vec{1}^{[i]} \\ \tilde{g} &= p\Big[(A+C)\tilde{g} + Cl\Big] + \frac{(1-p)}{N}\vec{1} \\ \tilde{g} &= g\end{aligned}$$

This completes the proof. ☐

Lemma 4.3.8. *For a layer i, its local centrality vector can be decomposed into local-set centrality of sets $\{set_k\}_{k=1,\dots,K}$, where $\{set_k\}_{k=1,\dots,K}$ is a partition of all nodes in layer i.*

$$\sum_{k=1}^{K} l_{layer_i}^{set_k} = l_{layer_i} \tag{4.12}$$

Proof.

$$\sum_{k=1}^{K} l_{layer_i}^{set_k} = pA^{[i]} \sum_{k=1}^{K} l_{layer_i}^{set_k} + \frac{(1-p)}{n} \sum_{k=1}^{K} \vec{1}_{layer_i}^{k}$$

$$\tilde{l} = pA^{[i]}(\tilde{l}) + \frac{(1-p)}{n} 1^{\vec{[i]}}$$

This equation is the same as the iterative equation defined for computing local centrality. This completes the proof. ☐

Lemma 4.3.9. *Layer-specific centrality of layer i can be decomposed into query-set centrality of sets $\{set_k\}_{k=1,\ldots,K}$ that together partition all nodes in layer i.*

$$\sum_{k} g_{layer_i}^{set_k} = g_{layer_i} \tag{4.13}$$

Proof.

$$\sum_{k} g_{layer_i}^{set_k} = p\Big[(A+C)\sum_{k} g_{layer_i}^{set_k} + C\sum_{k} l_{layer_i}^{set_k}\Big] + \frac{(1-p)}{N}\sum_{k} \vec{1}_{layer_i}^{k}$$

$$\tilde{g}_{layer_i}^{set_k} = p\Big[(A+C)\tilde{g}_{layer_i}^{set_k} + C\sum_{k} l_{layer_i}^{set_k}\Big] + \frac{(1-p)}{N}\sum_{k} \vec{1}_{layer_i}^{k}$$

By using Lemma 4.3.8

$$\tilde{g}_{layer_i}^{set_k} = p\Big[(A+C)\tilde{g}_{layer_i}^{set_k} + Cl_{layer_i}\Big] + \frac{(1-p)}{N} 1^{\vec{[i]}}$$

The right side of the equation is the same as equation 4.7. This completes the proof. ☐

Theorem 4.3.10 (Decomposability of MultiCens)**.** *In a multilayer network, versatility can be decomposed into local and global centrality, and global centrality into layer-specific centrality of all layers. Furthermore, layer-specific centrality of any layer can be decomposed into the query-set centrality of sets that collectively partition the nodes in the layer.*

Proof. Equation 4.10 presents the decomposability of versatility into *local centrality* and *global centrality*. Lemma 4.3.6 provides necessary proof for the decomposability of *versatility*.

Equations 4.11- 4.13 present the decomposability of MultiCens centrality measures. Lemmas 4.3.7-4.3.9 collectively prove the decomposability of centrality measures defined under MultiCens framework. This completes the proof. ☐

4.3.5 Complexity analysis of the proposed centrality measures

The power method can be used to implement our centrality measures, involving a matrix-vector product that can be computed in $\mathcal{O}(n^2)$ time at each iteration. After computing the products, adding vectors can be done in $\mathcal{O}(n)$ time, resulting in each iteration being executed with time complexity of $\mathcal{O}(n^2)$. The number of iterations required for convergence is another critical factor. Empirically, we found that a two-layered multilayer network with approximately 12000 nodes each requires 50 to 100 iterations to converge.

Convergence in the power method generally depends on the ratio of the dominant eigenvalue to the next largest eigenvalue Trefethen and Bau (2022), determining the required number of iterations for PageRank. Personalized PageRank's number of iterations also relies on seed nodes. If well-connected and with many outgoing links, the Personalized PageRank score can converge quickly as the seed nodes' influence spreads to other nodes in the network. The network's structure beyond the seed nodes can also affect the convergence rate, with networks with many tightly connected communities requiring more iterations to spread the seed nodes' influence across the clusters.

Similarly, our proposed centrality measures' iteration count depends on the query set of nodes' connectivity pattern. We also introduce an additional bias towards the target layer where the query set of nodes resides, impacting the number of iterations required. Further research could explore the relationship between iteration count and multilayer network structure in such centrality measures.

4.4 SUMMARY

Multilayer networks offer a natural representation of complex systems involving multiple forms of interactions among their constituents. Finding the centrality of nodes in multilayer networks has applications ranging from biological networks to transport networks and social systems. In this chapter, we began by introducing cross-coupled multilayer networks and discussed the existing forms of centrality methods. After emphasizing the need for a newer and more flexible framework for defining multilayer network centrality, we proposed a series of centrality measures. We discussed their theoretical properties, such as convergence and decomposability. The proposed framework of defining centrality to capture local and global effects can potentially be extended to other forms of centrality as well.

CHAPTER 5

Applications of MultiCens framework to analyze Biological Networks

In this chapter, we will discuss the applications of multilayer network centrality methods. We use the proposed centrality methods, particularly the most refined method *query-set* centrality, to identify genes involved in inter-tissue communication. We begin the discussion by presenting the scope of application in contrast with the related studies. The experimental details and results follow this, and we conclude with our findings and future works.

5.1 INTRODUCTION

For any multicellular organism with specialized tissue or organ structures, including humans, communication among the different tissues/organs is essential for the coherent integrated functioning of the whole body. The molecular mechanisms of such inter-organ communication, be it canonical communication routes such as the nervous system and hormonal system (or) non-canonical recently-discovered routes such as ones mediated by fat-derived extracellular vesicles (Huang and Xu (2021)) and microbiota-derived metabolites in the gut-brain axis, can be represented as a network of interactions among the biomolecules residing in different tissues/organs (and called the inter-organ communication network or ICN) (Droujinine and Perrimon (2013)). Rapidly gaining interest in the mapping of ICN (Droujinine *et al.* (2021)) and detailed mechanistic characterization of specific interactions in the ICN (Bodine *et al.* (2021)) have revealed a large ICN network among secreted proteins in model organisms like Drosophila, and the key roles played by certain ICN molecules or interactions in healthy and disease conditions. But these *in vivo* experimental techniques for ICN mapping or ICN analysis to identify key players are limited in non-model organisms, including humans, and also quite time-consuming even in model organisms due to the huge experimental space (to cover the quadratic number of all pairwise interactions among thousands of biomolecules in tens of tissues of interest). As a result, the ICN is vastly under-explored in both model as well as non-model organisms, and there is an immediate need to accelerate the mapping and analysis of ICNs in health and disease.

In this study, we deploy the proposed computational approaches to rapidly map and analyze a multi-tissue network, comprising not only inter-tissue but also intra-tissue gene-gene interactions. Our work is made possible by the recently accumulating multi-tissue genomic datasets (e.g., (Lonsdale *et al.* (2013); Wang *et al.* (2018*b*)), which can be used to infer inter/intra-tissue networks using the concept of gene-gene correlation or coexpression. Coexpression network mapping and analysis have been done before, for instance, using the popular WGCNA method (Langfelder and Horvath (2008)), and gene prioritization using network-based measures have also successfully guided downstream experiments before (Aerts *et al.* (2006); Schlicker *et al.* (2010); Moreau and Tranchevent (2012); López-Cortés *et al.* (2018); Kolosov *et al.* (2021)), but these existing studies have primarily focused on a single tissue of interest in a healthy condition or the single most affected tissue in disease. Our proposed multi-tissue centrality measures offer a systematic data/computation-driven prioritization of genes to be key regulators of inter-tissue signaling and thereby help generate hypotheses to guide downstream experiments that advance this emerging field of studying the whole body at the holistic (multiple organ/tissue) as well as molecular level.

The conceptual framework we use to model multi-tissue systems is more general, and can also be applied to study interactions among multiple regions of the brain, for instance, and to study the overall influence of a gene in the whole multi-tissue network or an entire tissue, rather than a specific query set of genes in a specific tissue. This is made possible by modeling this system as a multilayer network model, where each tissue/region contributes to a layer and nodes (genes) can have within-layer and across-layer connections (gene-gene interactions), and by proposing several PageRank-like iterative centrality measures that decompose the overall multilayer centrality of a node into relative contributions from intra- vs. inter-layer edges. The application focus of this study is multi-tissue network centrality analysis to discover genes responsible for inter-tissue communication via mediating hormones. We apply our proposed centrality measures to human multi-tissue datasets and retrieve genes that are involved in the production/processing/release of hormones in a source tissue or those that respond to hormones in the target tissues. Our study with well-studied hormones for humans, such as Insulin, identifies known as well as novel regulators of insulin signaling, with the latter including lncRNAs backed by biomedical literature support, either in terms of co-occurrence or in terms of similarity of literature-derived embedding vectors of hormones and gene symbols. The application to hormone-gene inter-tissue signaling is promising for broader applications of our work to understand communication between different functional structures within our human body. We now discuss the data collection steps and elaborate on the multilayer network construction.

5.2 DATASETS AND NETWORK GENERATION

We validate the proposed centrality methods on synthetic as well as real-world datasets. In this section, we discuss the process of constructing multilayer networks from raw data and making it available for experimentation.

5.2.1 Synthetic Multilayer Networks

To understand the working of our MultiCens measures, we generate an extensive set of synthetic multilayer networks. As shown in Fig. 5.1, we begin with a two-layered multilayer network where each layer has 500 nodes. Following the popular ER-random graph generation model Erdos *et al.* (1960), we consider all possible pairs of nodes (within and across layer) and put an edge with probability $p = 0.05$. This multilayer network is called the *base* network, and we mark 50 nodes in layer two as the query-set. On top of the base network, we add additional edges among the nodes in the query-set by another ER-based process of adding random edges. To add these additional edges, we vary this additional edge probability p (called *connection strength*) from $p = 0.05$ to $p = 1$ at steps of 0.05, and obtain a network structure at each step. If a node pair, say (i, j), gets connected in the base network and gets another edge while adding additional edges, we assign weight of two units to the original edge. Similarly, in the first layer, we mark a community of 50 nodes directly connected to the query-set, and call it *source set 1*. Another community of 50 nodes, *source set 2*, is connected to *source set 1*. The connection strength within these two communities and between *source set 1* and *source set 2*, and between *source set 1* and *query-set* is varied from 0.05 to 1. In our hormonal signaling example, *query-set* can be thought of as a set of genes that respond to a hormone, say insulin in skeletal muscle tissue. *Source set 1* and *source set 2* can be considered as genes in the pancreas tissue that interact with the *query-set* either by direct or two-hop long dense connections. Since the tissues will have multiple other clusters of genes that are not in the proximity of insulin-related genes, we mark three such communities of 50 nodes each. Connection strength within these three communities and across them is also varied.

In this synthetic multilayer network, our goal is to understand whether genes from *source set 1* and *source set 2* get top centrality-based ranks for a given *query-set*, across different values of connection strength.

5.2.2 Real-world Multilayer Networks

We evaluate our proposed centrality framework to reveal inter-tissue communication mediating genes in human multi-tissue systems and extend our experiments to Alzheimer's brain network application with four brain regions/tissues. For human multi-tissue datasets, we use the following resources.

GTEx.v8 Single-Tissue cis-QTL Data (Lonsdale *et al.* (2013)) [1]

This data is a result of the Genotype-Tissue Expression (GTEx) project. The dataset contains gene expression profiles of hundreds of individuals from over 30 tissues. The dataset is pre-processed to account for some known as well as derived covariates [2].

---

[1] File "GTEx_Analysis_v8_eQTL_expression_matrices.tar" accessed from "https://gtexportal.org/home/datasets" on Sep 25, 2020: 2100 hours IST

[2] We used the list of covariates from file "GTEx_Analysis_v8_eQTL_covariates.tar.gz" accessed from "https://gtexportal.org/home/datasets" on Sep 25, 2020, at 2100 hours IST.

Stanford Biomedical Network Dataset Collection (Zitnik *et al.* (2018)[3]

This dataset provides a tissue-specific protein-protein edge list for humans. The data is derived from a global protein-protein network. In the global interactions, if a pair of proteins is tissue-specific or if one protein is tissue-specific and the other protein is ubiquitous, then the tissue information is associated with the interaction, and hence the tissue-specific networks are obtained. Physical protein-protein interactions experimentally support the edges in the networks. We retrieve the hormone-producing and responding gene sets from HGv1 dataset[4] (Jadhav *et al.* (2022)). In HGv1, the source and target genes of hormones are first retrieved in a tissue-agnostic manner, and then through biomedical literature mining source and target tissues of a given hormone is designated. We treat these hormone producing and responding gene sets as the ground truth genes for hormonal signaling.

5.2.3 Hormone-related Network Construction

Gene coexpression networks are known to capture the patterns of underlying gene expression data that can reveal important biological biomarkers, functional associations between different genes, etc. In human experiments, we make use of the *GTEx.v8 Single-Tissue cis-QTL* data and compute Spearman correlation to find the correlation coefficients between all gene pairs (within and across tissue) and use it as an edge weight (absolute value) to signify the strength of interactions. In order to avoid the

---

[3] File "PPT-Ohmnet_tissues-combined.edgelist" accessed from "`http://snap.stanford.edu/biodata/datasets/10013/10013-PPT-Ohmnet.html`" on Sep 25, 2020, at 2100 hours IST.

[4] Files accessed from "https://cross-tissue-signaling.herokuapp.com/" on Jan 10, at 1600 hours IST.

blowup in the size of the multilayer network, we only use the top $10k$ varying genes in each tissue and take the union of these genes while constructing the multilayer network.

We also use the protein-protein interaction data as described earlier, in addition to using a gene coexpression network. For every gene-gene pair, if it is present in the protein interaction data, we increase its weight by 1 unit (adding edge weights) and work with the resultant network. In this thesis, we report results on this resultant network unless mentioned otherwise.

In GTEx dataset, combining multiple tissues in a network leads to fewer common samples and, hence, a less robust network; we restrict these experiments to multilayer networks only with two tissues (the predominant source and target tissue for a hormone; so these multilayer networks we construct and analyze are hormone-specific). However, our network generation mechanism as well as the MultiCens framework to compute centrality can be readily used for any number of tissues, as we illustrate in the Alzheimer's brain network application with four brain regions/tissues.

Evaluation of Hormone-gene Predictions

In one of MultiCens' applications, we use hormone-producing set as the *query-set* of genes and rank all genes in the target tissue to predict the hormone-responsive set; this process is repeated vice versa to predict hormone-producing genes from an input query set of hormone-responsive genes. We use the HGv1 database Jadhav *et al.* (2022) as ground truth and validate our gene rankings against it. We also perform disease enrichment analysis to find that whether our centrality-based gene rankings are enriched for hormone-related diseases using WebGestalt[5]. To obtain the enriched set of diseases for human gene rankings, we use the WebGestalt portal and select "Homo sapiens" as the Organism of Interest. Method of interest and Functional Database are set to Gene Set Enrichment Analysis (GSEA) and disease, respectively. We select OMIM functional database and set the significance level to 0.05 FDR. We give the gene symbols, and their corresponding centrality scores as input, and the portal returns the set of diseases enriched at the given FDR cut-off.

From the gene rankings obtained using our centrality measure, we find the support for top protein-coding genes based on co-occurrence with hormone-related terms in the PubMed corpus[6]. More information about these evaluation approaches are given below.

Recall-at-k plot

This plot can be used to validate the results visually. Both in synthetic as well as real-world datasets, we have a set of ground truth genes that we expect to come at the top as per their centrality scores. This can be verified by visualizing *recall-at-k* plots where the x-axis reports the top k predictions and the y-axis marks the number of hits from the ground truth at any given k.

---

[5] Tool `http://webgestalt.org/` accessed on Aug 5, 2021.

[6] Data accessed from “https://pubmed.ncbi.nlm.nih.gov/" on Aug 1, 2021.

Area under *recall-at-k* curve

Higher recall-at-k curve implies the better performance of a method. One way to quantify it is by calculating the area under it. We normalize the maximum possible area under *recall-at-k* curve to be 1 and report the area obtained by curves corresponding to the proposed method.

Support from literature

The evaluation metrics discussed above require the ground truth for evaluation. Many times, especially in biology, it is tough to have access to the complete ground set of hormone-producing/responding genes. Continuous research like this study pushes our knowledge boundaries, and we get access to more reliable and more complete ground truth datasets. In order to validate the novel findings, we rely on support from literature and use the following two metrics.

Co-occurrence in the PubMed database: We use articles present in the PubMed data and find the support for predicted genes. The support is calculated as an overlap between the gene name and the hormone/disease name. The support is calculated using the following formula.

$$Support = \frac{H \cap G}{\frac{H}{\text{number of articles on PubMed}} \times G}$$

Where H and G denote the number of articles that mention the hormone name and gene name, respectively, and $H \cap G$ denote the number of articles that contain both the hormone name and gene name. While finding support for the gene-disease association, we use articles that contain the disease name instead of hormone name. We use 27 million as the number of articles present in the PubMed database.

Cosine similarity in the embedding space: We find cosine similarity between the embedding vector of a gene symbol and that of a hormone or disease name. Since cosine similarity can range between -1 and 1, a positive number indicates that the gene-hormone or gene-disease association is supported in the embedded space. Our embeddings (also called as word embeddings or embedding vectors) are from BioWordVec[7], a deep learning model pretrained on the PubMed corpus Zhang *et al.* (2019). Both these metrics use articles present in the PubMed database, but they differ because the co-occurrence is based solely on the presence of two terms in an article, whereas the second metric also captures the contextual dependencies in the embedding space.

Our PubMed literature analysis focuses only on the peptide hormones insulin and somatotropin (out of all the four primary hormones considered), since we wanted to apply an informative filter to inspect predictions that are only among genes involved in peptide secretion[8]. This filter was inspired by a similar filter applied in an earlier study on endocrine interactions Seldin *et al.* (2018).

For the second application, we apply MultiCens framework to four-layered human brain multilayer network and identify shift in the gene rankings of control and AD groups.

---

[7] BioWordVec model/embeddings are downloaded from `https://github.com/ncbi-nlp/BioSentVec`.

[8] List of genes involved in peptide secretion accessed from this URL- www.ebi.ac.uk/QuickGO/GTerm?id=GO:0002790 on Dec 1, 2020

5.2.4 Multi-brain-region Data - Preprocessing and Correction

The covariate-adjusted transcriptomic (RNA-sequencing) data with the following synapse ids - syn16795931 – Brodmann Area (BM10) – frontal pole (FP), syn16795934 - BM22 - superior temporal gyrus (STG), syn16795937 - BM36 - parahippocampal gyrus (PHG), syn16795940 – BM44 - inferior frontal gyrus (IFG), were downloaded from AD Knowledge Portal – The Mount Sinai/JJ Peters VA Medical Center Brain Bank cohort (MSBB) study Wang *et al.* (2018*a*) (10.7303/syn3159438). The preprocessed data is corrected for library size differences using the trimmed mean of M-values normalization (TMM method – edge R package) and linearly corrected for sex, race, age, RIN (RNA Integration Number), PMI (Post-Mortem Interval), sequencing batch, exonic rate and rRNA (ribosomal RNA) rate. The normalization procedure was performed on the concatenated data from all four brain regions to avoid any artificial regional difference Wang *et al.* (2018*a*).

The clinical (MSBB_clinical.csv) and experimental metadata (MSBB_RNAseq_covariates_November2018Update.csv) files available on the portal are used to classify the samples into control (CTL) and Alzheimer's disease (AD) based on CERAD score (Consortium to Establish a Registry for AD). CERAD score 1 was used to define CTL samples, and 2 ('Definite AD') was used for defining AD samples Wang *et al.* (2018*a*). Probable AD (CERAD = 3) and Possible AD (CERAD = 4) samples were not considered for this study.

To mitigate the confounding effect of cellular composition on gene-gene coexpression relations, we corrected (linearly adjusted) the RNAseq gene expression data for cell type frequencies of four major brain cell types: astrocytes, microglia, neuron, and oligodendrocytes. We estimated these cell type frequencies in each brain region/tissue separately from the bulk tissue expression of the marker genes of these cell types using a cellular deconvolution method called CellCODE (Cell-type Computational Differential Estimation) Chikina *et al.* (2015). Specifically, we used the getAllSPVs function from the CellCODE, and provided its input arguments to select robust marker genes that do not change between AD vs. CTL groups (specified via the mix.par argument set at 0.3) from a starting set of 80 marker genes (top 20 per cell type, obtained from the BRETIGEA (BRain cEll Type specIfic Gene Expression Analysis) meta-analysis study McKenzie *et al.* (2018).

Network Construction and Enrichment Analysis of Gene Rankings

AD and CTL networks are separately constructed as before by computing the Spearman correlation between all pairs of genes in the four brain regions and taking absolute value of these correlations as the edge weights. To make the analysis computationally tractable, we restrict our focus to a subset of genes as follows - identify the 9000 most varying genes in each region for both AD and CTL populations, and then consider the union of all these gene sets as the final set of nodes in each layer of the multilayer network.

MultiCens scores are then calculated for all the nodes in the AD or CTL multilayer networks to obtain gene rankings, which were then subjected to enrichment analysis with WebGestalt as described before. Additionally, we applied redundancy reduction methods (affinity propagation and weighted set cover) and selected the significantly enriched terms, which passed both the methods. We use the centrality score of each of the three brain regions other than the query brain region to find the significantly enriched terms considering both Reactome pathways and Gene Ontology based Biological Process (GO-BP).

5.3 RESULTS

In this section, we present the results obtained using synthetic as well as real-world datasets.

5.3.1 Results with Synthetic Multilayer Networks

Figure 5.1 shows the construction and evaluation using synthetic multilayer networks. As discussed, we work with a two-layered multilayer network with a *query-set* present in the second layer. We begin with a configuration where only *source-set-1* makes the ground truth and rank all nodes in the first layer. In the *recall-at-k* plots, we show the overlap between the ground truth and the top 100 ranked nodes. It can be seen that all three methods can recover the ground truth as the density of edges in communities increases. As we start changing the ground truth by removing nodes from *source-set-1* and adding nodes from *source-set-2*, methods other than *query-set centrality* start failing. In the end, we reach a configuration where 100% nodes in the ground truth are from *source-set-2* where the proposed method outperforms the other methods drastically. In the context of predicting hormone-gene associations, it is unlikely when only the genes that are not directly connected to query-set produce or respond to a hormone. In real-world multilayer networks, we expect a mixture of nodes

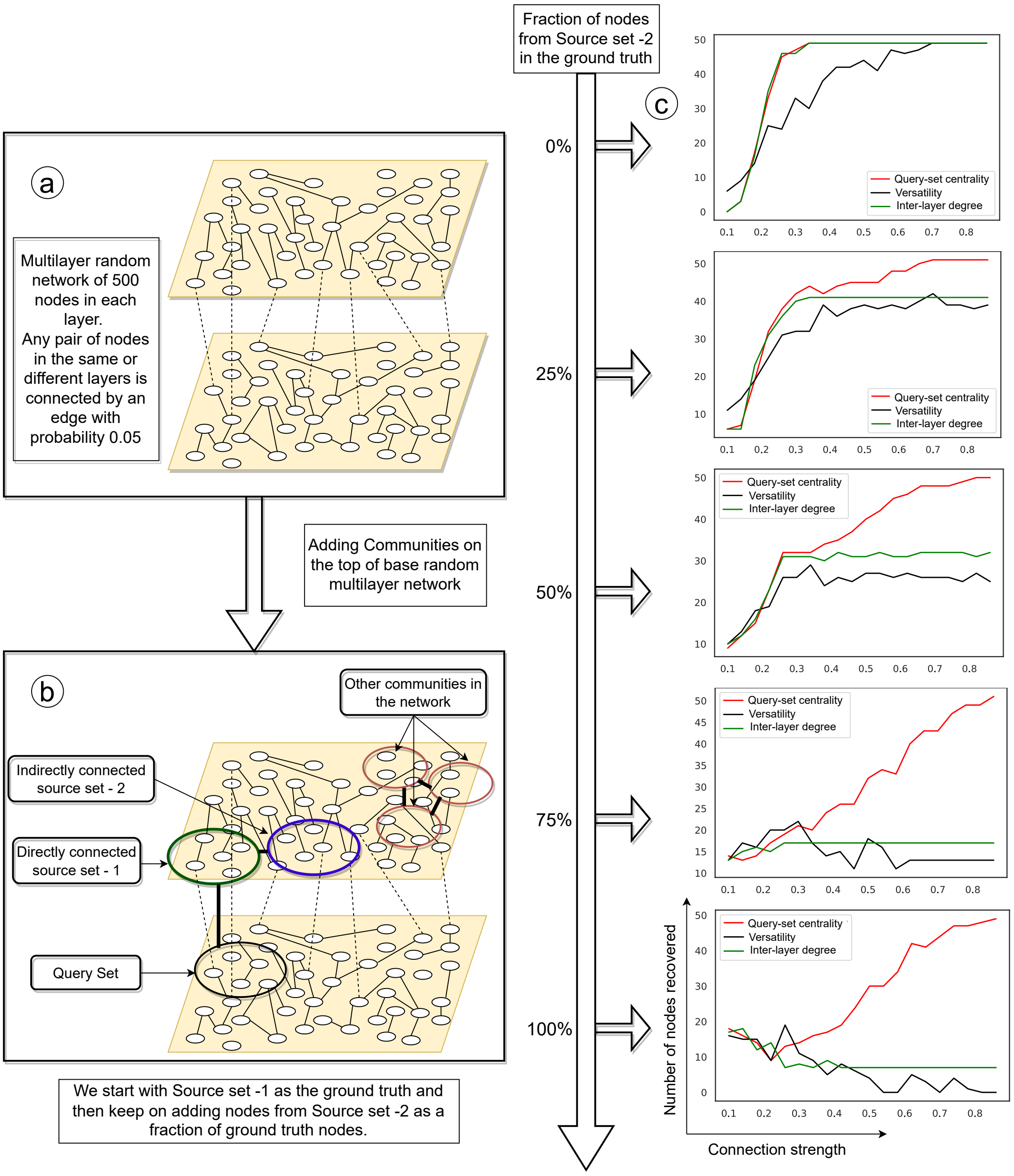

a
Multilayer random network of 500 nodes in each layer.
Any pair of nodes in the same or different layers is connected by an edge with probability 0.05
Adding Communities on the top of base random multilayer network
b
Other communities in the network
Indirectly connected source set - 2
Directly connected source set - 1
Query Set
We start with Source set -1 as the ground truth and then keep on adding nodes from Source set -2 as a fraction of ground truth nodes.
Fraction of nodes from Source set -2 in the ground truth
0%
25%
50%
75%
100%
c
Query-set centrality
Versatility
Inter-layer degree
Number of nodes recovered
Connection strength
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
10
20
30
40
50

Figure 5.1: **Synthetic multilayer network construction and evaluation.** (a) Synthetic network construction starts with a base random multilayer network with edge probability 0.05; (b) More edges are then added, according to the connection strength desired, both within the selected communities (indicated by circles) and between certain pairs of communities (indicated by thick dark edges connecting the pair; e.g. between *source-set 1* and *source-set 2*). (c) As more nodes from *source-set-2* become part of the ground truth (shown as increasing percentages), our MultiCens query-set centrality outperforms the existing methods to a larger extent. Each plot shows the connection strength (x-axis) against the number of ground truth nodes in the top 100 ranked nodes (y-axis).

directly connected (like *source-set-1*) to the query-set, and nodes connected by long hops (similar to *source-set-2*) make the ground truth; In such scenarios, the proposed method shows better results in these plots. Now we evaluate the proposed centrality measure on the real-world multilayer networks in the next section.

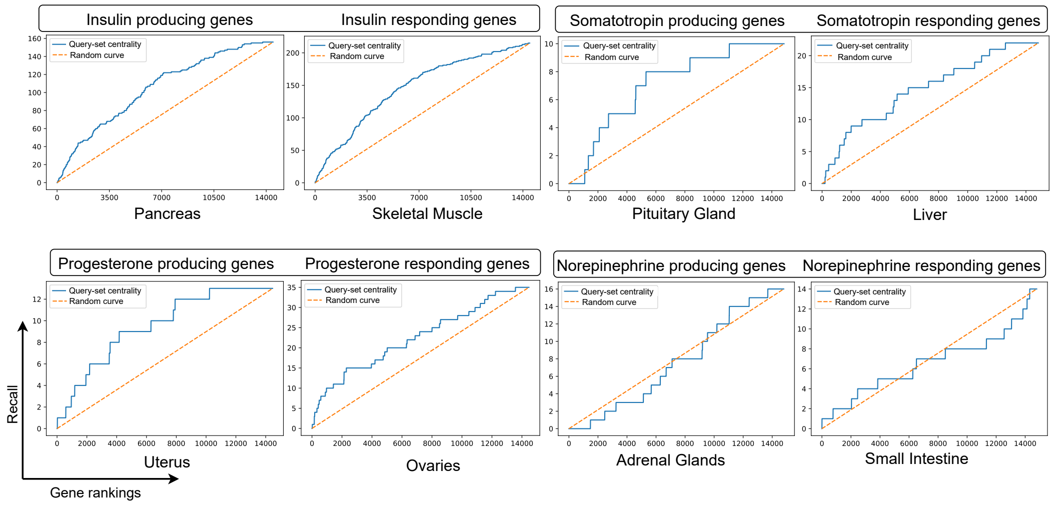
5.3.2 Results with Human Multi-tissue Data

MultiCens ranks the inter-tissue communication driver genes at the top.

We work with human multi-tissue datasets and find genes that regulate inter-tissue communication. We use the HGv1 dataset to obtain hormone-producing and responding genes. In the HGv1 dataset, there is a vast imbalance in the known hormone-gene associations between the well-studied hormones such as Insulin (156 producing genes) and Leptin (1 producing gene). We restrict our experiments to only those hormones with at least 10 genes in either the hormone-producing or responding sets. Further, we report only those hormones with at least 10 genes on both sides to perform the ground-truth-based evaluation. Four hormones viz. Insulin, Somatotropin, Progesterone, and Norepinephrine cross this threshold. We use either hormone-producing or responding gene set as the query-set and treat the other set as the ground truth to recover.

Figure 5.2(a) shows the *recall-at-k* plots for the hormones where we have more than 10 genes in both hormone-producing as well as responding sets. We compare the recall curve with a curve that stands for the recall had the ranking been random. It can be seen that the proposed centrality measure outperforms in three out of four hormones showing the promising potential in identifying the drivers for tissue-tissue

(a) *Recall-at-k* plots for primary hormones with at least ten genes in both hormone-producing and responding gene sets



(b) Area under *recall-at-k* plots for hormones with at least ten genes in hormone-producing, responding or both gene sets

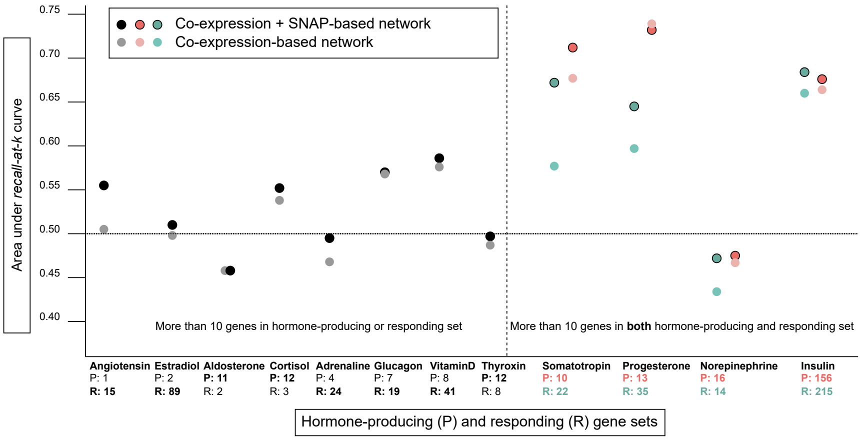


Figure 5.2: **MultiCens on human multilayer networks: ground-truth validation.** (a) Recall (# of ground truth genes recovered; y-axis) in the top k ranked genes (x-axis) are plotted using MultiCens query-set centrality based ranking *vis-à-vis* a random ranking (random curve). (b) For hormones with 10 or more genes in either producing or responding set, the smaller set is used as the query-set, and the plot reports AUC score for predicting the bigger set (marked in bold-face font in x-axis). For the four primary hormones having at least 10 genes on both producing and responding sets, plot reports AUC for predicting both sets.

communication. Figure 5.2 (b) shows the performance of the proposed centrality measure on the hormones with at least 10 genes in either hormone-producing or responding sets. The upper plot is obtained from a network constructed using both

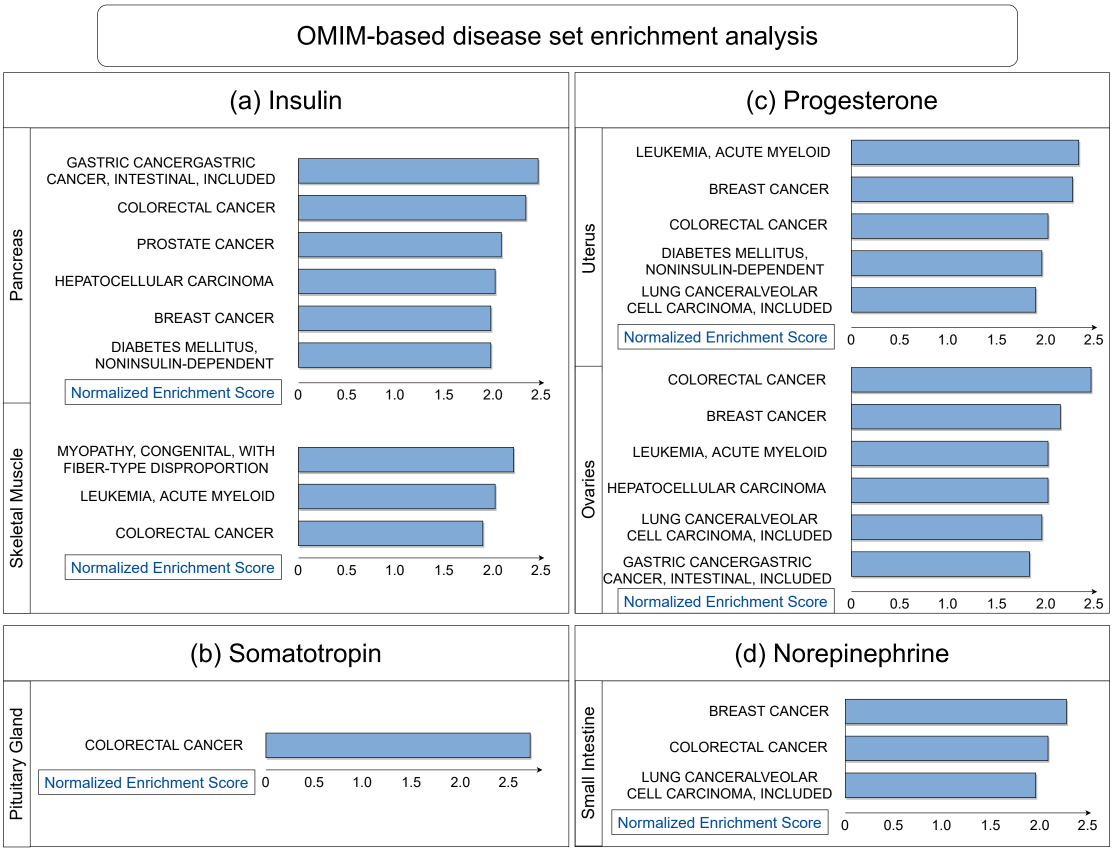


Figure 5.3: OMIM-based disease set enrichment analysis of the centrality scores. We use WebGestalt to get these enrichments and apply an FDR cut-off of 0.05. For Somatotropin and Norepinephrine, we do not see any enrichment crossing the FDR cut-off in Liver and Adrenal Glands, respectively.

coexpression-based edges and the SNAP repository, where the bottom plot is obtained using only coexpression-based edges.

In hormones, where we do not have at least 10 genes in either set, we report the performance of predicting the larger set using the smaller set as the *query-set* of genes. Figure 5.2 (b) shows a trend of increasing performance as the number of genes in the ground truth increase. It also shows the robustness of the method, as using only coexpression-based edges results in a small drop in the performance of some of the hormones.

Our validation using the ground truth hormone-gene associations from the HGv1 dataset affirms the potential of the proposed centrality framework to capture the inter-tissue communication driver genes. The encouraging performance of the proposed centrality measure led us to perform disease-based enrichment analysis (DSEA) of the centrality-based gene rankings. We use the WebGestalt platform to run DSEA on the genes along with their centrality scores. Figure 5.3 presents the enriched disease terms for each hormone at 0.05 FDR cut-off. We apply a stringent FDR cut-off to minimize the chances of encountering false positives; hence, the gene rankings corresponding to two hormones, Somatotropin and Norepinephrine, do not show any enrichment on the hormone-responding and producing sides, respectively.

Centrality-based Gene Rankings are Enriched for Hormone Related Diseases

Among the enriched disease terms, many of them are well-supported in the literature. A recent study on understanding the carcinogenic factors for the development of gastric cancer revealed the role of insulin resistance Kwon *et al.* (2019). Insulin resistance has also been found to be a potential cause for the Colorectal cancer Schoen *et al.* (1999), and Prostate cancer Hsing *et al.* (2003). The relationship between Insulin and Diabetes Type-2 is well studied in the literature Reaven (1980), as found by our enrichment analysis. Studies have shown that patients with congenital fiber-type disproportion myopathy on muscles develop insulin-resistance diabetes Vestergaard *et al.* (1995), confirming the term enrichment in our results. Further, Leukemia is known to undercut the capacity of healthy cells to consume glucose Ye *et al.* (2018). The role of (somatotropin) growth hormone-related genes is known for the development of colorectal cancer Yang *et al.* (2004), and our enrichment analysis confirms this association. Progesterone hormone plays a crucial role in the development of breast cancer Trabert *et al.* (2020), and our findings show this term enriched for the gene rankings in both hormone-producing as well as hormone-responding tissues. Norepinephrine is known to be an etiological factor in developing several types of cancers, including the ones we found in the enrichment analysis Fitzgerald (2009). However, the set of primarily associated diseases to Norepinephrine are not observed in our analysis, which is in line with the poor performance of this hormone-related gene ranking in other analyses as well.

Our Top Predicted Genes are Supported by The PubMed-based Literature Analysis.

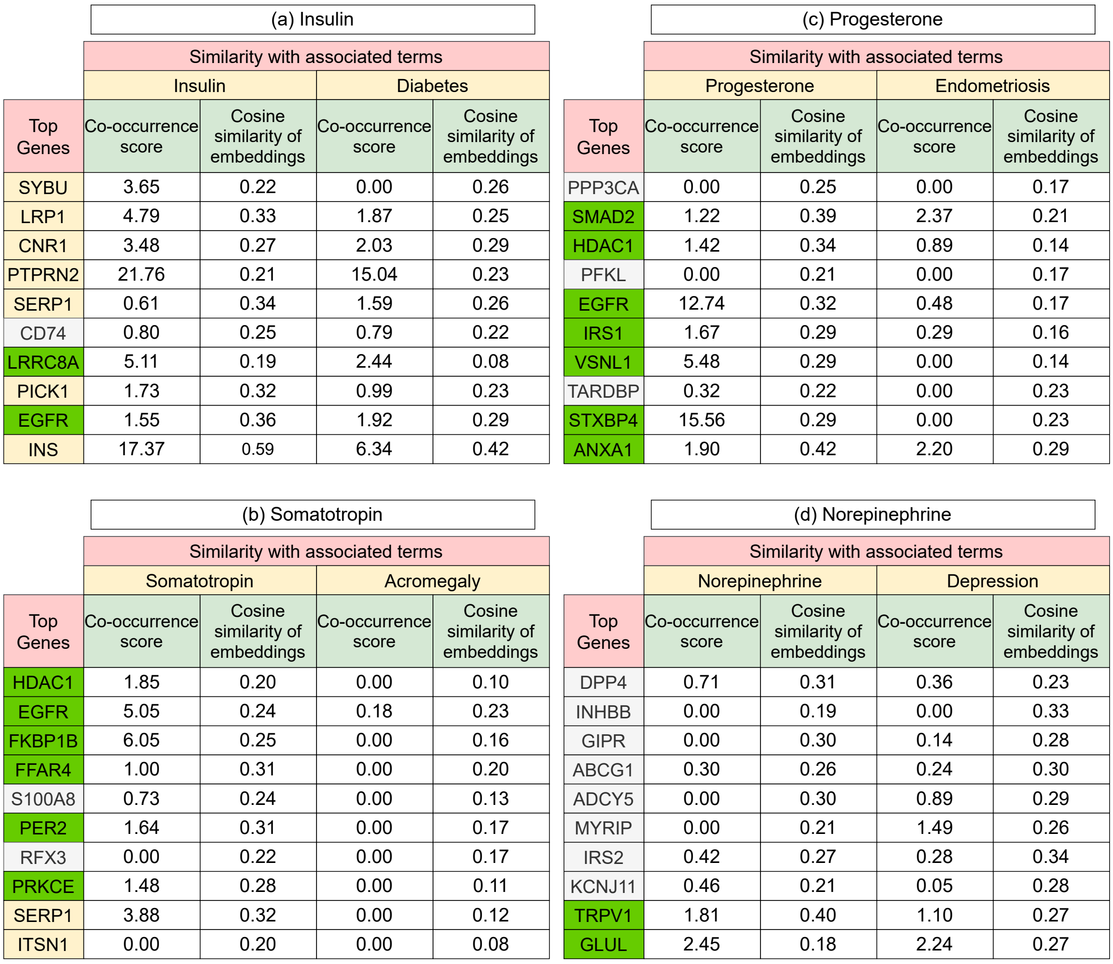
For the out-of-ground-truth analysis, we seek the validation of predicted hormone-gene associations from the PubMed database. In PubMed articles, a hormone-gene pair can co-occur and imply a potential relationship between them. We use a scoring mechanism, co-occurrence score, to quantify this evidence, as defined in section 5.2.3. In the PubMed database, some articles may involve the study of similar hormones, hormones with their associated diseases, diseases with the driver genes, etc. Such articles may not include the potential hormone-gene pairs in the text, but these hormones and genes tend to have a similar context. Text-based embedding methods have shown great success in capturing these similarities into word vectors Habibi *et al.* (2017). We investigate the cosine similarity between the word vectors corresponding to gene and the hormone-related terms to validate our predictions. In this analysis, we only test the predictions among top protein-coding[9] genes. Figure 5.4 shows the top ten predicted genes for each hormone along with their co-occurrence and embedding-based cosine similarity scores with the hormone-related terms. Genes with a green background are not present in the ground-truth HGv1 dataset of hormone-gene associations, but our centrality-based rankings find their connection which is confirmed by the high similarity scores with at least one of the hormone-related terms. Genes with a grey background are not present in the ground truth, and none of the hormone-related terms show a supportive score for both columns. Genes with a yellow background are present in the HGv1 database and ranked top in our centrality-based ranking.

In Figure 5.4, it can be seen that all hormones show multiple entries with a green background, confirming the identification of novel drivers of tissue-tissue communication. Recently, *LRRC8* has been found to enhance insulin secretion in pancreatic β-cells Stuhlmann *et al.* (2018). Later studies also confirm the role of *LRRC8* in insulin resistance and glucose resistance Kumar *et al.* (2020). Similarly, *EGFR* gene is known to have an association with diabetes disease Group *et al.* (2015). The role of *CD74* in Insulin secretion and related diseases was not well-established until the recent discovery of its participation in insulin resistance Chan *et al.* (2018). Based on the high centrality ranking and recent pieces of evidence, *CD74* warrants further exploration and can be prioritized in future experiments.

In the case of somatotropin, *S100A8* gene is not present in the ground truth HGv1 database, and it receives little support from the scores. Literature shows downregulation

---

[9] List of protein-coding genes accessed from this URL- www.ebi.ac.uk/QuickGO/GTerm?id=GO:0002790 on Dec 1, 2020, at 2100 hours IST



(a) Insulin

	Similarity with associated terms			
	Insulin		Diabetes	
Top Genes	Co-occurrence score	Cosine similarity of embeddings	Co-occurrence score	Cosine similarity of embeddings
SYBU	3.65	0.22	0.00	0.26
LRP1	4.79	0.33	1.87	0.25
CNR1	3.48	0.27	2.03	0.29
PTPRN2	21.76	0.21	15.04	0.23
SERP1	0.61	0.34	1.59	0.26
CD74	0.80	0.25	0.79	0.22
LRRC8A	5.11	0.19	2.44	0.08
PICK1	1.73	0.32	0.99	0.23
EGFR	1.55	0.36	1.92	0.29
INS	17.37	0.59	6.34	0.42

(c) Progesterone

	Similarity with associated terms			
	Progesterone		Endometriosis	
Top Genes	Co-occurrence score	Cosine similarity of embeddings	Co-occurrence score	Cosine similarity of embeddings
PPP3CA	0.00	0.25	0.00	0.17
SMAD2	1.22	0.39	2.37	0.21
HDAC1	1.42	0.34	0.89	0.14
PFKL	0.00	0.21	0.00	0.17
EGFR	12.74	0.32	0.48	0.17
IRS1	1.67	0.29	0.29	0.16
VSNL1	5.48	0.29	0.00	0.14
TARDBP	0.32	0.22	0.00	0.23
STXBP4	15.56	0.29	0.00	0.23
ANXA1	1.90	0.42	2.20	0.29

(b) Somatotropin

	Similarity with associated terms			
	Somatotropin		Acromegaly	
Top Genes	Co-occurrence score	Cosine similarity of embeddings	Co-occurrence score	Cosine similarity of embeddings
HDAC1	1.85	0.20	0.00	0.10
EGFR	5.05	0.24	0.18	0.23
FKBP1B	6.05	0.25	0.00	0.16
FFAR4	1.00	0.31	0.00	0.20
S100A8	0.73	0.24	0.00	0.13
PER2	1.64	0.31	0.00	0.17
RFX3	0.00	0.22	0.00	0.17
PRKCE	1.48	0.28	0.00	0.11
SERP1	3.88	0.32	0.00	0.12
ITSN1	0.00	0.20	0.00	0.08

(d) Norepinephrine

	Similarity with associated terms			
	Norepinephrine		Depression	
Top Genes	Co-occurrence score	Cosine similarity of embeddings	Co-occurrence score	Cosine similarity of embeddings
DPP4	0.71	0.31	0.36	0.23
INHBB	0.00	0.19	0.00	0.33
GIPR	0.00	0.30	0.14	0.28
ABCG1	0.30	0.26	0.24	0.30
ADCY5	0.00	0.30	0.89	0.29
MYRIP	0.00	0.21	1.49	0.26
IRS2	0.42	0.27	0.28	0.34
KCNJ11	0.46	0.21	0.05	0.28
TRPV1	1.81	0.40	1.10	0.27
GLUL	2.45	0.18	2.24	0.27

Figure 5.4: Literature support for the gene rankings obtained using the proposed centrality score. This Figure presents the top 10 predicted genes (ranked only among secretory genes) for each hormone, along with their co-occurrence scores and similarity in embedding space with hormone-related terms. Genes with a yellow background are present in the ground truth (HGv1 data); from the remaining genes, the green background represents genes supported by scores from either or both hormone-related terms, and genes with a grey background are not supported by any of the hormone-related term using both scores.

in *S100A8* on exogenous administration of growth hormone: such studies and a high ranking as per our centrality method command further exploration. Similarly, *RFX3* gene is known to play a role in hydrocephalus disease Baas *et al.* (2006), which is associated with the deficiency in growth hormone Wen *et al.* (2010). This context-based dependency is well captured in the cosine-based similarity in the embeddings pace, but the gene has no direct co-occurrence with hormone-related terms.

In Progesterone, three out of the top ten predicted genes are not present in the ground truth. These genes get nominal support from the literature indicating the requirement of further exploration of these associations. Studies show an increase in postprandial Norepinephrine levels by inhibiting the *DDP4* gene. The first rank obtained by *DDP4* gene demands further exploration on these lines.

5.3.3 Centrality of random node sets to assess statistical significance

In synthetic benchmarks or hormone-gene prediction applications discussed above, we typically compare the performance of the ranking given by a particular centrality measure to random rankings of all nodes in the network that need to be ranked.

Specifically, a ranking-based evaluation metric of a set of nodes of interest S (e.g., recall-at-k of a ground-truth set of genes) computed from the actual centrality-based ranking vs. random rankings are then compared to assess the statistical significance of the centrality scores of node(s) in S. This procedure is equivalent to comparing the centrality scores of S to that of a random set of nodes whose size matches the size of S.

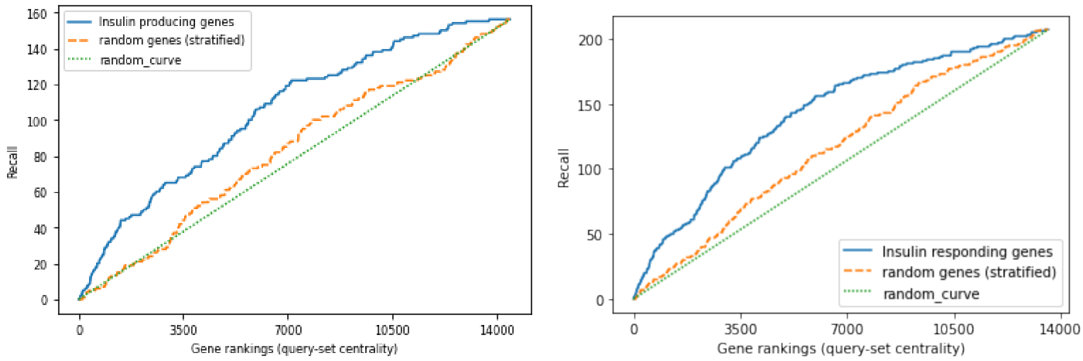


Figure 5.5: Comparison of gene rankings of ground truth sets (hormone-producing and responding) with a random gene set chosen by stratification on the gene variance level. Blue curve corresponds to the actual ground truth gene set, orange curve represents the ranking obtained by a random set of genes chosen by stratifying over gene variance levels, and green curve is the average curve obtained using any random set of genes. Note that the random gene sets we consider have the same size as the actual ground truth set.

To refine the above procedure, we can also have the random set match other properties of S, such as the expression values or variances of the genes in S across all the samples in the input dataset. More specifically, we can stratify genes into three classes of genes: ones with low, medium and high variance across all input samples. We use closed intervals of 0-33, 33-66, and 66-100 percentile-based cut-offs to classify the genes into low, medium, and high varying categories, respectively. A random gene set is now chosen such that the number of genes in each of these three classes matches the corresponding number of genes in S. We have performed this refinement for insulin-gene predictions, for instance, and show that (see Figure 5.5) the ground-truth producing or responding gene set of insulin to be predicted has better centrality than matched random sets of genes.

5.3.4 Novel Findings: lncRNAs and their Support from Literature

Our multi-tissue datasets include non-secretory genes such as lncRNAs. For a long time, these lncRNAs were not known to be participating in hormonal signaling until recent studies have shown their association with human physiology and diseases (Sun and Kraus (2015)). Our centrality results also return a ranking among lncRNAs for each hormone. We validate these findings by searching through Google Scholar to find articles that contain these lncRNAs and the hormone or disease names associated with them.

Rank	lncRNA symbol	References
1	LINC00672	Li *et al.* (2017, 2019*a*)
2	HOXA-AS2	Lian *et al.* (2017); Li and Yu (2020)
3	PRR34-AS1	Liu *et al.* (2019)
4	MIR22HG	Lin *et al.* (2017)
5	LINC00294	None

Table 5.1: Top predicted lncRNAs for Insulin in Pancreas along with their references in the literature. The references mentioned in the table include these lncRNAs with terms Insulin or Diabetes.

Rank	lncRNA symbol	cReferences
1	ZEB1-AS1	Gu *et al.* (2020); Meng *et al.* (2020); Song *et al.* (2019)
2	TNK2-AS1	None
3	PWAR6	Liu *et al.* (2019); Nagai and Mori (1999); Basheer *et al.* (2016)
4	PRRT3-AS1	Yang *et al.* (2021)
5	PRKCQ-AS1	Timmons *et al.* (2018)

Table 5.2: Top predicted lncRNAs for Insulin in Skeletal Muscle along with their references in the literature. The references mentioned in the table include these lncRNAs with terms Insulin or Diabetes.

Rank	lncRNA symbol	References
1	LINC01588	Volejnikova *et al.* (2020)
2	PTPRD-AS1	None
3	LINC01132	None
4	UCA1	None
5	LINC01473	None

Table 5.3: Top predicted lncRNAs for Somatotropin in Pituitary Gland along with the references mentioning terms Somatotropin (growth hormone) or Acromegaly.

Rank	lncRNA symbol	References
1	NEAT1	None
2	ZNF528-AS1	Rothzerg *et al.* (2021)
3	MIR210HG	Sun *et al.* (2021)
4	ALMS1-IT1	None
5	LINC01278	None

Table 5.4: Top predicted lncRNAs for Somatotropin in Liver along with the references mentioning terms Somatotropin (growth hormone) or Acromegaly.

Rank	lncRNA symbol	References
1	HAGLR	Jin *et al.* (2021); Tang *et al.* (2019)
2	TAF1A-AS1	None
3	LINC00602	None
4	PCAT19	Lange (2021)
5	HHIP-AS1	Li and Zhan (2019)

Table 5.5: Top predicted lncRNAs for Progesterone in Uterus along with the references mentioning terms Progesterone, Breast cancer or Endometriosis.

Rank	lncRNA symbol	References
1	CCDC18-AS1	None
2	LINC00641	Dastmalchi *et al.* (2021)
3	MIR210HG	Li *et al.* (2019*b*); Du *et al.* (2020)
4	LINC01016	Li *et al.* (2021); Pan *et al.* (2018)
5	BEAN1-AS1	None

Table 5.6: Top predicted lncRNAs for Progesterone in Ovaries along with their references in the literature. The references mentioned in the table include these lncRNAs with terms Progesterone, Breast cancer or Endometriosis.

Rank	lncRNA symbol	References
1	PGM5P4-AS1	None
2	CCDC18-AS1	None
3	MAGI2-AS3	Ghosal *et al.* (2021)
4	LINC01291	None
5	TOLLIP-AS1	None

Table 5.7: Top predicted lncRNAs for Norepinephrine in Adrenal Glands along with their references in the literature. The references mentioned in the table include these lncRNAs with terms Norepinephrine or Depression.

Rank	lncRNA symbol	References
1	RNF139-AS1	None
2	CARMN	None
3	SPATA41	Zhang *et al.* (2020)
4	GHET1	None
5	ATP1B3-AS1	None

Table 5.8: Top predicted lncRNAs for Norepinephrine in Small Intestine along with their references in the literature. The references mentioned in the table include these lncRNAs with terms Norepinephrine or Depression.

5.3.5 MultiCens detects Changes in Brain Networks between Alzheimer Disease and Control Populations

After recognizing the potential of MultiCens in identifying genes (both protein coding and lncRNAs) in hormone signaling pathways in health, we employ it to understand the change in the gene-gene network structures in disease, specifically Alzheimer's disease (AD) relative to a control (CTL) population. We retrieved data of 264 AD and 372 control human postmortem RNAseq samples from Mount Sinai Brain Bank dataset Wang *et al.* (2018*a*) for four brain regions: frontal pole (FP), superior temporal gyrus (STG), parahippocampal gyrus (PHG), and inferior frontal gyrus (IFG). We construct one multilayer network for the AD group of individuals and another for the

CTL group, with four layers in the network representing the four brain regions, and network nodes and edges representing respectively the genes in these brain regions and gene-gene coexpression relations (after adjusting for covariates). We use the genes involved in synaptic signaling (SSG) in the PHG region as the query set of genes (134 genes), and identify the disease-driven change in the centrality-based ranking of genes in the remaining three regions. We observed considerable shift in the ordering of these three brain regions in the AD vs. CTL multilayer networks according to their median gene centrality scores (see Fig. 5.6a, STG region's ordering for instance). In terms of individual genes, *ANKFN1*, *OR10AD1* and *PLCD3* gain the highest positive shift in AD-based ranking in the FP, STG and IFG regions respectively. *ANKFN1* is found to be upregulated in hippocampus tissues of AD patients Yan *et al.* (2019). Though *OR10AD1* (olfactory receptor family 10 subfamily AD member 1) is not yet connected to AD, olfactory impairments is recently reported to be one of the early phase' pathophysiological changes in AD Alves *et al.* (2014). *PLCD3* is known to be upregulated in the AD population along with other regulators of lipid metabolism Zhang *et al.* (2018*b*).

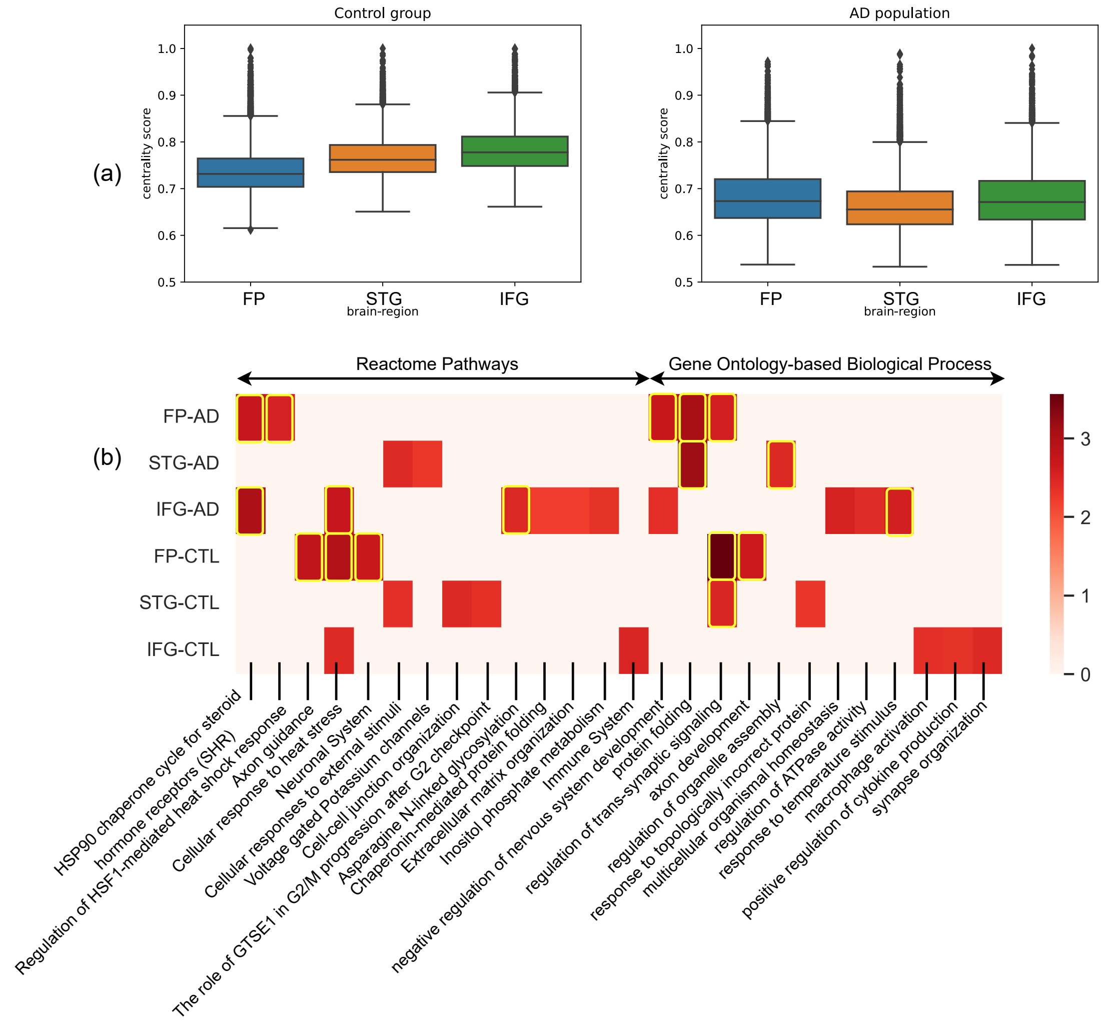
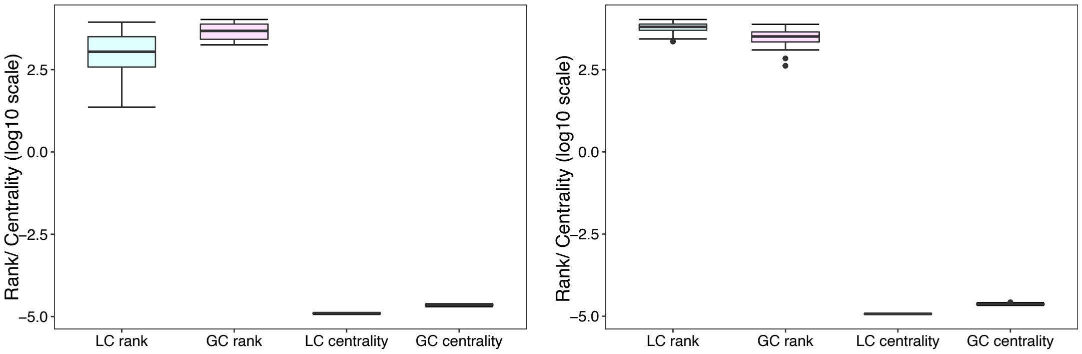


Figure 5.6: **MultiCens on multi-brain-region networks in disease:** Study of changes in MultiCens gene rankings of four-layer networks of control and Alzheimer affected population. We rank genes of frontal pole (FP), superior temporal gyrus (STG) and inferior frontal gyrus (IFG) using MultiCens centralities calculated using a query-set of synaptic signaling genes in parahippocampal gyrus (PHG). (a) Bar-plot showing region-wise shift of centrality scores of the three regions. (b) Reactome pathways and Gene Ontology-based process (GO-BP) enrichment analysis of each region in control and AD state. Color map represents the normalized enrichment score from WebGestalt. The highlighted boxes pass the 0.01 FDR cut-off. If centrality-based gene rankings of a region do not pass the 0.05 FDR cut off for an enrichment, we set the corresponding normalized enrichment score to 0.

MultiCens also offers an across-region view of gene importance in the AD or CTL multilayer networks. In the AD network, irrespective of brain regions, genes *JMJD6*, *SLC5A3*, *CIRBP*, *TARBP1* and *AHSA1* are among the top ten central genes correlated with the SSG set, of which *AHSA1* (activator of HSP90 ATPase activity 1) is already known to correlated with AD progression by promoting tau fibril formation Shelton *et al.* (2017). On the other hand, *CIRBP* (cold inducible RNA binding protein) shields neurons from amyloid toxicity mediated by antioxidative and antiapoptotic pathways, making it a favourable molecule contending for AD prevention or therapy Su *et al.* (2020). It may be worth studying the other three genes experimentally to test their connections to AD pathology. Similar to these individual genes, certain biological pathways were also enriched for top ranks, irrespective of the brain region, in the AD network (see Fig. 5.6b) – examples include HSP90 chaperone cycle for steroid hormone receptors (R-HSA-3371497) pathway and negative regulation of nervous system development (GO:0051961). Heat shock protein 90 (Hsp90), "a molecular chaperone", is known to induce microglial activation leading to amyloid-beta (Aβ) clearance Ou *et al.* (2014). The across-region consistency of top-ranking genes/pathways in the AD network is not observed in the CTL multilayer network. For example, gene *CDK5R2* (Cyclin Dependent Kinase 5 Regulatory Subunit 2) is ranked 3rd in FP, rank 224 in STG, and 2076 in IFG. Pathway enrichments are also more region-specific in the CTL network (relative to AD network; see Fig. 5.6b), such as Axon guidance in FP, Cell-cell junction organization in STG, and immune system in IFG. The intricate links between immune system and neuronal signaling is well-appreciated. Other enrichments that serve as a positive control to increase confidence in our MultiCens rankings are those of biological processes like ‘regulation of trans-synaptic signaling’ in FP and STG, and ‘synapse organization’ in IFG.

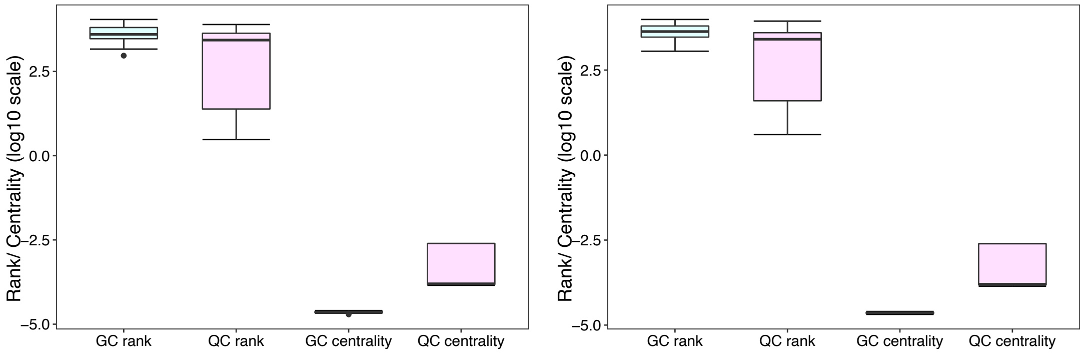


(a) RNA splicing (b) Acute inflammatory response

Figure 5.7: Different ranks and centrality scores of the genes (y-axis in log-scale), participating in enriched biological process resulting from LC-GC delta rank, are highlighted in the box plots. While RNA splicing seems to be predominant (influential) in the intra-region gene network, Acute inflammatory response seems to be influential in the inter-region gene network due to its better global centrality ranks.

Along with MultiCens query-set centrality (QC), we have further computed and analyzed (e.g., using WebGestalt) local centrality (LC) and global centrality (GC) measures of MultiCens. To highlight the difference among these three centrality measures, we also computed and analyzed “delta" rankings (i.e., differences in two rankings: “LC - GC", and “GC - QC"). Fig 5.7 reveals important biological insights from the different centrality measures - while better ranked genes in LC are enriched for RNA splicing, those in GC are enriched for acute inflammatory response. Further, the distribution of LC and GC ranks for the above mentioned GO-BPs show that while some genes have an active role to play within brain regions, other genes are influential in inter-brain-region connectivity. We observed a similar trend when inspecting GC-

QC delta ranks as shown in Fig. 5.8. Taken together, having multiple centrality values within our MultiCens framework is advantageous in bringing out different facets of the AD disease network.



(a) Neuronal System (b) Regulation of trans-synaptic signaling

Figure 5.8: Different ranks and centrality scores of the genes (y-axis in log-scale), participating in enriched pathways resulting from GC-QC delta rank, are highlighted in the box plots. Both neuronal system pathway and trans-synaptic signaling regulation are better connected to the query-set (synaptic signaling genes) as expected.

Finally, to find out whether changes in AD-network is specific to the query pathway or similar across pathways, we further use plaque-induced genes (PIGs, total 57 genes), prominent in the later phase of AD, as query-set in PHG instead of the SSG set and repeat the same analysis with MultiCens. We found predominant similarities as well as certain interesting differences in centrality ranks between the two query gene sets. While pathways related to heat stress was common for both query sets, synaptic signalling related process like “cell-cell junction organization” was prominent for SSG set and interleukin signaling was exclusively noted for PIG set (see Fig. 5.9).

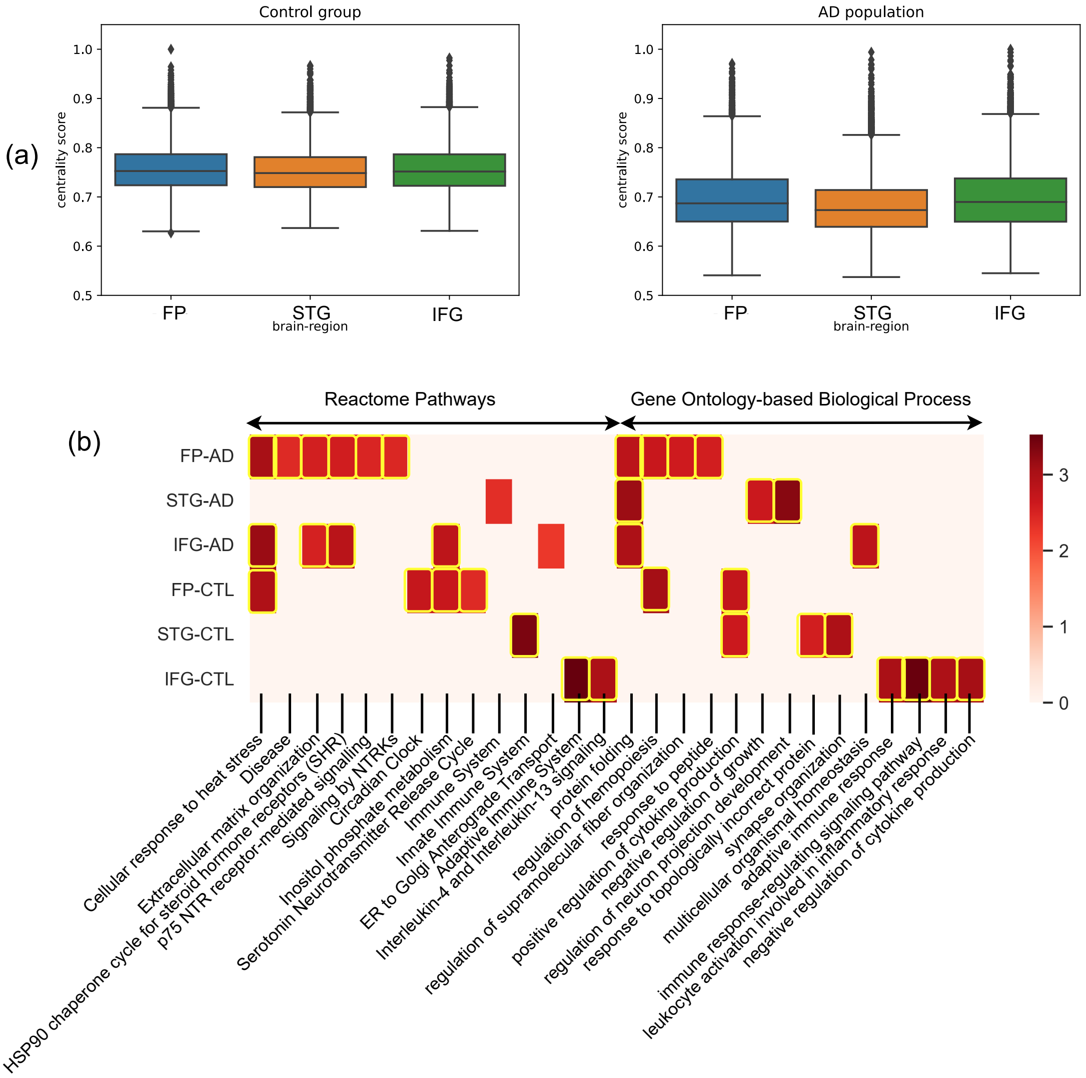


Figure 5.9: **MultiCens on multi-brain-region networks in disease (PIG-based query-set)**: Study of changes in the centrality-based gene rankings of four-layer networks of control and Alzheimer affected population. The PIG *query-set* is present in parahippocampal gyrus (PHG) and we rank genes of frontal pole (FP), superior temporal gyrus (STG) and inferior frontal gyrus (IFG). (a) Bar-plot showing region-wise shift of centrality scores of the three regions. (b) Reactome pathways and Gene Ontology-based process (GO-BP) enrichment analysis of each region in control and AD state. Color map represents the normalized enrichment score from WebGestalt. The highlighted boxes pass the 0.01 FDR cut-off. If centrality-based gene rankings of a region do not pass the 0.05 FDR cut off for an enrichment, we set the corresponding normalized enrichment score to 0.

In aggregate, these results on alterations of brain networks in Alzheimer's disease using different query sets show how MultiCens can provide a network-centric perspective and related hypotheses for prioritizing experimental investigations of disease mechanisms.

5.4 SUMMARY

We survey the centrality methods for multilayer networks and emphasize the need for novel methods to leverage the layered configuration of the underlying system to capture the centrality effects on local vs. global levels and to a particular layer and a query set of nodes in a specific layer. We propose MultiCens, a collection of centrality measures to capture the effect of nodes at different granularities of the multilayer structure. The proposed centrality measures have theoretical properties such as convergence guarantees and decomposability. We validate our proposed methods on an extensive set of synthetic multilayer networks and derive the conditions where the proposed methods outperform the existing methods. We experiment with multilayer networks arising from human multi-tissue datasets and identify genes regulating inter-tissue hormonal communication. Our detailed analysis showed the superior performance of the proposed method on multiple validation methods, both with and without using the ground truth data. Our gene rankings are enriched for processes and diseases related to the hormones, which assures us to consider this method for prioritizing genes in future experiments. This work also opens paths to using the proposed centrality measures to identify differential components in healthy and diseased populations. We extend our experiments to understand the shift in gene rankings in brain regions of the AD population than the control group and draw important insights.

Part III

Hypergraphs: Theoretical insights and implications for effective clustering and hyperedge prediction

CHAPTER 6

Hypergraph: Methods for Clustering and Hyperedge Prediction

Many real-world systems involve components interacting at a super-dyadic level. The representational power of pairwise graph models is insufficient to capture higher-order information and present it for analysis or learning tasks. These systems can be more precisely modeled using *hypergraphs* where nodes represent the interacting components, and hyperedges capture higher-order interactions (Bretto (2013); Klamt *et al.* (2009); Satchidanand *et al.* (2014); Lung *et al.* (2018)). A hyperedge can capture a multi-way relation; for example, in a co-authorship network, where nodes represent authors, a hyperedge could represent a group of authors who collaborated for a common paper. If this were modeled as a graph, we would be able to see which two authors are collaborating, but would not see if multiple authors worked on the same paper. This suggests that the hypergraph representation is not only more information-rich but is also conducive to higher-order learning tasks by virtue of its structure. Indeed, there is a recently expanding interest in research in learning on hypergraphs (Zhang *et al.* (2018*a*); Zhao *et al.* (2018); Saito *et al.* (2018); Feng *et al.* (2018); Chodrow and Mellor (2020)). In this work, we are interested in devising solutions for the problem of *Hypergraph clustering* and *Hyperedge prediction*. Analogous to the graph clustering task, *Hypergraph clustering* seeks to discover densely connected components within a hypergraph (Schaeffer (2007)). This has been the subject of several research works by various communities with applications to various problems such as VLSI placement (Karypis and Kumar (1998)), discovering research groups(Kamiński *et al.* (2019)), image segmentation (Kim *et al.* (2011)), de-clustering for parallel databases (Liu and

Wu (2001)) and modeling eco-biological systems (Estrada and Rodriguez-Velazquez (2005)), among others.

Hyperedge prediction is a less explored but extremely important problem of hypergraphs. Unlike edge prediction, hyperedge prediction has several bottlenecks, both semantically and computationally, making the problem more challenging. The inherent complexity of hypergraphs hinders edge-prediction methods from directly predicting hyperedges. Unlike graphs, where an edge can connect only two nodes, a hyperedge can connect an arbitrary number of nodes. Thus, while in a graph, the maximum possible number of edges is $\mathcal{O}(n^2)$, in a hypergraph, the maximum possible number of hyperedges is $\mathcal{O}(2^n)$. Searching through this enormous space for potential hyperedges exacerbates the modeling and search challenge as compared to the traditional edge or link prediction. In this chapter, we present the theoretical framework required to build solutions for hypergraph clustering and hyperedge prediction. The experimental details, results and its analysis are presented in the next chapter. We begin by introducing modularity in hypergraphs, and follow up by presenting an iterative algorithm for hypergraph clustering. In the end we discuss the scoring strategies required to predict new hyperedges in a given hypergraph.

6.1 HYPERGRAPH MODULARITY

One possible way to define hypergraph modularity is to introduce a hypergraph null model and utilize it to define a modularity function. Kaminski et al. (Kamiński *et al.* (2019)) follow this approach and use a generalized version of the Chung-Lu model (Chung and Lu (2002)) to define hypergraph modularity. The proposed modularity function only counts the participation of hyperedges entirely contained inside a cluster. Moreover, the modularity function requires separate processing of hypergraphs induced by hyperedges with different cardinalities. Though such assumptions can provide the analytic tractability of the solution, they limit its applicability to real-world hypergraphs where the hypergraphs can be of very large size with varying hyperedge cardinalities.

Another possible way to define hypergraph modularity is to convert the hypergraph to an appropriate graph and then define modularity on the resultant graph. Such an approach can get benefits from the already existing tools for graphs. In this section, we will follow the latter approach to introduce hypergraph modularity.

To begin, we represent a hypergraph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes (or vertices) and $E = e_1, e_2, \ldots, e_m$ is the set of m hyperedges, by an incidence matrix H as follows,

$$\mathbf{H}(v, e) = \begin{cases} 1 & \text{if } v \in e \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

The presence of 1 in the incidence matrix represents the participation of the corresponding node in that particular hyperedge. Degree of node v is defined as $d(v) = \sum_{e \in E, v \in e} w(e)$ where $w(e)$ represents the weight of the hyperedge e, and $N(v)$ is a set containing the one-hop neighbors of node v (nodes of hyperedges, v is part of). For a hyperedge e, its degree is defined as $\delta(e) = |e|$. $D_v \in \mathbb{R}^{n \times n}$, $D_e \in \mathbb{R}^{m \times m}$ and $W \in \mathbb{R}^{m \times m}$ are the diagonal matrices containing node degrees, hyperedge degrees and hyperedge weights at the diagonals and *zero* otherwise, respectively.

To introduce the hypergraph modularity, we start by proposing a null model on the graphs generated by reducing hypergraphs. In a reduced graph, we desire the nodes to possess the same degree as that of the original hypergraph. In a thus reduced graph, the expected number of edges connecting nodes i and j can be given as

$$P_{ij}^{hyp} = \frac{d(i) \times d(j)}{\sum_{v \in V} d(v)} \tag{6.2}$$

Where $d(i)$ and $d(j)$ represents the node degrees of nodes i and j respectively. The proposed null model can be interpreted as a mechanism to generate random graphs where the node degree sequence of a given hypergraph is preserved irrespective of the count and cardinality of hyperedges. In order to define a modularity matrix, we need to obtain a graph reduction where the node degree sequence should remain preserved. One straightforward way could be to use a clique reduction of the original hypergraph. However, during clique reduction, the degree of a node in the resultant graph does not remain the same as its degree in the original hypergraph, as verified below.

Lemma 6.1.1. *For the clique reduction of a hypergraph with incidence matrix H, the degree of a node i in the reduced graph is given by:*

$$k_i = \sum_{e \in E} H(i,e) w(e) (\delta(e) - 1)$$

where $\delta(e)$ and $w(e)$ are the degree and weight of a hyperedge e respectively.

Proof. For the clique reduction, the adjacency matrix of the resultant graph is given by

$$A^{clique} = HWH^T$$

$$(HWH^T)_{ij} = \sum_{e \in E} H(i,e)w(e)H(j,e)$$

In the resultant graph, each node has a self-loop that can be removed, since they are not cut during the clustering process. This is achieved by explicitly setting $A^{clique}_{ii} = 0$ for all i. Considering this, the degree of a node i in the resultant graph can be written as:

$$\begin{aligned} k_i &= \sum_j A^{clique}_{ij} \\ &= \sum_j \sum_{e \in E} H(i,e)w(e)H(j,e) \\ &= \sum_{e \in E} H(i,e)w(e) \sum_{j:j \neq i} H(j,e) \\ &= \sum_{e \in E} H(i,e)w(e)(\delta(e) - 1) \end{aligned}$$

This completes the proof. □

From the above lemma, we can infer that in the clique reduction of a hypergraph, the degree of a node is not preserved and for each hyperedge e, it is overcounted by a factor of $(\delta(e) - 1)$. We can hence scale down the node degree in the reduced graph by a factor of $(\delta(e) - 1)$. This results in the following reduction equation,

$$A^{hyp} = HW(D_e - I)^{-1}H^T \tag{6.3}$$

We can now verify that the above adjacency matrix preserves the hypergraph node degree.

Proposition 6.1.2. *For the reduction of a hypergraph given by the adjacency matrix* $A^{hyp} = HW(D_e - I)^{-1}H^T$*, the degree of a node* i *in the reduced graph (denoted* k_i*) is equal to its degree* $d(i)$ *in the original hypergraph.*

Proof. We have,

$$(HW(D_e - I)^{-1}H^T)_{ij} = \sum_{e \in E} \frac{H(i,e)w(e)H(j,e)}{\delta(e) - 1}$$

Following a similar argument from the previous theorem, we can explicitly set $A^{hyp}_{ii} = 0$ for all i. The degree of a node in the reduced graph can be written as

$$\begin{aligned} k_i &= \sum_j A^{hyp}_{ij} \\ &= \sum_{e \in E} \frac{H(i,e)w(e)}{\delta(e) - 1} \sum_{j:j \neq i} H(j,e) \\ &= \sum_{e \in E} H(i,e)w(e) \\ &= d(i) \end{aligned}$$

□

With Eq. 6.3, we can reduce a given hypergraph to a weighted graph and zero out its diagonals by explicitly setting the diagonal entries to zero. The hypergraph modularity matrix can subsequently be written as,

$$B_{ij}^{hyp} = A_{ij}^{hyp} - P_{ij}^{hyp}$$

This new modularity matrix can be used in Eq. 2.4 to obtain an expression for the hypergraph modularity and can then be used in conjunction with a Louvain-style algorithm.

$$Q^{hyp} = \frac{1}{2m} \sum_{ij} B_{ij}^{hyp} \delta(g_i, g_j) \tag{6.4}$$

6.1.1 Fundamental Observations:

- B^{hyp} exhibits all spectral properties of an undirected weighted graph's modularity matrix (Bolla *et al.* (2015); Fasino and Tudisco (2016)).
- As with any undirected weighted graph (Blondel *et al.* (2008)), Q^{hyp} ranges from -1 to $+1$.
- A negative value of Q^{hyp} indicates a clustering assignment, where a node pair (i, j) from the same cluster participates in lesser than the expected number of hyperedges. This situation may arise when the number of within-cluster edges is lower than the number of across cluster edges.
- A positive value of Q^{hyp} indicates a clustering assignment, where a node pair (i, j) from the same cluster participates in more than the expected number of hyperedges. In graphs, typically, a modularity value higher than 0.3 is considered to be significant (Clauset *et al.* (2004)).
- $Q^{hyp} = 0$ indicates a clustering assignment, where a node pair (i, j) from the same cluster participates in the expected number of hyperedges. This situation can occur because of the random assignment of nodes to the clusters.

In the rest of the section, we will analyze the properties of the proposed modularity function. We will relate the graph reduction equation to the random walk model for hypergraphs. The relation establishes the link with earlier works on hypergraph clustering, where the random walk strategies were employed (Zhou *et al.* (2007)).

6.1.2 Connection to Random Walks:

Consider the clique reduction of the hypergraph. We can distribute the weight of each hyperedge uniformly among the edges in its associated clique. All nodes within a single hyperedge are assumed to contribute equally; a given node would receive a fraction of the weight of each hyperedge it belongs to. The number of edges each node is connected to from a hyperedge e is $\delta(e) - 1$. Hence by dividing each hyperedge weight by the number of edges in the clique, we obtain the normalized weight matrix $W(D_e - I)^{-1}$. Introducing this in the weighted clique formulation results in the proposed reduction $A^{hyp} = HW(D_e - I)^{-1}H^T$.

Another way of interpreting this reduction is to consider a random walk on the hypergraph in the following manner -

- pick a start node i
- select a hyperedge e containing i, proportional to its weight $w(e)$
- select a *new* node from e uniformly (there are $\delta(e) - 1$ choices)

The behaviour described above is captured by the following random walk transition model -

$$P_{ij} = \sum_{e \in E} \frac{w(e)h(i,e)}{d(i)} \frac{h(j,e)}{\delta(e) - 1}$$

$$\implies P = D_v^{-1} HW(D_e - I)^{-1} H^T$$

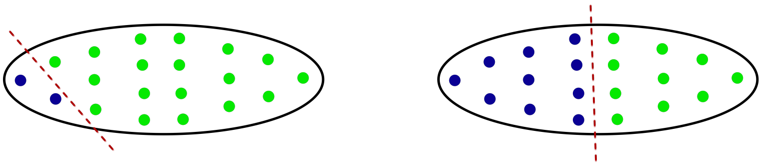
By comparing the above with the random walk probability matrix for graphs ($P = D^{-1}A$) we can recover the reduction $A^{hyp} = HW(D_e - I)^{-1}H^T$.

6.2 ITERATIVE HYPEREDGE REWEIGHTING

When clustering graphs, it is desired that edges within clusters are greater in number than edges between clusters. Hence when trying to improve clustering, we look at minimizing the number of between-cluster edges that get cut. For a hypergraph, this would be done by minimizing the total volume of the hyperedge cut (Zhou *et al.* (2007)). Consider the two-clustering problem, where the task is to divide the set V into two clusters S and S^c. Zhou et al. (Zhou *et al.* (2007)) observed that the volume of the cut ∂S is directly proportional to $\sum_e w(e)|e \cap S||e \cap S^c|$, for a hypergraph whose vertex set is partitioned into two sets S and S^c. For a hyperedge e, which has its vertices in both S and S^c, the product $|e \cap S||e \cap S^c|$ can be interpreted as the number of cut sub-edges within a clique reduction. It can be seen that this product is maximized when the cut is balanced and there are an equal number of vertices in S and S^c. In such a case, there will be $\left(\frac{\delta(e)}{2}\right)^2$ sub-edges getting cut. On the other hand, when all vertices of e go into one partition and the other partition is left empty, the product is zero. Similarly, if one of the vertices of e go into one partition and the other partition contains all $\delta(e) - 1$ vertices, then the product is $\delta(e) - 1$. A min-cut algorithm would favor cuts that are as unbalanced as possible, as a consequence of the minimization of $|e \cap S||e \cap S^c|$. In the sequel, we will present the intuition behind our proposed iterative re-weighting technique followed by its mathematical formulation.

Intuition: While clustering in graphs, when an edge gets cut between two clusters, one of its nodes becomes a member of the first cluster, and the other node becomes part of the second cluster. But in hypergraphs, a hyperedge can get cut in multiple ways. When a hyperedge gets cut, if the majority of its vertices go into the first cluster c_1 and only a smaller fraction of vertices go into the second cluster c_2, then it is more likely that the vertices going into second cluster are similar to the rest and should be drawn into the first cluster. On the other hand, if a hyperedge gets cut equally across clusters, then its vertices are equally likely to be part of any cluster; hence it is less informative than a hyperedge that gets an unbalanced cut. Building on this idea, we would want to cut the less informative hyperedges (the ones getting balanced cut), and more informative hyperedges that got unbalanced cut to be left uncut.

This can be done by increasing the weights of hyperedges that get unbalanced cuts, and (relatively) decreasing the weights of hyperedges that get more balanced cuts. We know that an algorithm that tries to minimize the volume of the hyperedge boundary would try to cut as few heavily weighted hyperedges as possible. Since the hyperedges that had more unbalanced cuts get a higher weight, they are less likely to be cut after reweighting, and instead would reside inside a cluster. Hyperedges that had more balanced cuts get a lower weight, and on reweighting, continue to get balanced cuts. Thus after reweighting and clustering, we would observe fewer hyperedges between clusters, and more hyperedges pushed into clusters. Moreover, after reweighting, we expect that the hyperedges getting cut between clusters should get balanced cuts. In the remaining section, we will formally present the solution mentioned above.



$$t = \left(\frac{1}{2} + \frac{1}{18}\right) \times 20 = 11.111 \qquad t = \left(\frac{1}{10} + \frac{1}{10}\right) \times 20 = 4$$

Figure 6.1: Reweighting for different hyperedge cuts

Now, we formally develop a reweighting scheme that satisfies the properties described above - increasing weight for a hyperedge that received a more unbalanced cut, and decreasing weight for a hyperedge that received a more balanced cut. Considering the case where a hyperedge gets partitioned into two clusters with k_1 and k_2 nodes in each partition ($k_1, k_2 \neq 0$), the following equation operationalizes the above metnioned scheme -

$$t = \left(\frac{1}{k_1} + \frac{1}{k_2}\right) \times \delta(e) \tag{6.5}$$

Here the multiplicative coefficient, $\delta(e)$, seeks to keep t independent of the number of vertices in the hyperedges. Note that for a hyperedge e with two partitions, $\delta(e) = k_1 + k_2$. Figure 6.1 illustrates an example where t takes two different values depending on the cut.

To see why this satisfies our desired property, note that t is minimized when k_1 and k_2 are equal. It can be verified by the following proposition.

Proposition 6.2.1. *In the function,* $t = \left(\frac{1}{k_1} + \frac{1}{k_2}\right) \times \delta(e)$*, the minimum value of* $t = 4$*, and it is achieved when* $k_1 = k_2 = \frac{\delta(e)}{2}$*. Here, for a hyperedge* e*,* $\delta(e)$ *is its cardinality and* k_i *represents the number of nodes in the* i^{th} *partition.*

Proof. Let $k_i \in \mathbb{Z}^+$

Then,

$$\begin{aligned} t &= \Big(\frac{1}{k_1} + \frac{1}{k_2}\Big) \times \delta(e) \\ &\text{(by substituting } \delta(e) = k_1 + k_2) \\ &= \frac{k_1^2}{k_1 k_2} + \frac{k_2^2}{k_1 k_2} + 2 \\ &= \frac{k_1}{k_2} + \frac{k_2}{k_1} + 2 + (2 - 2) \\ &= \Big(\sqrt{\frac{k_1}{k_2}} - \sqrt{\frac{k_2}{k_1}}\Big)^2 + 4 \end{aligned}$$

$\Big(\sqrt{\frac{k_1}{k_2}} - \sqrt{\frac{k_2}{k_1}}\Big)^2$ is minimized when $k_1 = k_2$ and the resultant value of $t = 4$.

□

Note: It can be observed that Eq. 6.5 coincides with the ratio between arithmetic mean (AM) and harmonic mean (HM) of the two numbers k_1 and k_2. More precisely, we can write

$$t = 4 \times \frac{\text{AM}(k_1, k_2)}{\text{HM}(k_1, k_2)}$$

By using the fact that $\text{AM}(k_1, k_2) \geq \text{HM}(k_1, k_2)$, and $\text{AM}(k_1, k_2) = \text{HM}(k_1, k_2)$ only when $k_1 = k_2$, we can obtain the similar result to Proposition 6.2.1.

We can then generalize Eq. 6.5 to c partitions as follows -

$$w'(e) = \frac{1}{m} \sum_{i=1}^{c} \frac{1}{k_i + 1} [\delta(e) + c] \tag{6.6}$$

Here, $+1$ term in the denominator accounts for the cases when $k_i = 0$. To compensate for this extra $+1$, $+c$ has been added to the numerator. Additionally, m is the number of hyperedges, and the division by m is added to normalize the weights (Fig. 6.1). During the first iteration of the algorithm, we find clusters in the hypergraph using its default weights. At the end of the first iteration, we find the updated weights using the Eq. 6.6. It can be seen that for a hyperedge e if it does not get balanced cut, the $w'(e)$ will not be minimized, and its value will be proportional to the extent to which it gets unbalanced cut. Thus, updating hyperedge weights by Eq. 6.6 suffices our purpose.

At step $t + 1$, let $w_t(e)$ be the weight of hyperedge e till the previous iteration. Using Eq. 6.6, $w'(e)$ can be computed for the current iteration. The weight update equation can be written as,

$$w_{t+1}(e) = \alpha w_t(e) + (1 - \alpha)w'(e) \tag{6.7}$$

Here, α is a hyperparameter which decides the importance to be given to newly calculated weights over the current weights of hyperedges. The complete algorithm for modularity maximization on hypergraphs with iterative reweighting, entitled *Iteratively Reweighted Modularity Maximization (IRMM)*, is described in Algorithm 1. In rest of the section, we will demonstrate the effectiveness of the hyperedge reweighting scheme by using a toy example.

Now, we use the A^{hyp} formulation to define Hypergraph Resource Allocation, which is further used for hyperedge prediction.

Algorithm 1: Iteratively Reweighted Modularity Maximization (IRMM)

```
input  : Hypergraph incidence matrix $H$, vertex degree matrix $D_v$, hyperedge
         degree matrix $D_e$, hyperedge weights $W$
output: Cluster assignments cluster_ids, number of clusters $c$
// Initialize weights as $W \leftarrow I$ if the hypergraph is unweighted
repeat
    // Compute reduced adjacency matrix
    $A \leftarrow HW(D_e - I)^{-1}H^T$
    // Zero out the diagonals of A
    $A \leftarrow zero\_diag(A)$
    // Return number of clusters and cluster assignments
    cluster_ids, c = LOUVAIN_MOD_MAX(A)
    // Compute new weight for each hyperedge
    for $e \in E$ do
        // Compute the number of nodes in each cluster
        for $i \in [1, .., c]$ do
            // Set of nodes in cluster i
            $C_i \leftarrow cluster\_assignments[i]$
            $k_i = |e \cap C_i|$
        end
        // Compute new weight
        $w'(e) = \frac{1}{m} \sum_{i=1}^{c} \frac{1}{k_i+1} (\delta(e) + c)$
        // Take moving average with previous weight
        $W_{prev}(e) \leftarrow W(e)$
        $W(e) = \alpha(w'(e) + (1 - \alpha)W_{prev}(e))$
    end
until $||W - W_{prev}|| < threshold$
```

6.2.1 Hypergraph Resource Allocation (HRA)

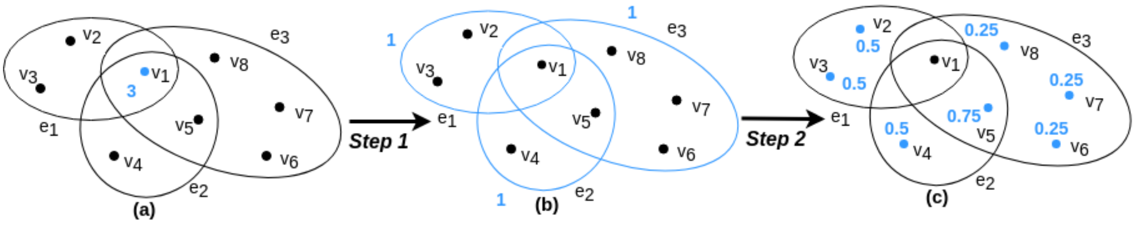


Figure 6.2: Example illustrating resource transfer directly between nodes in a hypergraph.

HRA index is motivated by the *RA* index defined for graphs. *RA* index is defined for the node-pairs (x, y) which are not directly connected, as these node-pairs are potential new edges. Unlike graphs, new hyperedges can have nodes that are already connected in the hypergraph. Thus, node x can transfer its resources to y by either a direct connection (HRA_{direct}) or via common neighbors ($HRA_{indirect}$). To determine HRA_{direct}, assume node x has a resource of $d(x)$ units. Node x uniformly distributes its resources to all hyperedges that include x. In the next step, the resource allocated to each hyperedge is uniformly distributed to the nodes of rhe hyperedge apart from x. The amount of resource node y receives directly from node x is given by $HRA_{direct}(x, y) = \sum_{e, s.t., x, y \in e} \frac{1}{\delta(e)-1}$ (this is equal to $A^{hyp}(x, y)$). Figure 6.2 illustrates the direct transfer of resource between nodes in a toy hypergraph. Initially, v_1 distributes its resources to all three hyperedges uniformly, as shown in Figure 6.2(b). In the next step, the unit resource allocated to each hyperedge is uniformly distributed to its nodes excluding v_1.

To determine $HRA_{indirect}$, assume node z is a common neighbor of x and y. Node z receives $HRA_{direct}(x, z)$ amount of resource from x and then distributes it to all its neighbors. Node y being a neighbor of z, receives $HRA_{direct}(x, z) \times \frac{1}{d(z)} \times HRA_{direct}(z, y)$ amount of resource. Total resource received by y through all common neighbors is given by:

$$HRA_{indirect}(x, y) = \sum_{z \in N(x) \cap N(y)} HRA_{direct}(x, z) \times \frac{1}{d(z)} \times HRA_{direct}(z, y)$$

Combining HRA_{direct} score with $HRA_{indirect}$, we define the similarity between x and y as,

$$HRA_{xy} = HRA_{direct}(x, y) + HRA_{indirect}(x, y)$$

Notice that the *HRA* computation depends only on the local neighborhood. Thus, HRA can be computed efficiently for very large networks.

6.2.2 Node-Hyperedge Attachment Score (NHAS)

Social networks are known to possess *homophily*; for instance, consider a musical band e looking for a $guitarist$. Assume guitarists x and y are known to be equally good, and guitarist x has previously worked with few members from e. Then guitarist x is more likely to be part of e as compared to y. Here, the musical band represents a hyperedge and band members are the nodes. Following the principles of *homophily* and using *HRA* to capture node-node similarity, we formally define $NHAS$ as follows:

$$NHAS_{x,e} = \frac{1}{|e|}\Big(\sum_{y \in e} HRA_{xy}\Big)$$

6.2.3 HPRA: Hyperedge Prediction using Resource Allocation

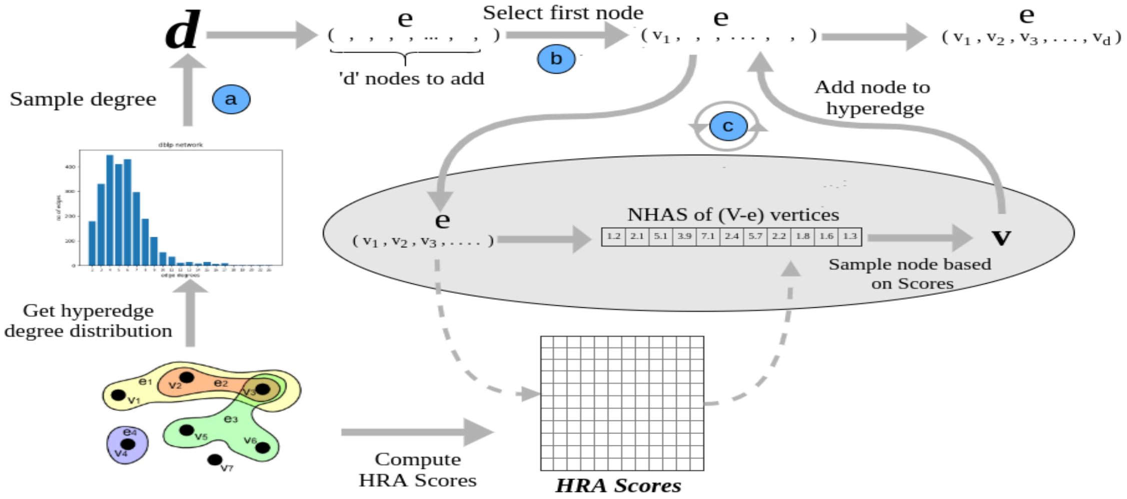


Figure 6.3: An Illustration of HPRA. (a) cardinality of new hyperedge *d* is sampled proportional to hyperedge degree distribution. (b) first node is chosen based on Preferential Attachment and added to *e*. (c) subsequent nodes are sampled from set V\e based on the *NHAS* scores and added to *e*. This step is repeated until *d* nodes are added to *e*

The problem of hyperedge prediction can be disintegrated into the following sub-problems:

1. What should be the cardinality of the new hyperedge?
2. Once the hyperedge-cardinality is determined, which node should be the first member of the new hyperedge?
3. What are the other nodes that should be part of the new hyperedge?

We propose *HPRA* (Algorithm 2), which addresses the aforementioned problems in the following ways, and also depict the same in Figure 6.3.

Algorithm 2: Hyperedge Prediction using Resource Allocation (HPRA)

```
Input: Hypergraph Incidence Matrix H, Node set V, Hyperedge Degree
       Distribution HDD
Output: Predicted Hyperedge e
// Initialize hyperedge e
$e \leftarrow \{\}$
// Sample hyperedge degree from HDD
$d \leftarrow get\_degree(hyperedge\_degrees, prob = HDD)$
// Select first node using Preferential Attachment
$v_{new} = get\_node(V, prob = node\_degrees)$
$e.add(v_{new})$
while $size(e) < d$ do
    // Compute NHAS for remaining nodes
    $scores \leftarrow NHAS(e, V)$
    // Select a node based on NHAS
    $v \leftarrow get\_node(V, prob = scores)$
    $e.add(v)$
end
```

1. A hyperedge prediction algorithm is expected to preserve the structural properties of the hypergraph (Guo *et al.* (2016*a*)). One such structural property is hyperedge degree distribution. To preserve it, the cardinality of the new hyperedge has to be *in line* with the observed hyperedge degree distribution. Therefore, we sample the cardinality of the new hyperedge from the observed hyperedge degree distribution. In other words, the higher the number of observed hyperedges with degree d, the higher the probability that the new hyperedge cardinality is d.

However, there is a possibility of not encountering the hyperedges of a specific cardinality in the hypergraph. To handle such scenerios, we smoothen the hyperedge degree distribution by following a Laplace smoothing operator and work with the resultant distribution (Valcarce *et al.* (2016)). Thus, we always have a *fail-safe* probability to handle missing cardinalities in the hypergraph.

2. More often than not, social (Guo *et al.* (2016*b*)) and web networks (Kunegis *et al.* (2013)) evolve by following the principles of preferential attachment, i.e., nodes with a higher degree are more likely to form new links. Following this, once the cardinality d of new hyperedge is determined, we choose the first member of the hyperedge with probability proportional to the node degrees.
3. As the new hyperedge e is initialized with one node v_{new}, we compute *NHAS* (Algorithm 3) of all the remaining nodes and the new hyperedge e. We repeatedly sample $(d-1)$ node from the set $V \setminus e$ based on their *NHAS* score and add it to e.

Algorithm 3: Node-Hyperedge Attachment Scores

```
Input:  Edge e, Node set V, HRA score matrix HRA
Output: Node-Hyperedge Attachment Scores scores)
// Initialize scores
$scores \leftarrow zeroes(size(V))$
// Compute NHAS for each node in $V \setminus e$
for $v_i$ $in$ $V$ do
    if $v_i$ $not$ $in$ $e$ then
        for $v_j$ $in$ $e$ do
            $scores[i] \leftarrow scores[i] + HRA(v_i, v_j)$
        end
        $scores[i] \leftarrow \frac{1}{size(e)} * (scores[i])$
    end
end
```

6.3 SUMMARY

In this chapter, we introduced the mathematical framework of hypergraphs and presented methods for hypergraph clustering and hyperedge prediction. We make use of the hypergraph reduction equation in the modularity maximization algorithm, and the same reduction is instrumental in defining the Node Hyperedge Attachment Score for hyperedge prediction. Now, we are in a position to validate the Algorithm 1 and Algorithm 2 in the next chapter.

CHAPTER 7

Applications: Hypergraph clustering and Hyperedge prediction

In this chapter, we present the experimental setup for hypergraph clustering and hyperedge prediction using the proposed methods. We also present how the existing methods can be deployed to solve these problems and use these methods as our baselines. We conduct our experiments on extensive datasets and analyze the results. We begin with the problem of Hypergraph clustering.

7.1 HYPERGRAPH CLUSTERING

Analogous to the graph clustering task, *Hypergraph clustering* seeks to discover densely connected components within a hypergraph (Schaeffer (2007)). This has been the subject of several research works by various communities with applications to various problems such as VLSI placement (Karypis and Kumar (1998)), discovering research groups(Kamiński *et al.* (2019)), image segmentation (Kim *et al.* (2011)), de-clustering for parallel databases (Liu and Wu (2001)) and modeling eco-biological systems (Estrada and Rodriguez-Velazquez (2005)), among others. A few early works on hypergraph clustering (Leordeanu and Sminchisescu (2012); Bulo and Pelillo (2013); Agarwal *et al.* (2005); Shashua *et al.* (2006); Liu *et al.* (2010)) are confined to k-uniform hypergraphs where each hyperedge connects exactly k number of nodes. However, most of the real-world hypergraphs have arbitrary-sized hyperedges, which makes these methods unsuitable for several practical applications. Within the machine learning community, Zhou et al. (Zhou *et al.* (2007)), were among the earliest to look at learning on non-uniform hypergraphs. They sought to support *spectral clustering* methods (for example see (Shi and Malik (2000); Ng *et al.* (2002)) on hypergraphs and defined a suitable hypergraph Laplacian for this purpose. This effort, like many other existing methods for hypergraph learning, makes use of a reduction of the hypergraph to a graph (Agarwal *et al.* (2006)) and has led to follow-up work (Louis (2015)). Spectral based methods involve expensive computations to determine the eigenvector (multiple eigenvectors in case of multiple clusters), which makes these methods less suitable for large hypergraphs.

An alternative methodology for clustering on simple graphs (those with just dyadic relations) is *modularity maximization* (Newman (2006)). This class of methods, in addition to providing a useful metric for evaluating cluster quality through the *modularity* function, also returns the number of clusters automatically and avoids the expensive eigenvector computation step - typically associated with other popular methods such as spectral clustering. In practice, a greedy optimization algorithm known as the Louvain method (Blondel *et al.* (2008)) is commonly used, as it is known to be fast and scalable and can operate on large graphs.

However, extending the modularity function to hypergraphs is a non-trivial task, as a node-degree preserving null model would be required, analogous to the graph setting. A straightforward procedure would be to leverage clique reduction, to reduce a hypergraph to a simple graph and then apply a conventional modularity-based solution. Such an approach ignores the underlying super-dyadic nature of interactions and thus loses critical information. Additionally, a clique reduction method would not preserve the node degree sequence of the original hypergraph, which is vital for the null model that modularity maximization techniques are typically based on.

Recently, there have been several attempts to define the null models on the hypergraphs. Chodrow (Chodrow (2020)) proposed a Monte Carlo Markov Chain based method, in which random hypergraphs are generated by pairwise reshuffling the edges in the bipartite projection. A more recent study involves the generalization of the celebrated Chung-Lu random graph model (Chung and Lu (2002)) to hypergraphs, and employs it to solve the problem of hypergraph clustering (Kamiński *et al.* (2019)). The hypergraph modularity objective proposed by Kaminski et al. (Kamiński *et al.* (2019)) only counts the participation of hyperedges completely contained inside a cluster. Though this assumption enables the analytic tractability of the solution, it limits its applicability to real world hypergraphs where hyperedges can be of arbitrary size. There exists a parallel line of inquiry where hypergraphs are viewed as simplicial complexes, and null models are defined through the preservation of topological features of interest (Giusti *et al.* (2016); Courtney and Bianconi (2016); Young *et al.* (2017)). Such models make a strong assumption - that of subset-inclusion[1], which may not hold often in real-world data. We use the *IRMM* algorithm defined in the previous chapter for hypergraph clustering. Now, we discuss the experimental details.

7.1.1 Evaluation on Ground Truth

In this section, we will present the experiments conducted to validate the proposed methods. We used the Rand Index, average F1 measure (Yang and Leskovec (2012) and purity, three popular metrics to evaluate the clustering quality. We will start with a

---

[1] Subset inclusion assumes that, for each hyperedge, any subset of the nodes is also a hyperedge. For example, if authors (A, B, C, D) publish a paper together and form a hyperedge in a co-authorship hypergraph, subset-inclusion would also include all possible subsets such as (A, B), (B, C, D), etc. as observed hyperedges, which may not hold in the real-world datasets.

brief introduction to the Louvain method, followed by details on the experimental setup and datasets used.

The Louvain method: The Louvain method is a greedy optimization method for detecting communities in large networks (Blondel *et al.* (2008)). The method works on the principle of grouping the nodes that maximize the overall modularity. Since checking all possible cluster assignments is impractical, the Louvain algorithm uses a heuristic that is known to work well on real-world graphs. The method starts by assigning each node to its own cluster and merging those clusters, resulting in the highest modularity gain. Merged clusters are treated as single nodes, and again those cluster-pairs merge that result in the highest modularity gain. If there are no cluster pairs left that will further increase the overall network modularity, the algorithm stops and returns the clusters.

Fixing the number of clusters: We use the Louvain algorithm to maximize the hypergraph modularity as per Eq 6.4. Since this method uses a node-degree-preserving graph reduction, we refer to it as *NDP-Louvain* (Node Degree Preserving Louvain). Louvain algorithm automatically returns the number of clusters. To get a predefined number of clusters c, we use agglomerative clustering (Ding and He (2002)) on the top of clusters obtained by the Louvain algorithm. For the linkage criterion, we use the average linkage. It is a bottom-up hierarchical clustering method. The algorithm constructs a dendrogram that exhibits pairwise similarity among clusters. At each step, two clusters with the shortest distance are merged into a single cluster. The distance between any two clusters c_i and c_j is taken to be the average distance of all distances $d(x, y)$, where node $x \in c_i$ and node $y \in c_j$.

The proposed methods are shown in the results table as *NDP-Louvain* and *IRMM*.

7.1.2 Settings for IRMM

We investigate the effect of the hyperparameter α using a grid search over the set $[0.1, 0.9]$ with a step size of 0.1. We did not observe any difference in the resultant Rand Index, purity, and F1 scores. While tuning the α, we witnessed a very minimal difference in the convergence rate, over a wide range of values (for example, 0.3 to 0.9 on the TwitterFootball dataset). It can be noted that α is a scalar value in a moving average; it will not cause any significant variation in the resulting weights. In our experiments, we decided to set it at $\alpha = 0.5$. We stop the iterations if the difference between the mod of two subsequent weight assignments is less than a set threshold. In our experiments, we set chose to set this threshold at $threshold = 0.01$

7.1.3 Compared Methods

To evaluate the performance of our proposed methods, we compared the following baselines.

Clique Reductions: We reduced the original hypergraph using a clique reduction ($A = HWH^T$) and then applied the Louvain method and Spectral Clustering.

Hypergraph-based Spectral Clustering: We use the hypergraph-based spectral clustering method, as defined in (Zhou *et al.* (2007)). The given hypergraph is reduced to a graph ($A = D_v^{\frac{-1}{2}} H W D_e^{-1} H^T D_v^{\frac{-1}{2}}$) and its Laplacian is calculated. The top k eigenvectors of the Laplacian are found and clustered by the bisecting-k-means clustering procedure. In the results table, this method is referred to as *Zhou-Spectral*.

PaToH[2] **and hMETIS**[3]**:** These are popular hypergraph partitioning algorithms that work on the principles of coarsening the hypergraph before partitioning. The coarsened hypergraph is partitioned using expensive heuristics. In our experiments, we used the original implementations from the corresponding authors.

7.1.4 Datasets

Dataset	# nodes	# hyperedges	Avg. hyperedge degree	Avg. node degree	# classes
TwitterFootball	234	3587	15.491	237.474	20
Cora	2708	2222	3.443	2.825	7
Citeseer	3264	3702	27.988	31.745	6
MovieLens	3893	4677	79.875	95.961	2
Arnetminer	21375	38446	4.686	8.429	10

Table 7.1: Dataset Description

Dataset statistics are furnished in Table 7.1. For all datasets, we use the largest connected component of the hypergraph for our experiments. All the datasets are classification datasets, where the class labels accompany the data points. We use these class labels as the proxy for clusters. The detailed description of the hypergraph construction is given below:

MovieLens [4]**:** This is a multi-relational dataset provided by GroupLens research, where movies are represented by nodes. We construct a co-director hypergraph by using the *director* relationship to represent hyperedges. A hyperedge would connect a group of

---

[2] http://bmi.osu.edu/umit/software.html

[3] http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download

[4] http://ir.ii.uam.es/hetrec2011/datasets.html

nodes if the same individual directed them. Here, the genre of a movie represents the class of the corresponding node.

Cora and Citeseer: These are bibliographic datasets, where the nodes represent papers. In each dataset, a set of nodes is connected by a hyperedge if they involve the same set of words (after removing low frequency and stop words). Different disciplines were used as clusters. (Sen *et al.* (2008)).

TwitterFootball: This is a social network taken from the Twitter dataset (Greene *et al.* (2012). This dataset involves players of 20 football clubs (classes) of the English Premier League. Here, the nodes represent players, and if a set of players are co-listed, then the corresponding nodes are connected by a hyperedge.

Arnetminer: This is a large bibliographic dataset (Tang *et al.* (2008)). Here, the nodes represent papers, and a set of nodes are connected if the corresponding papers are co-cited. The nodes in the hypergraph are accompanied by Computer Science sub-disciplines. Different sub-disciplines were used as clusters.

7.1.5 Experiments

For the different datasets, we compare the Rand Index (Rand (1971)), purity (Schütze *et al.* (2008)), and average F1 scores (Yang and Leskovec (2013)) on all the methods discussed earlier. The number of clusters was first set to that returned by the Louvain method, in an unsupervised fashion. This is what would be expected in a real-world setting, where the number of clusters is not given apriori. Table 7.2 shows the results of this experiment.

	Citeseer	Cora	MovieLens	TwitterFootball	Arnetminer
hMETIS	0.6504	0.7592	0.4970	0.7639	0.0416
PaToH	0.6612	0.6919	0.4987	0.7553	0.0052
Spectral	0.7164	0.2478	0.4806	0.7486	0.0610
Zhou-Spectral	**0.8210**	0.5743	0.4977	0.9016	0.0628
Louvain	0.7361	0.7096	0.4898	0.6337	0.0384
NDP-Louvain	0.7899	0.8238	0.4988	0.9056	0.0821
IRMM	0.7986	**0.8646**	**0.5091**	**0.9448**	**0.0967**

(a) Rand Index scores against ground truth.

	Citeseer	Cora	MovieLens	TwitterFootball	Arnetminer
hMETIS	0.5894	0.6596	0.6893	0.2556	0.6831
PaToH	0.6271	0.5912	0.7017	0.3176	0.3928
Spectral	0.4629	0.3897	0.6832	0.8114	0.9216
Zhou-Spectral	0.5287	0.4145	0.7118	0.8325	0.9378
Louvain	0.7190	0.6836	0.7189	0.8054	0.9138
NDP-Louvain	0.7307	0.7597	0.7245	0.8829	0.9691
IRMM	**0.7659**	**0.8138**	**0.7291**	**0.8948**	**0.9765**

(b) Purity scores against ground truth.

	Citeseer	Cora	MovieLens	TwitterFootball	Arnetminer
hMETIS	0.1087	0.1075	0.1291	0.3197	0.0871
PaToH	0.0532	0.1171	0.1104	0.1132	0.0729
Spectral	0.1852	0.1291	0.1097	0.4496	0.0629
Zhou-Spectral	0.2774	0.2517	0.118	0.5055	0.0938
Louvain	0.1479	0.2725	0.1392	0.2238	0.1378
NDP-Louvain	0.2782	0.3248	0.1447	0.5461	0.1730
IRMM	**0.4019**	**0.3709**	**0.1963**	**0.5924**	**0.1768**

(c) Average F1 scores against ground truth.

Table 7.2: Rand Index, Purity and Average F1 scores against ground truth; the number of clusters for hMETIS, PaToH, Spectral, and Zhou-Spectral is set to the number of clusters returned by the IRMM method are 13, 79, 8, 18, and 1358 for Citeseer, Cora, Movielens, TwitterFootball, and Arnetminer, respectively. Louvain, NDP-Louvain, and IRMM return the number of clusters on their own. IRMM performs significantly ($p < 0.1$) better than those baseline methods that are underlined.

Secondly, we ran the same set of methods with the number of ground truth classes set as the number of clusters. In the case of Louvain method, the clusters obtained are merged using the post-processing technique explained earlier. The results of this experiment are given in Table 7.3. When *Louvain* method and IRMM return fewer clusters than the number of ground truth classes, we do not report the results and leave the entries as “-."

We also plotted the results for varying number of clusters using the same methodology described above, to assess our method’s robustness (Figure 7.1). In all datasets but *Arnetminer*, we set the number of clusters to a minimum value such as two and then increase it by a factor of two. For *Arnetminer*, IRMM returns a very large number of clusters; we set the initial number of clusters to ten and increase it by a factor of ten. For all datasets, the maximum number of clusters is set to the number of clusters returned by the IRMM method. When *Louvain* and *NDP-Louvain* methods return a fewer number of clusters than IRMM, the corresponding curves in Figure 7.1 are left truncated.

7.1.6 Results and Analysis

We show that the proposed methods - *NDP-Louvain* and IRMM perform consistently better on all the datasets (except on one dataset with RI measure). To test the robustness of the proposed method, we vary the number of clusters and report the results in the latter half of the section. To investigate the effect of the reweighting scheme, we report the distribution of the sizes of hyperedges getting cut. This is followed by testing the scalability of the proposed algorithm against one of the competitive baseline. We will start by discussing the empirical evaluation of the proposed methods.

	Citeseer	Cora	MovieLens	TwitterFootball	Arnetminer
hMETIS	0.6891	0.7853	0.5028	0.7697	0.3116
PaToH	0.7312	0.7208	0.4984	0.7618	0.1820
Spectral	0.7369	0.3117	0.4812	0.7765	0.3762
Zhou-Spectral	**0.8267**	0.5845	0.5006	0.9112	0.3851
Louvain	-	0.7096	0.4982	-	0.4198
NDP-Louvain	0.8197	0.8441	0.5119	-	0.5359
IRMM	0.8245	**0.889**	**0.5347**	-	**0.5506**

(a) Rand Index scores; number of clusters set to the number of ground truth classes

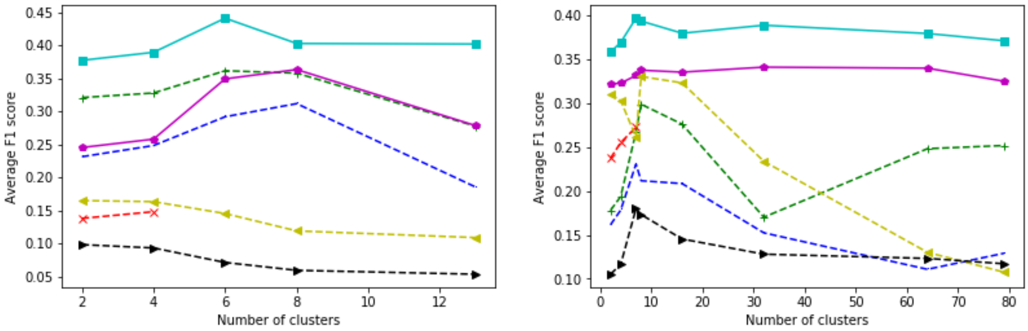
	Citeseer	Cora	MovieLens	TwitterFootball	Arnetminer
hMETIS	0.5249	0.6359	0.6914	0.2354	0.2984
PaToH	0.5724	0.6498	0.7139	0.2419	0.2391
Spectral	0.4839	0.5819	0.7294	0.7815	0.5169
Zhou-Spectral	0.5374	0.6115	0.742	0.8191	0.5827
Louvain	-	0.7136	0.7364	-	0.4837
NDP-Louvain	0.7495	0.7441	0.7429	-	0.5968
IRMM	**0.7732**	**0.779**	**0.7737**	-	**0.6173**

(b) Cluster purity scores; number of clusters set to the number of ground truth classes

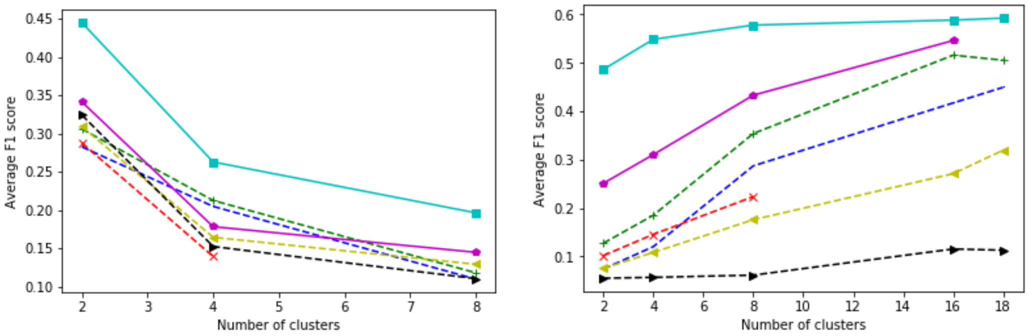
	Citeseer	Cora	MovieLens	TwitterFootball	Arnetminer
hMETIS	0.1451	0.2611	0.4445	0.3702	0.3267
PaToH	0.0710	0.1799	0.3239	0.1036	0.2756
Spectral	0.2917	0.2305	0.2824	0.4345	0.387
Zhou-Spectral	0.3614	0.2672	0.3057	0.5377	0.4263
Louvain	-	0.2725	0.2874	-	0.4587
NDP-Louvain	0.3491	0.3314	0.3411	-	0.4948
IRMM	**0.4410**	**0.3966**	**0.4445**	-	**0.5299**

(c) Average F1 scores; number of clusters set to the number of ground truth classes

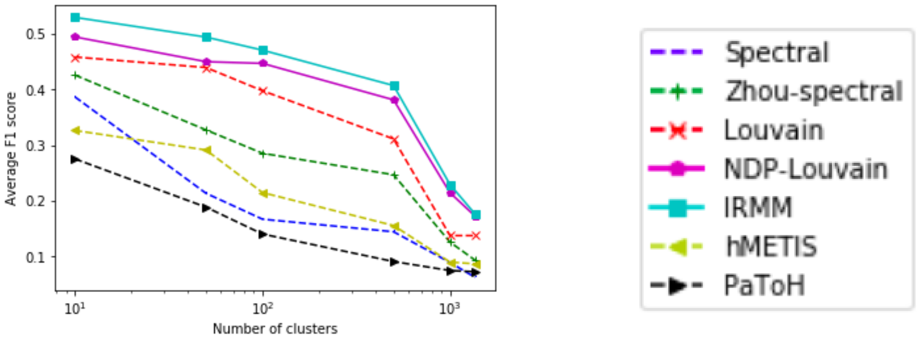
Table 7.3: Rand Index, Purity and Average F1 scores against ground truth; the number of clusters is set to the number of ground truth classes. Citeseer, Cora, Movielens, TwitterFootball, and Arnetminer have 6, 7, 2, 20, and 10 classes, respectively. On some datasets, the *Louvain* and IRMM methods return fewer clusters than the number of ground truth classes. In such cases, we do not report the results and leave the entries as “-." IRMM performs significantly ($p < 0.1$) better than those baseline methods that are underlined.



(a) Citeseer (b) Cora



(c) MovieLens (d) TwitterFootball



(e) Arnetminer

Figure 7.1: F1 scores for varying number of clusters. Here, x-axis represent the number of clusters and y-axis indicates F1 score.

From the Tables 7.2 and 7.3, it is evident that IRMM gives the highest cluster purity scores and average F1 scores across all the datasets and the highest Rand Index scores are obtained on all except *Citeseer* dataset. Besides the fact that IRMM significantly outperforms over other methods, we want to emphasize on the following two observations:

Superior performance of hypergraph based methods:

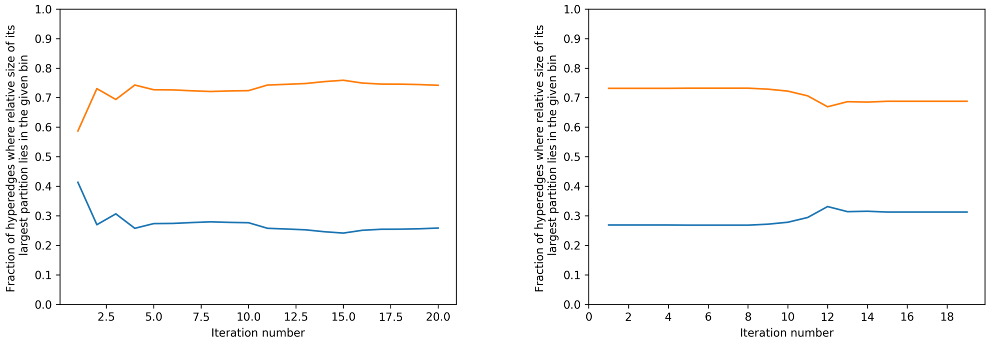
It is evident that hypergraph based methods perform consistently better than their clique based equivalents. Results indicate that *Zhou-Spectral* and *NDP-Louvain* are better than *Spectral* and *Louvain* respectively. Hence, preserving the super-dyadic structure helps in getting a better cluster assignment.

The proposed iterative reweighting scheme helps to boost up the performance:

The proposed hyperedge reweighting scheme aids in the performance across all datasets. It must be noted that the first iteration of IRMM is the *NDP-Louvain* and IRMM performance is consistently better than the *NDP-Louvain* method, which shows that balancing the hyperedge cut enhances the cluster quality.

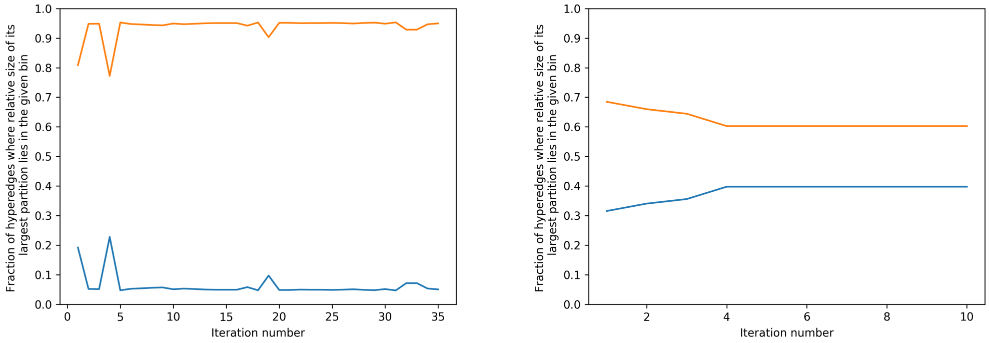
Effect of Reweighting on Hyperedge Cuts

Consider a hyperedge that is cut into different clusters. Looking at Eq. 6.6, we can see that $w'(e)$ is minimized when all the partitions are of equal size, and maximized when one of the partitions is much larger than the other. The iterative reweighting procedure is designed to increase the number of hyperedges with balanced partitioning, and decrease the number of hyperedges with unbalanced partitioning. As iterations pass, hyperedges that are more unbalanced should be pushed into neighbouring clusters, and the hyperedges that lie between clusters should be more balanced.



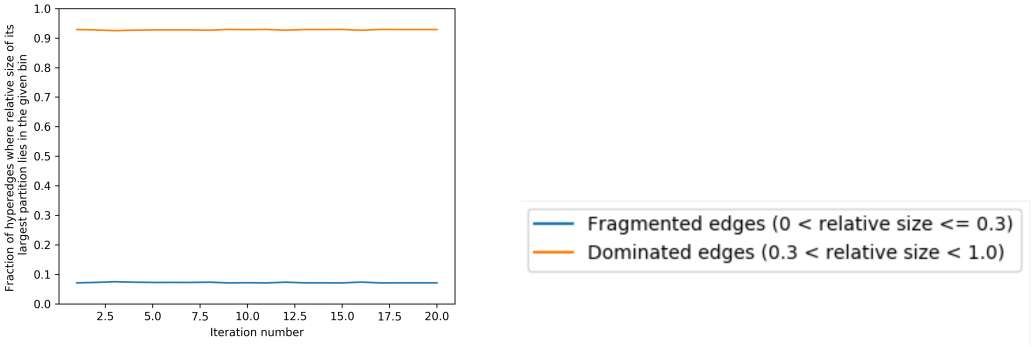
(a) Citeseer

(b) Cora



(c) MovieLens

(d) TwitterFootball



(e) Arnetminer

Figure 7.2: Effect of iterative hyperedge reweighting: % of hyperedges where the relative size of its largest partition falls in a given bin vs. no. of iterations

We analyze the effect of hyperedge reweighting in Figure 7.2. For each hyperedge, we find the relative proportion of the biggest partition and add them in the bins with interval size = 0.1. The plot illustrates the variation in the size of each bin over along with iterations.

$$\text{relative size}(e) = \max_i \frac{\text{number of nodes in cluster } i}{\text{number of nodes in the hyperedge } e}$$

If a hyperedge is a balanced cut, then the proportion of its largest partition is low; we call such hyperedges as *fragmented*. On the other hand, if a hyperedge has a very high proportion of its largest partition, then the hyperedge is not a balanced cut; we call such hyperedges as *dominated*.

On *TwitterFootball* dataset, the effect of reweighting is distinctly visible as the number of fragmented edges increases with iterations. This behavior confirms our intuition of achieving more balanced cuts with the proposed reweighting procedure. After four iterations, the method converges as we don't observe any change in the hyperedge distribution.

A similar trend is observed with the *Cora* dataset. Here, the number of fragmented edges fluctuate before their final convergence.

In the case of *Arnetminer* dataset, the change in fragmented and dominated edges is very minimal. One possible reason for such behavior could be its significantly large size as compared to the number of ground truth clusters.

In the case of *Citeseer* and *Movielens* datasets, we could not see the convergence in the change of hyperedge weights in a pre-fixed number of iterations. Though the number of hyperedges seems to fluctuate with iterations, the algorithm tries to find the best clustering at each step by using the *NDP-Louvain* algorithm. This results in the improved performance of the overall algorithm after following the refinement procedure.

Both in *Citeseer* and *Movielens* datasets, IRMM returns lesser number of clusters than *NDP-Louvain*. *NDP-Louvain* returns 16 clusters for *Citeseer* and 13 clusters for *Movielens* dataset. These number of clusters are reduced to 13 and 8 for *Citeseer* and *Movielens* datasets respectively. Thus, the refinement procedure tends to minimize the cut value along with cut-balacing.

7.1.7 Complexity analysis of the proposed method

Our proposed method, IRMM, comprises three steps - hypergraph reduction, modularity computation, and hyperedge reweighting. The hypergraph reduction involves matrix-matrix multiplication, which can be computed with $\mathcal{O}(n^3)$ complexity. We use the Louvain method to maximize the modularity of the reduced graph, which has computational complexity proportional to the number of edges in the reduced graph Traag (2015), so the overall complexity of the NDP Louvain method remains $\mathcal{O}(n^3)$. For hyperedge reweighting, we check every hyperedge that gets cut - this step has an upper bound of $\mathcal{O}(m)$. In IRMM, we iterate over these three steps k times. So the overall complexity of the proposed algorithm IRMM is $\mathcal{O}(k(n^3 + m))$. We don't get rid of the term m because, theoretically, m can be as large as 2^n. However, in practice, we see m and n attain similar values - in that case, the overall complexity of the IRMM algorithm is $O(n^3)$. Empirically, for the largest hypergraph, ArnetMiner, our method takes around an hour for each iteration. This complexity can be further improved by using modularized methods Higham (1990) of matrix-matrix multiplication.

To further motivate the extension of modularity maximization methods to the hypergraph clustering problem, we look at the scalability of the *NDP-Louvain* method against the strongest baseline, *Zhou-Spectral*. Table 7.4 shows the CPU timesfor the *NDP-Louvain* and *Zhou-Spectral* on the real-world datasets. We see that while the difference is less pronounced on a smaller dataset like *TwitterFootball*, it is much greater on the larger datasets. In particular, the runtime on Arnetminer for *NDP-Louvain* is lower by a significant margin, not having to compute an expensive eigendecomposition.

	Citeseer	Cora	MovieLens	TwitterFootball	Arnetminer
Zhou-Spectral	84.16	41.44	155.8	3.88	34790
NDP-Louvain	41.21	24.23	35.9	3.32	4311.2

Table 7.4: CPU times (in seconds) for the hypergraph clustering methods on all datasets

Note: To compute the eigenvectors for spectral clustering based method, we use of the *eig(.)* function from MATLAB. The *eig(.)* function makes use of orthogonal similarity transformations to convert the matrix into upper Hessenberg matrix followed by QR algorithm to find its eigenvectors.

Analysis on synthetic hypergraphs: On the real-world data, modularity maximization showed improved scalability as the dataset size increased. To evaluate this trend, we compared the CPU times for the *Zhou-Spectral* and *NDP-Louvain* methods on synthetic hypergraphs of different sizes. For each hypergraph, we first ran *NDP-Louvain* and found the number of clusters returned, then ran the *Zhou-Spectral* method with the same number of clusters.

Following the hypergraph generation method used in EDRW: Extended Discriminative Random Walk[5] (Satchidanand *et al.* (2015)), we generated hypergraphs with 2 classes and a homophily of 0.4 (40% of the hyperedges deviate from the expected class distribution). The hypergraph followed a modified power-law distribution, where 75% of its hyperedges contained less than 3% of the nodes, 20% of its hyperedges contained 3%-50% of the nodes, and the remaining 5% contained over half the nodes in the dataset. To generate a hypergraph, we first set the number of hyperedges to 1.5 times the number of nodes. For each hyperedge, we sampled its size k from the modified power-law distribution and chose k different nodes based on the homophily of the hypergraph. We generated hypergraphs of sizes ranging from 1000 nodes up to 10000 nodes, at intervals of 500 nodes.

Figure 7.3 shows how the CPU time varies with the number of nodes, on the synthetic hypergraphs generated as given above.

While *NDP-Louvain* is shown to run consistently faster than *Zhou-Spectral* for the same number of nodes, the difference increases as the hypergraph grows larger. In Figure 7.3, this is shown by the widening in the gap between the two curves as the number of nodes increases.

---

[5] https://github.com/HariniA/EDRW

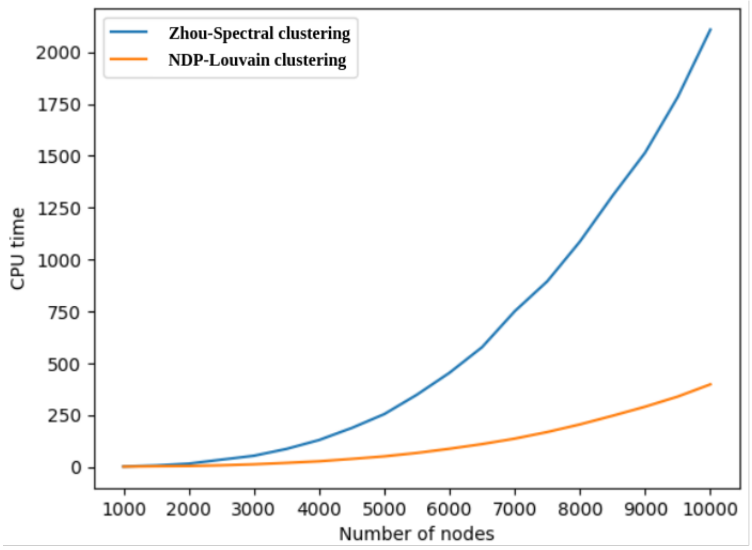


Figure 7.3: CPU time (in secs) on synthetic hypergraphs

7.2 HYPEREDGE PREDICTION

Hyperedge prediction is the problem of finding missing or future hyperedges in a given hypergraph. The problem has many real-world use cases, including reaction prediction in a network of metabolites (Zhang *et al.* (2018*a*)), predicting collaborations in an actor-actor network(Sharma *et al.* (2014)), etc. Despite having significant importance, the problem of hyperedge prediction hasn't received adequate attention, mainly because of its inherent complexity. In a graph with n nodes the number of potential edges is $\mathcal{O}(n^2)$, whereas in a hypergraph, the number of potential hyperedges is $\mathcal{O}(2^n)$. To avoid searching through the enormous space of hyperedges, current methods restrain the problem in the following two ways. One class of algorithms assumes the hypergraphs to be k-uniform. However, many real-world systems are not confined only to have interactions involving k components. Thus, these algorithms are not suitable for many real-world applications. The second class of algorithms requires a candidate set of hyperedges from which the potential hyperedges are chosen. In the absence of domain knowledge, the candidate set can have $\mathcal{O}(2^n)$ possible hyperedges, which makes this problem intractable. More often than not, domain knowledge is not readily available, thus limiting these methods. Our proposed method *HPRA - Hyperedge Prediction using Resource Allocation*, overcomes these issues and predicts hyperedges of any cardinality without using any candidate hyperedge set. *HPRA* is a similarity-based method working on the principles of the resource allocation process. We also demonstrate that *HPRA* can predict future hyperedges in a wide range of hypergraphs. In this chapter, we will demonstrate an extensive set of experiments to show that HPRA achieves statistically significant improvements over state-of-the-art methods.

7.3 EXPERIMENTS

We evaluate the performance of *HPRA* on a broad range of networks. We propose new baselines by extending state-of-the-art similarity measures to be used with *HPRA* framework by replacing the *HRA* index. Before elaborating on the experimental setup, we first introduce the baselines and datasets used for evaluation.

7.3.1 Baselines

Coordinated Matrix Minimisation (CMM) (Zhang *et al.* (2018*a*)): CMM is based on matrix factorisation in adjacency space of hypergraph. It uses the EM algorithm to determine the presence or absence of candidate hyperedges.

Spectral Hypergraph Clustering (SHC) (Zhou *et al.* (2007)): SHC models the task of hyperedge prediction as a classification problem. Hypergraph Laplacian is used to classify the new hyperedges into positive or negative class.

Common Neighbors (CN) (Newman (2001)), Katz (Katz (1953)) : CN and Katz are pairwise similarity indices for link prediction. CN is a local measure that assigns a similarity score based on the common neighbors of two nodes. Katz index is a global measure that captures the similarity between two nodes by considering paths connecting the nodes. A damping factor β is used to assign higher importance to relatively shorter paths. β is determined by searching over $\{0.005, 0.01, 0.05, 0.1, 0.5\}$ using cross-validation.

7.3.2 Datasets

	Datasets	**# nodes**	**# hyperedges**	**Average hyperedge degree**	**Average node degree**
(a)	Citeseer Co-reference	1299	626	4.610	2.222
(b)	Citeseer Co-citation	1016	817	3.420	2.750
(c)	Cora Co-reference	1961	875	5.259	2.347
(d)	Cora Co-citation	1339	1503	3.060	3.458
(e)	DBLP Co-authorship	4695	2561	5.618	3.064
(f)	Movielens	3893	4677	79.875	95.961
(g)	HiggsTwitter	9948	9605	47.741	46.095
(h)	Amazon Co-view	18565	10839	13.906	8.119
(i)	Amazon Co-purchase	24944	27675	41.759	46.331
(j)	ArnetMiner Co-citation	21375	17300	4.130	3.343
(k)	ArnetMiner Co-reference	16620	26640	4.539	7.275

Table 7.5: Datasets Description: Datasets used to evaluate the HPRA method.

For our experiments, we only use the largest connected component of the network. Statistics of the datasets are shown in Table 7.5.

Cora, Citeseer (Sen *et al.* (2008)) and ArnetMiner (Tang *et al.* (2008)): We built two networks from each dataset; co-citation and co-reference where a node represents a paper. In a co-citation network, a hyperedge connects papers cited together. Similarly, in a co-reference network, if a set of papers refer to the same paper, they are connected by a hyperedge.

HiggsTwitter (De Domenico *et al.* (2013*a*)): This dataset captures messages posted on Twitter about the Higgs boson discovery. We built a social network, where a node represents a person and hyperedge connects all people following the same person.

DBLP (Ley (2002)): This is a co-authorship dataset. Here, a node represents an author and hyperedge connects authors of the paper.

Movielens (Harper and Konstan (2015)): This is a multi-relational dataset, where nodes represent movies and a hyperedge connects movies directed by the same individual.

Amazon Product Metadata (He and McAuley (2016)): We used metadata of products from the video games category and built two networks; co-view and co-purchase. In both networks, nodes represent products. In a co-view network, a hyperedge connects products viewed by customers at the time of purchase. Similarly, in a co-purchase network, a hyperedge connects products purchased together by customers.

	Katz	**CN**	**HPRA**
(a)	0.1346 ± 0.0366	0.1221 ± 0.0259	**0.1449 ± 0.0127**
(b)	0.2570 ± 0.0219	0.2568 ± 0.0170	**0.2949 ± 0.2030**
(c)	0.1199 ± 0.0125	0.1024 ± 0.0177	**0.1303 ± 0.0225**
(d)	0.3644 ± 0.0110	0.3389 ± 0.0058	**0.3866 ± 0.0075**
(e)	0.2480 ± 0.0051	0.2215 ± 0.0073	**0.2855 ± 0.0077**
(f)	0.1050 ± 0.0007	0.1049 ± 0.0008	**0.1215 ± 0.0007**
(g)	0.1472 ± 0.0071	0.1529 ± 0.0046	**0.1921 ± 0.0090**
(h)	0.1290 ± 0.0034	0.1469 ± 0.0072	**0.2218 ± 0.0061**
(i)	0.1405 ± 0.0025	0.1565 ± 0.0033	**0.2234 ± 0.0048**
(j)	0.2256 ± 0.0059	0.2225 ± 0.0056	**0.2495 ± 0.0058**
(k)	0.2676 ± 0.0034	0.2530 ± 0.0031	**0.2895 ± 0.0027**

Table 7.6: Average F1 Scores of HPRA and baselines. First column represents datasets described in Table 7.5.

	CMM	**SHC**	**Katz**	**CN**	**HPRA**
(a)	0.297 ± 0.034	0.588 ± 0.038	0.840 ± 0.081	0.835 ± 0.015	**0.901** ± **0.007**
(b)	0.382 ± 0.071	0.751 ± 0.025	0.883 ± 0.017	0.846 ± 0.014	**0.900** ± **0.015**
(c)	0.407 ± 0.041	0.549 ± 0.017	0.829 ± 0.027	0.788 ± 0.028	**0.851** ± **0.022**
(d)	0.366± 0.006	0.801 ± 0.020	**0.937** ± **0.008**	0.907 ± 0.009	0.923 ± 0.010
(e)	0.072 ± 0.027	0.808 ± 0.030	0.989 ± 0.009	0.981 ± 0.010	**0.990** ± **0.008**
(f)	0.061 ± 0.032	0.658 ± 0.009	0.568 ± 0.097	0.969 ± 0.001	**0.994** ± **0.002**
(g)	-	0.606 ± 0.027	0.476 ± 0.064	0.805 ± 0.012	**0.987** ± **0.002**
(h)	-	0.570 ± 0.012	0.581 ± 0.035	0.986 ± 0.004	**0.990** ± **0.003**
(i)	-	0.594 ± 0.011	0.380 ± 0.144	0.984 ± 0.002	**0.998** ± **0.001**
(j)	-	0.664 ± 0.009	**0.929** ± **0.007**	0.913 ± 0.006	0.924 ± 0.006
(k)	-	0.606 ± 0.006	0.807 ± 0.042	0.908 ± 0.003	**0.942** ± **0.003**

Table 7.7: AUC results of HPRA and aforementioned baselines. First column represents datasets described in Table 7.5. The missing entries correspond to experiments that did not complete even after 24 hours of execution.

	CMM	**SHC**	**Katz**	**CN**	**HPRA**
(a)	0.040 ± 0.025	0.124 ± 0.040	0.690 ± 0.091	0.667 ± 0.060	**0.780** ± **0.131**
(b)	0.150 ± 0.051	0.372 ± 0.031	0.801 ± 0.016	0.734 ± 0.028	**0.830** ± **0.033**
(c)	0.059 ± 0.028	0.119 ± 0.015	0.697 ± 0.037	0.658 ± 0.054	**0.762** ± **0.023**
(d)	0.110 ± 0.039	0.458 ± 0.041	**0.865** ± **0.016**	0.839 ± 0.016	0.859 ± 0.016
(e)	0.007 ± 0.006	0.479 ± 0.053	**0.963** ± **0.019**	0.915 ± 0.024	0.952 ± 0.010
(f)	0.022 ± 0.008	0.445 ± 0.013	0.332 ± 0.077	0.872 ± 0.008	**0.956** ± **0.002**
(g)	-	0.391 ± 0.013	0.310 ± 0.050	0.555 ± 0.013	**0.922** ± **0.005**
(h)	-	0.453 ± 0.014	0.570 ± 0.035	0.956 ± 0.005	**0.970** ± **0.007**
(i)	-	0.416 ± 0.016	0.352 ± 0.136	0.958 ± 0.003	**0.991** ± **0.001**
(j)	-	0.427 ± 0.020	**0.867** ± **0.011**	0.817 ± 0.010	0.839 ± 0.011
(k)	-	0.478 ± 0.007	0.724 ± 0.036	0.648 ± 0.014	**0.858** ± **0.003**

Table 7.8: Precision results of HPRA and aforementioned baselines. First column represents datasets described in Table 7.5. The missing entries correspond to experiments that did not complete even after 24 hours of execution.

7.3.3 Evaluation of HPRA

For experimentation, we randomly divide the hyperedges into two sets: Training set (E^T) and Missing set (E^M). To remove any unwanted bias, we partition the hyperedges into K subsets. Every time we select one subset as E^M and the remaining $K-1$ subsets jointly as E^T. This way, each hyperedge is used for testing exactly once. However, this approach has a limitation that after splitting the hyperedge set, few nodes may not be connected to any other node in the E^T. It is not practical to expect the method to predict hyperedges having such nodes. Therefore, we remove these hyperedges from E^M. Once we have the final E^T and E^M, we generate $|E^M|$ number of new hyperedges by treating E^T as observed hyperedges using *HPRA* and call it as the predicted hyperedges set (E^P). We evaluate the performance of our algorithm by computing the Average F1 score (Yang and Leskovec (2013)).

- **Average F1 Score**: This measure quantifies the closeness of predicted hyperedges to the missing hyperedge set. Average F1 score is the average of the F1-score of the best matching missing hyperedge to each predicted hyperedge and the F1-score of the best-matching predicted hyperedge to each missing hyperedge:

$$Average\ F1\ Score = \frac{1}{2}\Big(\frac{1}{|E^M|}\sum_{e_i \in E^M} F1(e_i, \hat{e}_{g(i)}) + \frac{1}{|E^P|}\sum_{\hat{e}_i \in E^P} F1(e_{g'(i)}, \hat{e}_i)\Big)$$

where $g(i) = argmax_j(F1(e_i, \hat{e}_j))$ and $g'(i) = argmax_j(F1(e_j, \hat{e}_i))$.

To compare *HPRA* with Katz and CN, we use the respective pairwise scores instead of the *HRA* score in our framework. For datasets (a) to (e), we used 5-fold cross validation. For rest of the datasets, we used 10-fold cross validation.

7.3.4 HPRA with a Candidate Hyperedge Set (HPRA-CHS)

To compare the performance against the methods which use a candidate set, we propose a variant of *HPRA*. In *HPRA-CHS*, we select the top $|E^M|$ hyperedges based on *HRA* score as predictions. For a candidate hyperedge, *HRA* score is computed by taking the average of all pairwise ($\frac{m(m-1)}{2}$) *HRA* indices.

Similar to the above setting, for evaluating *HPRA-CHS*, we divide the hyperedges into a Training set (E^T) and Missing set (E^M). We build a candidate set consisting of the missing set E^M and a set of distractor hyperedges. Distractor hyperedges are generated randomly based on the hyperedge degree distribution of the network. In our experiments, the distractor hyperedges set is ten times the size of missing hyperedges set. We generalize the Katz and CN pairwise indices using a method similar to *HRA*. We evaluate our method using two standard metrics, AUC (Table 7.7) and Precision (Table 7.8), similar to (Lü and Zhou (2011); Zhang *et al.* (2018*a*)). In Tables, '-'correspond to experiments that did not complete even after 24 hours of execution on a 64GB, Intel Xeon processor.

- **AUC**: AUC score can be interpreted as the probability that a randomly chosen missing hyperedge is assigned a higher score than a randomly chosen distractor hyperedge.
- **Precision**: Given the rank of the hyperedges in the candidate set, precision is defined as the ratio of actual missing hyperedges to the number of predicted hyperedges. That is to say, if we choose the top L ones (L is the size of missing hyperedges set) as predicted hyperedges, among which L_m hyperedges are in missing hyperedge set, then precision is equal to ($\frac{L_m}{L}$).

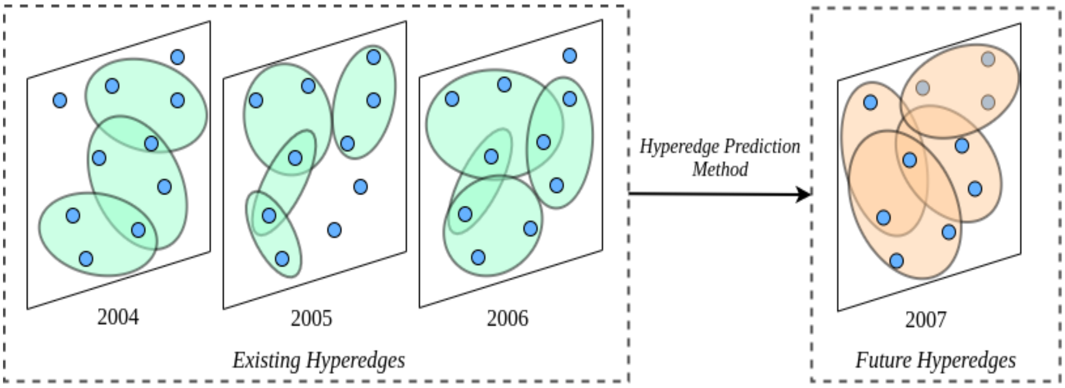


Figure 7.4: An illustration of the temporal dataset. Each graph represents the co-citation hypergraph in the respective year. Here, the task is to predict the hyperedges in the latter hypergraph (2007 in the figure) by using the information from previous years.

	Existing hyperedges	**Future hyperedges**	**# nodes**	**# existing/future hyperedges**	**Avg hyperedge degree**	**Avg node degree**
(1)	2000-2002	2003	10140	20234/6941	6.5527	13.0758
(2)	2001-2003	2004	11827	24018/7574	6.8275	13.8652
(3)	2002-2004	2005	13007	33452/15265	6.4968	16.7089
(4)	2003-2005	2006	16903	45090/17489	6.8317	18.2242
(5)	2004-2006	2007	22143	60265/20007	7.1386	19.4288

Table 7.9: Temporal ACM Cocitation Dataset Description.

	Average F1 score			**AUC**					**Precision**				
	Katz	**CN**	**HPRA**	**CMM**	**SHC**	**Katz**	**CN**	**HPRA**	**CMM**	**SHC**	**Katz**	**CN**	**HPRA**
(1)	0.329	0.302	**0.366**	0.251	0.568	0.933	0.936	**0.953**	0.309	0.612	0.872	0.880	**0.906**
(2)	0.317	0.295	**0.353**	0.241	0.559	0.942	0.944	**0.956**	0.313	0.637	0.875	0.896	**0.917**
(3)	0.348	0.319	**0.386**	-	0.538	0.944	0.945	**0.953**	-	0.621	0.882	0.909	**0.929**
(4)	0.331	0.306	**0.373**	-	0.576	0.954	0.958	**0.972**	-	0.632	0.893	0.923	**0.939**
(5)	0.317	0.293	**0.357**	-	0.581	0.961	0.965	**0.978**	-	0.642	0.900	0.928	**0.939**

Table 7.10: AUC, Precision and Average F1 score results. First column represents datasets described in Table 7.9. The missing entries correspond to experiments that did not complete even after 24 hours of execution.

7.3.5 Future Hyperedge Prediction with HPRA

In this setting, we consider the task of predicting hyperedges of later years by using the previous years' hyperedges, as shown in Figure 7.4. Here, the future hyperedge set consists of hyperedges only from a particular year, and the similarity scores are calculated using network snapshots from previous years. We use ACM co-citation dataset (Tang *et al.* (2008)) for our experiments, and the dataset statistics are shown in

Table 7.9. We evaluate both variants of HPRA- with and without a candidate hyperedge set, and report the results in Table 7.10.

7.3.6 Results and Discussions

From Tables 7.6, 7.7, 7.8 and 7.10, following observations are made:

- *HPRA* outperforms other baselines on most of the datasets and achieves highly competitive performance with the best results on the rest. In order to statistically validate the results, we performed the *Wilcoxon rank-sum test* (also known as the *Mann-Whitney U test)* and observed that in tables 7.7, 7.8 and 7.10, *HPRA* performs significantly ($p < 0.1$) better than all baselines methods.
- None of the other baselines performed consistently well on all datasets, while HPRA is either the best performing or close to the best on all the datasets.
- Though *katz* has a reasonably good performance on many datasets, it fails to perform when hypergraphs have high average hyperedge and node degrees. One possible reason could be, in such hypergraphs, even a small damping factor may involve a large proportion of the hypergraph in score calculation, which may lead to identical similarity scores for multiple node-pairs.
- *HPRA* performs remarkably well on *HiggsTwitter* dataset, while other methods perform poorly. One distinguishing characteristic of this dataset is that the nodes with a low degree are part of hyperedges with high cardinality, and the nodes with a high degree participate in low cardinality hyperedges. This distinct pattern of nodes' participation causes the CN approach to perform poorly. High average node degree and hyperedge cardinality of the hypergraph introduces unwanted influences from a large part of the network, which makes this graph hard for the Katz method.
- We attribute the poor performance of CMM to the way in which its objective function is designed. The CMM objective function is designed in a way that, in the pursuit of optima, it prefers hyperedges of extremely low cardinality over the rest of the hyperedges. If the candidate hyperedge set has distractor hyperedges of low cardinality, then CMM chooses these hyperedges over genuine high cardinality hyperedges. In real-world networks, more often than not, we observe high hyperedge cardinality (refer Tables 7.5 and 7.9).

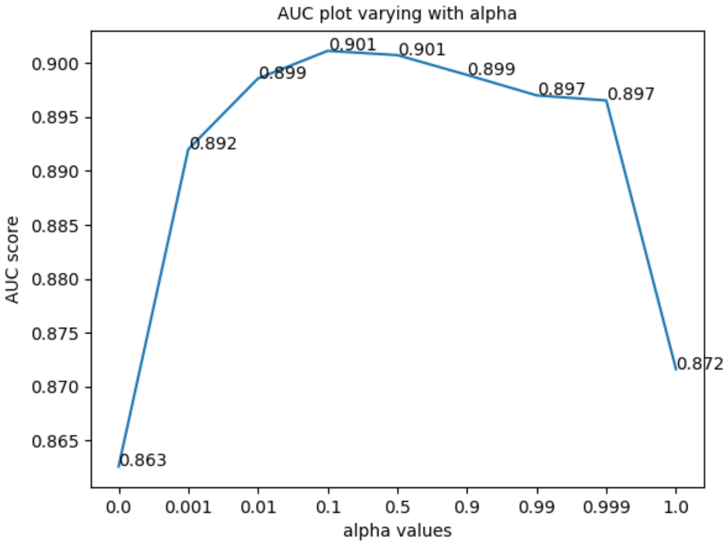
7.3.7 Computational complexity of HPRA

Our proposed algorithm, HPRA, involves computing HRA scores for each node pair and repeated retrieval from it to find the nodes of the new hyperedge. The HRA matrix can be computed with time complexity of $\mathcal{O}(n^2)$. Additionally, finding the cardinality of the hyperedge can be done in constant time. Therefore, the overall time complexity of the HPRA algorithm is $\mathcal{O}(n^2)$. Empirically, for the experiments conducted, it took us less than ten minutes to find the HRA matrix for Amazon co-purchase, the largest hypergraph.

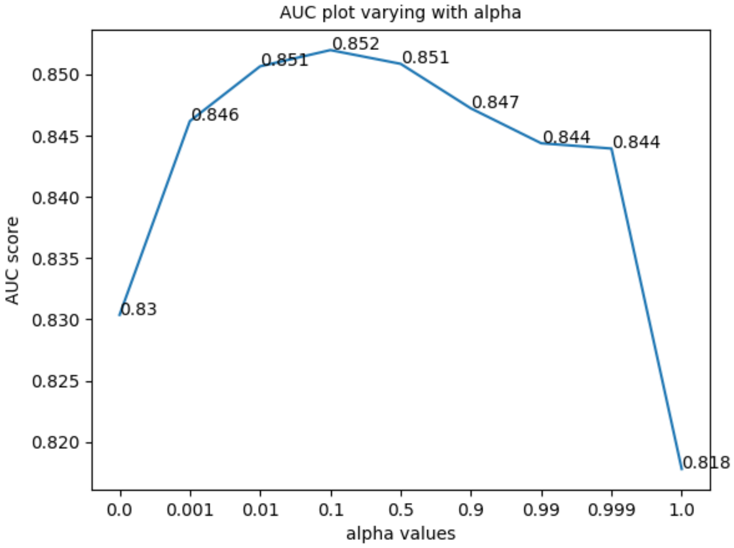
Ablation Study: Our definition of *HRA* has two parts: similarity due to direct connections and due to common neighbors. To analyze the effect of each part, we introduce a weight α, and modify the *HRA* equation as follows:

$$HRA_{xy} = \alpha HRA_{direct} + (1 - \alpha) HRA_{indirect}$$

We vary α over [0,1] to examine the effect of each part on the AUC score (Figure 7.5). We observe low AUC scores at both extremes, which reveals that both parts are essential in precisely predicting the hyperedges.



Citeseer Coreference



Cora Coreference

Figure 7.5: AUC scores vs α. We observe lower performance at both extremes implying that both HRA_{direct} & $HRA_{indirect}$ are essential.

CHAPTER 8

Conclusion

In this work, we explored two complex network structures, multilayer networks, and hypergraphs. We worked on the problems of node centrality in multilayer networks, and clustering and hyperedge prediction in hypergraphs. In multilayer networks and hypergraphs, we present the problems and solutions in contrast with the traditional graph modeling, where the multilayer organization of the system and super-dyadic interactions are not preserved. Our experiments followed by in-depth analysis show the effectiveness of our proposed solutions.

We begin this work by presenting different coupling schemes in multilayer networks focusing on the centrality methods. We take a multi-tissue system as an example where a cross-coupled multilayer network is best suited for modeling. Then we present a set of centrality measures that can capture different aspects of the node effects in a multilayer network. We applied our proposed centrality measures to identify genes involved in tissue-tissue communication. Our results shows that the method carries excellent potential in unwiring inter-tissue communication paths. The proposed centrality measures also show desired theoretical properties such as convergence and decomposability. Our comprehensive analysis of gene rankings revealed that the centrality scores not only helped us in recovering the hormones-producing/responding genes from the existing datasets such as HGv1 but also revealed out-of-ground-truth genes that conform to the existing literature on PubMed. In addition to the protein-encoding genes, our experiments also revealed some long noncoding RNAs (lncRNAs), which are gaining the research community's attention very recently. In the future, our method can be readily applied to understand the genomic-level changes between diseased and healthy populations in systems biology, understand the traffic congestion behavior in multi-mode transport in a city, etc. Our encouraging results also open up directions for further exploration of other centrality measures for complex networks such as multilayer graphs.

In the later parts, we discuss another important complex network structure, hypergraphs. A hypergraph provides a natural representation of the systems that involve super-dyadic interactions among their constituents. In our work, we focus on modeling social networks, collaboration networks, and item-purchase networks as hypergraphs and focus on the problems of hypergraph clustering and hyperedge prediction. In hypergraph clustering, we work with a reduction mechanism that projects a hypergraph to a graph and show that maximizing modularity in the reduced graph and original hypergraph are equivalent. Further, we present a balancing scheme for hyperedge cut that improves the cluster quality in most datasets. Our work on hypergraph clustering can be applied in multiple domains, including VLIS to design chip layout, computer vision for image segmentation, social networks for group identification, etc. Working on a reduced hypergraph opens up directions for other applications, where it is challenging to work in the original hypergraph space, and existing network science tools can be applied after reduction with desired properties. Towards the end, we present, *HPRA*, an algorithm to predict hyperedges in a given hypergraph. The proposed method is the first to predict hyperedges of any cardinality without using any candidate set. Our results show the superiority of the proposed algorithm on a vast range of datasets. In the future, a similar pipeline can be used to predict hyperedges in other domains such as chemical reaction networks, where the principles of social networks may not always hold. All the above-discussed problems open up exciting directions for further exploration of multilayer networks, hypergraphs, and beyond. The universality of the proposed solutions carries the potential of applying them to areas other than our discussions.

REFERENCES

1. **Adamic, L. A.** and **E. Adar** (2003). Friends and neighbors on the web. *Social networks*, **25**(3), 211–230.

2. **Aerts, S.**, **D. Lambrechts**, **S. Maity**, **P. Van Loo**, **B. Coessens**, **F. De Smet**, **L.-C. Tranchevent**, **B. De Moor**, **P. Marynen**, **B. Hassan**, *et al.* (2006). Gene prioritization through genomic data fusion. *Nature biotechnology*, **24**(5), 537–544.

3. **Agarwal, S.**, **K. Branson**, and **S. Belongie**, Higher order learning with graphs. *In ICML'06: Proceedings of the 23rd International Conference on Machine learning*. 2006.

4. **Agarwal, S.**, **J. Lim**, **L. Zelnik-Manor**, **P. Perona**, **D. Kriegman**, and **S. Belongie**, Beyond pairwise clustering. *In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2. 2005. ISSN 1063-6919.

5. **Aleta, A.**, **S. Meloni**, and **Y. Moreno** (2017). A multilayer perspective for the analysis of urban transportation systems. *Scientific reports*, **7**, 44359.

6. **Alves, J.**, **A. Petrosyan**, and **R. Magalhães** (2014). Olfactory dysfunction in dementia. *World J Clin Cases*, **2**(11), 661–667.

7. **Awangga, R. M.**, **M. Yusril**, and **H. Setyawan**, Ontology design of influential people identification using centrality. *In Journal of Physics: Conference Series*, volume 1007-1. IOP Publishing, 2018.

8. **Baas, D.**, **A. Meiniel**, **C. Benadiba**, **E. Bonnafe**, **O. Meiniel**, **W. Reith**, and **B. Durand** (2006). A deficiency in rfx3 causes hydrocephalus associated with abnormal differentiation of ependymal cells. *European Journal of Neuroscience*, **24**(4), 1020–1030.

9. **Basheer, R.**, **M. J. A. Jalal**, and **R. Gomez** (2016). An unusual case of adolescent type 2 diabetes mellitus: Prader–willi syndrome. *Journal of family medicine and primary care*, **5**(1), 181.

10. **Battiston, F.**, **V. Nicosia**, and **V. Latora** (2017). The new challenges of multiplex networks: Measures and models. *The European Physical Journal Special Topics*, **226**(3), 401–416.

11. **Bazzi, M.**, **M. A. Porter**, **S. Williams**, **M. McDonald**, **D. J. Fenn**, and **S. D. Howison** (2016). Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling & Simulation*, **14**(1), 1–41.

12. **Blondel, V. D.**, **J. loup Guillaume**, **R. Lambiotte**, and **E. Lefebvre** (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, **10**, P10008.

13. **Bodine, S. C.**, **H. L. Brooks**, **N. W. Bunnett**, **H. A. Coller**, **M. R. Frey**, **B. Joe**, **T. R. Kleyman**, **M. L. Lindsey**, **A. Marette**, **R. E. Morty**, *et al.* (2021). An american physiological society cross-journal call for papers on "inter-organ communication in homeostasis and disease".

14. **Bolla, M.**, **B. Bullins**, **S. Chaturapruek**, **S. Chen**, and **K. Friedl** (2015). Spectral properties of modularity matrices. *Linear Algebra and Its Applications*, **473**, 359–376.

15. **Brandes, U.**, **D. Delling**, **M. Gaertler**, **R. Görke**, **M. Hoefer**, **Z. Nikoloski**, and **D. Wagner** (2006). Maximizing modularity is hard. *arXiv preprint physics/0608255*.

16. **Bretto, A.**, *Hypergraph Theory: An Introduction*. Springer Publishing Company, 2013. ISBN 3319000799.

17. **Browaeys, R.**, **W. Saelens**, and **Y. Saeys** (2020). Nichenet: modeling intercellular communication by linking ligands to target genes. *Nature methods*, **17**(2), 159–162.

18. **Bulo, S. R.** and **M. Pelillo** (2013). A game-theoretic approach to hypergraph clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(6), 1312–1327. ISSN 0162-8828.

19. **Chan, P.-C.**, **T.-N. Wu**, **Y.-C. Chen**, **C.-H. Lu**, **M. Wabitsch**, **Y.-F. Tian**, and **P.-S. Hsieh** (2018). Targeted inhibition of cd74 attenuates adipose cox-2-mif-mediated m1 macrophage polarization and retards obesity-related adipose tissue inflammation and insulin resistance. *Clinical Science*, **132**(14), 1581–1596.

20. **Chikina, M.**, **E. Zaslavsky**, and **S. C. Sealfon** (2015). CellCODE: a robust latent variable approach to differential expression analysis for heterogeneous cell populations. *Bioinformatics*, **31**(10), 1584–1591.

21. **Chodrow, P.** and **A. Mellor** (2020). Annotated hypergraphs: models and applications. *Applied network science*, **5**, 1–25.

22. **Chodrow, P. S.** (2020). Configuration models of random hypergraphs. *Journal of Complex Networks*, **8**(3), cnaa018.

23. **Chung, F.** and **L. Lu** (2002). Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, **6**(2), 125–145.

24. **Chung, F.** and **W. Zhao**, Pagerank and random walks on graphs. *In Fete of combinatorics and computer science*. Springer, 2010, 43–62.

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

26. **Courtney, O. T.** and **G. Bianconi** (2016). Generalized network structures: The configuration model and the canonical ensemble of simplicial complexes. *Physical Review E*, **93**(6), 062311.

27. **Dastmalchi, N.**, **R. Safaralizadeh**, **S. Latifi-Navid**, **S. M. Banan Khojasteh**, **B. Mahmud Hussen**, and **S. Teimourian** (2021). An updated review of the role of lncrnas and their contribution in various molecular subtypes of breast cancer. *Expert Review of Molecular Diagnostics*, **21**(10), 1025–1036.

28. **De Domenico, M.**, **A. Lima**, **P. Mougel**, and **M. Musolesi** (2013*a*). The anatomy of a scientific rumor. *Scientific reports*, **3**, 2980.

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

30. **De Domenico, M.**, **A. Solé-Ribalta**, **E. Omodei**, **S. Gómez**, and **A. Arenas** (2015). Ranking in interconnected multilayer networks reveals versatile nodes. *Nature communications*, **6**, 6868.

31. **del Rio, G.**, **D. Koschützki**, and **G. Coello** (2009). How to identify essential genes from molecular networks? *BMC systems biology*, **3**(1), 1–12.

32. **Ding, C.** and **X. He**, Cluster merging and splitting in hierarchical clustering algorithms. *In 2002 IEEE International Conference on Data Mining, 2002. Proceedings.*. IEEE, 2002.

33. **Dolev, S.**, **Y. Elovici**, and **R. Puzis** (2010). Routing betweenness centrality. *Journal of the ACM (JACM)*, **57**(4), 1–27.

34. **Droujinine, I.** and **N. Perrimon** (2013). Defining the interorgan communication network: systemic coordination of organismal cellular processes under homeostasis and localized stress. *Frontiers in cellular and infection microbiology*, **3**, 82.

35. **Droujinine, I. A.**, **A. S. Meyer**, **D. Wang**, **N. D. Udeshi**, **Y. Hu**, **D. Rocco**, **J. A. McMahon**, **R. Yang**, **J. Guo**, **L. Mu**, *et al.* (2021). Proteomics of protein trafficking by in vivo tissue-specific labeling. *Nature communications*, **12**(1), 1–22.

36. **Du, Y.**, **N. Wei**, **R. Ma**, **S.-H. Jiang**, and **D. Song** (2020). Long noncoding rna mir210hg promotes the warburg effect and tumor growth by enhancing hif-1α translation in triple-negative breast cancer. *Frontiers in oncology*, **10**.

37. **Ducournau, A.** and **A. Bretto** (2014). Random walks in directed hypergraphs and application to semi-supervised image segmentation. *Computer Vision and Image Understanding*, **120**, 91–102.

38. **Erdos, P.**, **A. Rényi**, *et al.* (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, **5**(1), 17–60.

39. **Estrada, E.** and **J. A. Rodriguez-Velazquez** (2005). Complex networks as hypergraphs. *arXiv preprint physics/0505137*.

40. **Fasino, D.** and **F. Tudisco** (2016). Generalized modularity matrices. *Linear Algebra and its Applications*, **502**, 327–345.

41. **Feng, F.**, **X. He**, **Y. Liu**, **L. Nie**, and **T.-S. Chua**, Learning on partial-order hypergraphs. *In Proceedings of the 2018 World Wide Web Conference*, WWW '18. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2018. ISBN 978-1-4503-5639-8.

42. **Fitzgerald, P. J.** (2009). Is norepinephrine an etiological factor in some types of cancer? *International journal of cancer*, **124**(2), 257–263.

43. **Gallotti, R.** and **M. Barthelemy** (2015). The multilayer temporal network of public transport in great britain. *Scientific data*, **2**, 140056.

44. **Georgiadis, L.**, **M. J. Neely**, **L. Tassiulas**, *et al.* (2006). Resource allocation and cross-layer control in wireless networks. *Foundations and Trends® in Networking*, **1**(1), 1–144.

45. **Ghosal, S.**, **B. Zhu**, **T.-T. Huynh**, **L. Meuter**, **A. Jha**, **S. Talvacchio**, **M. Knue**, **M. Patel**, **T. Prodanov**, **S. Das**, *et al.* (2021). A long noncoding rna–microrna expression signature predicts metastatic signature in pheochromocytomas and paragangliomas. *Endocrine*, 1–10.

46. **Giusti, C.**, **R. Ghrist**, and **D. S. Bassett** (2016). Two's company, three (or more) is a simplex. *Journal of computational neuroscience*, **41**(1), 1–14.

47. **Gleich, D. F.** (2015). Pagerank beyond the web. *SIAM Review*, **57**(3), 321–363.

48. **Gomez, S.**, **A. Diaz-Guilera**, **J. Gomez-Gardenes**, **C. J. Perez-Vicente**, **Y. Moreno**, and **A. Arenas** (2013). Diffusion dynamics on multiplex networks. *Physical review letters*, **110**(2), 028701.

49. **Greene, D.**, **G. Sheridan**, **B. Smyth**, and **P. Cunningham**, Aggregating content and network information to curate twitter user lists. *In Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web*, RSWeb '12. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1638-5.

50. **Group, G. D.**, **H. Bilo**, **L. Coentrão**, **C. Couchoud**, **A. Covic**, **J. De Sutter**, **C. Drechsler**, **L. Gnudi**, **D. Goldsmith**, **J. Heaf**, *et al.* (2015). Clinical practice guideline on management of patients with diabetes and chronic kidney disease stage 3b or higher (egfr< 45 ml/min). *Nephrology Dialysis Transplantation*, **30**(suppl_2), ii1–ii142.

51. **Gu, L.**, **H. Sun**, and **Z. Yan** (2020). lncrna zeb1-as1 is downregulated in diabetic lung and regulates lung cell apoptosis. *Experimental and Therapeutic Medicine*, **20**(6), 1–1.

52. **Guo, J.-L.**, **Q. Suo**, **A.-Z. Shen**, and **J. Forrest** (2016*a*). The evolution of hyperedge cardinalities and bose-einstein condensation in hypernetworks. *Scientific reports*, **6**, 33651.

53. **Guo, J.-L.**, **X.-Y. Zhu**, **Q. Suo**, and **J. Forrest** (2016*b*). Non-uniform evolving hypergraphs and weighted evolving hypergraphs. *Scientific reports*, **6**, 36648.

54. **Habibi, M.**, **L. Weber**, **M. Neves**, **D. L. Wiegandt**, and **U. Leser** (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, **33**(14), i37–i48.

55. **Hadley, S. W.**, **B. L. Mark**, and **A. Vannelli** (1992). An efficient eigenvector approach for finding netlist partitions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **11**(7), 885–892.

56. **Halu, A.**, **M. De Domenico**, **A. Arenas**, and **A. Sharma** (2019). The multiplex network of human diseases. *NPJ systems biology and applications*, **5**(1), 15.

57. **Hamers, L.** *et al.* (1989). Similarity measures in scientometric research: The jaccard index versus salton's cosine formula. *Information Processing and Management*, **25**(3), 315–18.

58. **Harper, F. M.** and **J. A. Konstan** (2015). The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, **5**(4), 19:1–19:19. ISSN 2160-6455. URL `http://doi.acm.org/10.1145/2827872`.

59. **He, R.** and **J. McAuley**, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *In Proceedings of the 25th international conference on world wide web*. 2016.

60. **Higham, N. J.** (1990). Exploiting fast matrix multiplication within the level 3 blas. *ACM Transactions on Mathematical Software (TOMS)*, **16**(4), 352–368.

61. **Holme, P.** (2003). Congestion and centrality in traffic flow on complex networks. *Advances in Complex Systems*, **6**(02), 163–176.

62. **Hristova, D.**, **M. J. Williams**, **M. Musolesi**, **P. Panzarasa**, and **C. Mascolo**, Measuring urban social diversity using interconnected geo-social networks. *In Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016.

63. **Hsing, A. W.**, **Y.-T. Gao**, **S. Chua Jr**, **J. Deng**, and **F. Z. Stanczyk** (2003). Insulin resistance and prostate cancer risk. *Journal of the National Cancer Institute*, **95**(1), 67–71.

64. **Huang, Z.** and **A. Xu** (2021). Adipose extracellular vesicles in intercellular and inter-organ crosstalk in metabolic health and diseases. *Frontiers in Immunology*, **12**, 463.

65. **Iacovacci, J.**, **C. Rahmede**, **A. Arenas**, and **G. Bianconi** (2016). Functional multiplex pagerank. *EPL (Europhysics Letters)*, **116**(2), 28004.

66. **Jadhav, A.**, **T. Kumar**, **M. Raghavendra**, **T. Loganathan**, and **M. Narayanan** (2022). Predicting cross-tissue hormone–gene relations using balanced word embeddings. *Bioinformatics*, **38**(20), 4771–4781.

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

68. **Jin, L.**, **C. Luo**, **X. Wu**, **M. Li**, **S. Wu**, and **Y. Feng** (2021). Lncrna-haglr motivates triple negative breast cancer progression by regulation of wnt2 via sponging mir-335-3p. *Aging (Albany NY)*, **13**(15), 19306.

69. **Kamiński, B.**, **V. Poulin**, **P. Prałat**, **P. Szufel**, and **F. Théberge** (2019). Clustering via hypergraph modularity. *PloS one*, **14**(11).

70. **Karypis, G.**, **R. Aggarwal**, **V. Kumar**, and **S. Shekhar** (1999). Multilevel hypergraph partitioning: Applications in vlsi domain. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **7**(1), 69–79.

71. **Karypis, G.** and **V. Kumar** (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, **20**(1), 359–392. ISSN 1064-8275.

72. **Katz, L.** (1953). A new status index derived from sociometric analysis. *Psychometrika*, **18**(1), 39–43. ISSN 1860-0980.

73. **Kim, H.** and **R. Anderson** (2012). Temporal node centrality in complex networks. *Physical Review E*, **85**(2), 026107.

74. **Kim, S.**, **S. Nowozin**, **P. Kohli**, and **C. D. Yoo**, Higher-order correlation clustering for image segmentation. *In Advances in Neural Information Processing Systems*. 2011.

75. **Klamt, S.**, **U.-U. Haus**, and **F. Theis** (2009). Hypergraphs and cellular networks. *PLoS computational biology*, **5**(5).

76. **Kolosov, N.**, **M. J. Daly**, and **M. Artomov** (2021). Prioritization of disease genes from gwas using ensemble-based positive-unlabeled learning. *European Journal of Human Genetics*, 1–9.

77. **Kumar, A.**, **L. Xie**, **C. M. Ta**, **A. O. Hinton**, **S. K. Gunasekar**, **R. A. Minerath**, **K. Shen**, **J. M. Maurer**, **C. E. Grueter**, **E. D. Abel**, *et al.* (2020). Swell1 regulates skeletal muscle cell size, intracellular signaling, adiposity and glucose metabolism. *Elife*, **9**, e58941.

78. **Kunegis, J.**, **M. Blattner**, and **C. Moser**, Preferential attachment in online networks: Measurement and explanations. *In Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13. ACM, New York, NY, USA, 2013. ISBN 9781450318891. URL `https://doi.org/10.1145/2464464.2464514`.

79. **Kwon, H. J.**, **M. I. Park**, **S. J. Park**, **W. Moon**, **S. E. Kim**, **J. H. Kim**, **Y. J. Choi**, and **S. K. Lee** (2019). Insulin resistance is associated with early gastric cancer: a prospective multicenter case control study. *Gut and liver*, **13**(2), 154.

80. **Lange, M.** (2021). Analysis of single nucleotide polymorphisms in regulatory elements of oncogenic lncrnas. *Master's Thesis*.

81. **Langfelder, P.** and **S. Horvath** (2008). Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, **9**(1), 1–13.

82. **Leordeanu, M.** and **C. Sminchisescu**, Efficient hypergraph clustering. *In* **N. D. Lawrence** and **M. Girolami** (eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*. PMLR, La Palma, Canary Islands, 2012.

83. **Ley, M.**, The dblp computer science bibliography: Evolution, research issues, perspectives. *In International symposium on string processing and information retrieval*. Springer, 2002.

84. **Li, D.**, **J. Lu**, **H. Li**, **S. Qi**, and **L. Yu** (2019*a*). Identification of a long noncoding rna signature to predict outcomes of glioblastoma. *Molecular medicine reports*, **19**(6), 5406–5416.

85. **Li, N.** and **X. Zhan** (2019). Identification of clinical trait–related lncrna and mrna biomarkers with weighted gene co-expression network analysis as useful tool for personalized medicine in ovarian cancer. *EPMA Journal*, **10**(3), 273–290.

86. **Li, W.**, **H. Li**, **L. Zhang**, **M. Hu**, **F. Li**, **J. Deng**, **M. An**, **S. Wu**, **R. Ma**, **J. Lu**, *et al.* (2017). Long non-coding rna linc00672 contributes to p53 protein-mediated gene suppression and promotes endometrial cancer chemosensitivity. *Journal of Biological Chemistry*, **292**(14), 5801–5813.

87. **Li, X.**, **F. Jin**, and **Y. Li** (2021). A novel autophagy-related lncrna prognostic risk model for breast cancer. *Journal of Cellular and Molecular Medicine*, **25**(1), 4–14.

88. **Li, X.** and **H. Yu** (2020). Overexpression of hoxa-as2 inhibits inflammation and apoptosis in podocytes via sponging mirna-302b-3p to upregulate timp3. *Eur. Rev. Med. Pharmacol. Sci*, **24**, 4963–4970.

89. **Li, X.-y.**, **L.-y. Zhou**, **H. Luo**, **Q. Zhu**, **L. Zuo**, **G.-y. Liu**, **C. Feng**, **J.-y. Zhao**, **Y.-y. Zhang**, and **X. Li** (2019*b*). The long noncoding rna mir210hg promotes tumor metastasis by acting as a cerna of mir-1226-3p to regulate mucin-1c expression in invasive breast cancer. *Aging (Albany NY)*, **11**(15), 5646.

90. **Lian, Y.**, **Z. Li**, **Y. Fan**, **Q. Huang**, **J. Chen**, **W. Liu**, **C. Xiao**, and **H. Xu** (2017). The lncrna-hoxa-as2/ezh2/lsd1 oncogene complex promotes cell proliferation in pancreatic cancer. *American journal of translational research*, **9**(12), 5496.

91. **Liao, H.**, **M. S. Mariani**, **M. Medo**, **Y.-C. Zhang**, and **M.-Y. Zhou** (2017). Ranking in evolving complex networks. *Physics Reports*, **689**, 1–54.

92. **Lin, D.**, An information-theoretic definition of similarity. *In Proceedings of the Fifteenth ICML*, ICML '98. Morgan Kaufmann Publishers Inc., 1998. ISBN 1-55860-556-8.

93. **Lin, Z.**, **X. Li**, **X. Zhan**, **L. Sun**, **J. Gao**, **Y. Cao**, and **H. Qiu** (2017). Construction of competitive endogenous rna network reveals regulatory role of long non-coding rnas in type 2 diabetes mellitus. *Journal of cellular and molecular medicine*, **21**(12), 3204–3213.

94. **Liu, D.-R.** and **M.-Y. Wu** (2001). A hypergraph based approach to declustering problems. *Distributed and Parallel Databases*, **10**(3), 269–288.

95. **Liu, G.-M.**, **H.-D. Zeng**, **C.-Y. Zhang**, and **J.-W. Xu** (2019). Key genes associated with diabetes mellitus and hepatocellular carcinoma. *Pathology-Research and Practice*, **215**(11), 152510.

96. **Liu, H.**, **L. J. Latecki**, and **S. Yan**, Robust clustering as ensembles of affinity relations. *In Advances in Neural Information Processing Systems*. 2010.

97. **Liu, S.**, **P.-Y. Chen**, **A. Hero**, and **I. Rajapakse** (2018). Dynamic network analysis of the 4d nucleome. *bioRxiv*, 268318.

98. **Lonsdale, J.**, **J. Thomas**, **M. Salvatore**, **R. Phillips**, **E. Lo**, **S. Shad**, **R. Hasz**, **G. Walters**, **F. Garcia**, **N. Young**, *et al.* (2013). The genotype-tissue expression (gtex) project. *Nature genetics*, **45**(6), 580–585.

99. **López-Cortés, A.**, **C. Paz-y Miño**, **A. Cabrera-Andrade**, **S. J. Barigye**, **C. R. Munteanu**, **H. González-Díaz**, **A. Pazos**, **Y. Pérez-Castillo**, and **E. Tejera** (2018). Gene prioritization, communality analysis, networking and metabolic integrated pathway to better understand breast cancer pathogenesis. *Scientific reports*, **8**(1), 1–15.

100. **Louis, A.**, Hypergraph markov operators, eigenvalues and approximation algorithms. *In Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, STOC '15. ACM, New York, NY, USA, 2015. ISBN 978-1-4503-3536-2.

101. **Lü, L.** and **T. Zhou** (2011). Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, **390**(6), 1150–1170.

102. **Lung, R. I.**, **N. Gaskó**, and **M. A. Suciu** (2018). A hypergraph model for representing scientific output. *Scientometrics*, **117**(3), 1361–1379.

103. **Malek, M.**, **S. Zorzan**, and **M. Ghoniem** (2020). A methodology for multilayer networks analysis in the context of open and private data: biological application. *Applied Network Science*, **5**(1), 1–28.

104. **McKenzie, A. T.**, **M. Wang**, **M. E. Hauberg**, **J. F. Fullard**, **A. Kozlenkov**, **A. Keenan**, **Y. L. Hurd**, **S. Dracheva**, **P. Casaccia**, **P. Roussos**, and **B. Zhang** (2018). Brain Cell Type Specific Gene Expression and Co-expression Network Architectures. *Sci Rep*, **8**(1), 8868.

105. **McPherson, M.**, **L. Smith-Lovin**, and **J. M. Cook** (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, **27**(1), 415–444.

106. **Meng, Q.**, **X. Zhai**, **Y. Yuan**, **Q. Ji**, and **P. Zhang** (2020). lncrna zeb1-as1 inhibits high glucose-induced emt and fibrogenesis by regulating the mir-216a-5p/bmp7 axis in diabetic nephropathy. *Brazilian Journal of Medical and Biological Research*, **53**(4).

107. **Meyer, C. D.**, *Matrix analysis and applied linear algebra*, volume 71. SIAM, 2000.

108. **Mistry, D.**, **R. P. Wise**, and **J. A. Dickerson** (2017). Diffslc: A graph centrality method to detect essential proteins of a protein-protein interaction network. *PloS one*, **12**(11), e0187091.

109. **Moreau, Y.** and **L.-C. Tranchevent** (2012). Computational tools for prioritizing candidate genes: boosting disease gene discovery. *Nature Reviews Genetics*, **13**(8), 523–536.

110. **Nagai, T.** and **M. Mori** (1999). Prader-willi syndrome, diabetes mellitus and hypogonadism. *Biomedicine & pharmacotherapy*, **53**(10), 452–454.

111. **Newman, M.**, *Networks*. Oxford university press, 2018.

112. **Newman, M. E.** (2001). Clustering and preferential attachment in growing networks. *Physical review E*, **64**(2), 025102.

113. **Newman, M. E.** (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, **103**(23), 8577–8582.

114. **Newman, M. E.**, *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010. ISBN 0199206651, 9780199206650.

115. **Ng, A. Y.**, **M. I. Jordan**, and **Y. Weiss**, On spectral clustering: Analysis and an algorithm. *In Advances in neural information processing systems*. 2002.

116. **Ou, J. R.**, **M. S. Tan**, **A. M. Xie**, **J. T. Yu**, and **L. Tan** (2014). Heat shock protein 90 in Alzheimer's disease. *Biomed Res Int*, **2014**, 796869.

117. **Page, L.**, **S. Brin**, **R. Motwani**, and **T. Winograd** (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

118. **Pan, X.**, **D. Li**, **J. Huo**, **F. Kong**, **H. Yang**, and **X. Ma** (2018). Linc01016 promotes the malignant phenotype of endometrial cancer cells by regulating the mir-302a-3p/mir-3130-3p/nfya/satb1 axis. *Cell death & disease*, **9**(3), 1–18.

119. **Pilosof, S.**, **M. A. Porter**, **M. Pascual**, and **S. Kéfi** (2017). The multilayer nature of ecological networks. *Nature Ecology & Evolution*, **1**(4), 0101.

120. **Rand, W. M.** (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, **66**(336), 846–850.

121. **Reaven, G. M.** (1980). Insulin-independent diabetes mellitus: metabolic characteristics. *Metabolism*, **29**(5), 445–454.

122. **Rothzerg, E.**, **X. D. Ho**, **J. Xu**, **D. Wood**, **A. Märtson**, and **S. Kõks** (2021). Upregulation of 15 antisense long non-coding rnas in osteosarcoma. *Genes*, **12**(8), 1132.

123. **Saito, S.**, **D. Mandic**, and **H. Suzuki**, Hypergraph p-laplacian: A differential geometry view. *In AAAI Conference on Artificial Intelligence*. 2018.

124. **Sarkar, P.**, **D. Chakrabarti**, and **A. W. Moore**, Theoretical justification of popular link prediction heuristics. *In Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.

125. **Satchidanand, S. N.**, **H. Ananthapadmanaban**, and **B. Ravindran**, Extended discriminative random walk: a hypergraph approach to multi-view multi-relational transductive learning. *In Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

126. **Satchidanand, S. N.**, **S. K. Jain**, **A. Maurya**, and **B. Ravindran**, Studying indian railways network using hypergraphs. *In 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2014.

127. **Schaeffer, S. E.** (2007). Graph clustering. *Computer science review*, **1**(1), 27–64.

128. **Schlicker, A.**, **T. Lengauer**, and **M. Albrecht** (2010). Improving disease gene prioritization using the semantic similarity of gene ontology terms. *Bioinformatics*, **26**(18), i561–i567.

129. **Schoen, R. E.**, **C. M. Tangen**, **L. H. Kuller**, **G. L. Burke**, **M. Cushman**, **R. P. Tracy**, **A. Dobs**, and **P. J. Savage** (1999). Increased blood glucose and insulin, body size, and incident colorectal cancer. *Journal of the National Cancer Institute*, **91**(13), 1147–1154.

130. **Schütze, H.**, **C. D. Manning**, and **P. Raghavan**, *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

131. **Seldin, M. M.**, **S. Koplev**, **P. Rajbhandari**, **L. Vergnes**, **G. M. Rosenberg**, **Y. Meng**, **C. Pan**, **T. M. Phuong**, **R. Gharakhanian**, **N. Che**, *et al.* (2018). A strategy for discovery of endocrine interactions with application to whole-body metabolism. *Cell metabolism*, **27**(5), 1138–1155.

132. **Sen, P.**, **G. Namata**, **M. Bilgic**, **L. Getoor**, **B. Galligher**, and **T. Eliassi-Rad** (2008). Collective classification in network data. *AI Magazine*, **29**(3), 93.

133. **Shamir, A.**, A survey on mesh segmentation techniques. *In Computer graphics forum*, volume 27-6. Wiley Online Library, 2008.

134. **Sharma, A.**, **J. Srivastava**, and **A. Chandra** (2014). Predicting multi-actor collaborations using hypergraphs. *arXiv preprint arXiv:1401.6404*.

135. **Shashua, A.**, **R. Zass**, and **T. Hazan**, Multi-way clustering using super-symmetric non-negative tensor factorization. *In Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*, ECCV'06. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 3-540-33838-1, 978-3-540-33838-3.

136. **Shelton, L. B.**, **J. D. Baker**, **D. Zheng**, **L. E. Sullivan**, **P. K. Solanki**, **J. M. Webster**, **Z. Sun**, **J. J. Sabbagh**, **B. A. Nordhues**, **J. Koren**, **S. Ghosh**, **B. S. J. Blagg**, **L. J. Blair**, and **C. A. Dickey** (2017). Hsp90 activator Aha1 drives production of pathological tau aggregates. *Proc Natl Acad Sci U S A*, **114**(36), 9707–9712.

137. **Shi, J.** and **J. Malik** (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, **22**(8), 888–905.

138. **Shinde, P.** and **S. Jalan** (2015). A multilayer protein-protein interaction network analysis of different life stages in caenorhabditis elegans. *EPL (Europhysics Letters)*, **112**(5), 58001.

139. **Sideris, G.**, **D. Katsaros**, **A. Sidiropoulos**, and **Y. Manolopoulos**, The science of science and a multilayer network approach to scientists' ranking. *In Proceedings of the 22nd International Database Engineering & Applications Symposium*. ACM, 2018.

140. **Solá, L.**, **M. Romance**, **R. Criado**, **J. Flores**, **A. García del Amo**, and **S. Boccaletti** (2013). Eigenvector centrality of nodes in multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **23**(3), 033131.

141. **Solé-Ribalta, A.**, **M. De Domenico**, **S. Gómez**, and **A. Arenas** (2016*a*). Random walk centrality in interconnected multilayer networks. *Physica D: Nonlinear Phenomena*, **323**, 73–79.

142. **Solé-Ribalta, A.**, **S. Gómez**, and **A. Arenas** (2016*b*). Congestion induced by the structure of multiplex networks. *Physical review letters*, **116**(10), 108701.

143. **Song, Y.**, **C. Miao**, and **J. Wang** (2019). Lncrna zeb1-as1 inhibits renal fibrosis in diabetic nephropathy by regulating the mir-217/mafb axis. *RSC advances*, **9**(52), 30389–30397.

144. **Sorz, J.**, **B. Wallner**, **H. Seidler**, and **M. Fieder** (2015). Inconsistent year-to-year fluctuations limit the conclusiveness of global higher education rankings for university management. *PeerJ*, **3**, e1217.

145. **Stuhlmann, T.**, **R. Planells-Cases**, and **T. J. Jentsch** (2018). Lrrc8/vrac anion channels enhance β-cell glucose sensing and insulin secretion. *Nature communications*, **9**(1), 1–12.

146. **Su, F.**, **S. Yang**, **H. Wang**, **Z. Qiao**, **H. Zhao**, and **Z. Qu** (2020). CIRBP Ameliorates Neuronal Amyloid Toxicity via Antioxidative and Antiapoptotic Pathways in Primary Cortical Neurons. *Oxid Med Cell Longev*, **2020**, 2786139.

147. **Sun, M.** and **W. L. Kraus** (2015). From discovery to function: the expanding roles of long noncoding rnas in physiology and disease. *Endocrine reviews*, **36**(1), 25–64.

148. **Sun, Q.**, **Y. J. Song**, and **K. V. Prasanth** (2021). One locus with two roles: microrna-independent functions of microrna-host-gene locus-encoded long noncoding rnas. *Wiley Interdisciplinary Reviews: RNA*, **12**(3), e1625.

149. **Tang, J.**, **J. Ren**, **Q. Cui**, **D. Zhang**, **D. Kong**, **X. Liao**, **M. Lu**, **Y. Gong**, and **G. Wu** (2019). A prognostic 10-lncrna expression signature for predicting the risk of tumour recurrence in breast cancer patients. *Journal of cellular and molecular medicine*, **23**(10), 6775–6784.

150. **Tang, J.**, **J. Zhang**, **L. Yao**, **J. Li**, **L. Zhang**, and **Z. Su**, Arnetminer: Extraction and mining of academic social networks. *In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08. Association for Computing Machinery, New York, NY, USA, 2008. ISBN 9781605581934.

151. **Taylor, D.**, **S. A. Myers**, **A. Clauset**, **M. A. Porter**, and **P. J. Mucha** (2017). Eigenvector-based centrality measures for temporal networks. *Multiscale Modeling & Simulation*, **15**(1), 537–574.

152. **Timmons, J. A.**, **P. J. Atherton**, **O. Larsson**, **S. Sood**, **I. O. Blokhin**, **R. J. Brogan**, **C.-H. Volmar**, **A. R. Josse**, **C. Slentz**, **C. Wahlestedt**, *et al.* (2018). A coding and non-coding transcriptomic perspective on the genomics of human metabolic disease. *Nucleic acids research*, **46**(15), 7772–7792.

153. **Traag, V. A.** (2015). Faster unfolding of communities: Speeding up the louvain algorithm. *Physical Review E*, **92**(3), 032801.

154. **Trabert, B.**, **M. E. Sherman**, **N. Kannan**, and **F. Z. Stanczyk** (2020). Progesterone and breast cancer. *Endocrine reviews*, **41**(2), 320–344.

155. **Trefethen, L. N.** and **D. Bau**, *Numerical linear algebra*, volume 181. SIAM, 2022.

156. **Türker, İ.** and **E. E. Sulak** (2018). A multilayer network analysis of hashtags in twitter via co-occurrence and semantic links. *International Journal of Modern Physics B*, **32**(04), 1850029.

157. **Valcarce, D.**, **J. Parapar**, and **Á. Barreiro**, Additive smoothing for relevance-based language modelling of recommender systems. *In Proceedings of the 4th CERI*. 2016.

158. **Vestergaard, H.**, **H. H. Klein**, **T. Hansen**, **J. Müller**, **F. Skovby**, **C. Bjørbæk**, **M. Røder**, **O. Pedersen**, *et al.* (1995). Severe insulin-resistant diabetes mellitus in patients with congenital muscle fiber type disproportion myopathy. *The Journal of clinical investigation*, **95**(4), 1925–1932.

159. **Volejnikova, J.**, **P. Vojta**, **H. Urbankova**, **R. Mojzíkova**, **M. Horvathova**, **I. Hochova**, **J. Cermak**, **J. Blatny**, **M. Sukova**, **E. Bubanska**, *et al.* (2020). Czech and slovak diamond-blackfan anemia (dba) registry update: Clinical data and novel causative genetic lesions. *Blood Cells, Molecules, and Diseases*, **81**, 102380.

160. **Wang, M.**, **N. D. Beckmann**, **P. Roussos**, **E. Wang**, **X. Zhou**, **Q. Wang**, **C. Ming**, **R. Neff**, **W. Ma**, **J. F. Fullard**, **M. E. Hauberg**, **J. Bendl**, **M. A. Peters**, **B. Logsdon**, **P. Wang**, **M. Mahajan**, **L. M. Mangravite**, **E. B. Dammer**, **D. M. Duong**, **J. J. Lah**, **N. T. Seyfried**, **A. I. Levey**, **J. D. Buxbaum**, **M. Ehrlich**, **S. Gandy**, **P. Katsel**, **V. Haroutunian**, **E. Schadt**, and **B. Zhang** (2018*a*). The Mount Sinai cohort of large-scale genomic, transcriptomic and proteomic data in Alzheimer's disease. *Sci Data*, **5**, 180185.

161. **Wang, M.**, **N. D. Beckmann**, **P. Roussos**, **E. Wang**, **X. Zhou**, **Q. Wang**, **C. Ming**, **R. Neff**, **W. Ma**, **J. F. Fullard**, *et al.* (2018*b*). The mount sinai cohort of large-scale

genomic, transcriptomic and proteomic data in alzheimer's disease. *Scientific data*, **5**(1), 1–16.

162. **Wen, M.-H.**, **H.-P. Hsiao**, **M.-C. Chao**, and **F.-J. Tsai** (2010). Growth hormone deficiency in a case of crouzon syndrome with hydrocephalus. *International journal of pediatric endocrinology*, **2010**, 1–4.

163. **Yan, T.**, **F. Ding**, and **Y. Zhao** (2019). Integrated identification of key genes and pathways in Alzheimer's disease via comprehensive bioinformatical analyses. *Hereditas*, **156**, 25.

164. **Yang, J.** and **J. Leskovec**, Defining and evaluating network communities based on ground-truth. *In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1546-3.

165. **Yang, J.** and **J. Leskovec**, Overlapping community detection at scale: a nonnegative matrix factorization approach. *In Proceedings of the sixth ACM international conference on Web search and data mining*. 2013.

166. **Yang, S.**, **Y. Zhou**, **X. Zhang**, **L. Wang**, **J. Fu**, **X. Zhao**, and **L. Yang** (2021). The prognostic value of an autophagy-related lncrna signature in hepatocellular carcinoma. *BMC bioinformatics*, **22**(1), 1–16.

167. **Yang, X.**, **F. Liu**, **Z. Xu**, **C. Chen**, **G. Li**, **X. Wu**, and **J. Li** (2004). Growth hormone receptor expression in human colorectal cancer. *Digestive diseases and sciences*, **49**(9), 1493–1498.

168. **Ye, H.**, **B. Adane**, **N. Khan**, **E. Alexeev**, **N. Nusbacher**, **M. Minhajuddin**, **B. M. Stevens**, **A. C. Winters**, **X. Lin**, **J. M. Ashton**, *et al.* (2018). Subversion of systemic glucose metabolism as a mechanism to support the growth of leukemia cells. *Cancer Cell*, **34**(4), 659–673.

169. **Yin, R.-R.**, **Q. Guo**, **J.-N. Yang**, and **J.-G. Liu** (2018). Inter-layer similarity-based eigenvector centrality measures for temporal networks. *Physica A: Statistical Mechanics and its Applications*, **512**, 165–173.

170. **Yoon, S.-e.**, **H. Song**, **K. Shin**, and **Y. Yi**, How much and when do we need higher-order information in hypergraphs? a case study on hyperedge prediction. *In Proceedings of The Web Conference 2020*. 2020.

171. **Young, J.-G.**, **G. Petri**, **F. Vaccarino**, and **A. Patania** (2017). Construction of and efficient sampling from the simplicial configuration model. *Physical Review E*, **96**(3), 032312.

172. **Zhang, M.**, **Z. Cui**, **S. Jiang**, and **Y. Chen**, Beyond link prediction: Predicting hyperlinks in adjacency space. *In Thirty-Second AAAI Conference on Artificial Intelligence*. 2018*a*.

173. **Zhang, Q.**, **C. Ma**, **M. Gearing**, **P. G. Wang**, **L. S. Chin**, and **L. Li** (2018*b*). Integrated proteomics and network analysis identifies protein hubs and network alterations in Alzheimer's disease. *Acta Neuropathol Commun*, **6**(1), 19.

174. **Zhang, Y.**, **Q. Chen**, **Z. Yang**, **H. Lin**, and **Z. Lu** (2019). BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data*, **6**(52).

175. **Zhang, Y.**, **X. You**, **S. Li**, **Q. Long**, **Y. Zhu**, **Z. Teng**, and **Y. Zeng** (2020). Peripheral blood leukocyte rna-seq identifies a set of genes related to abnormal psychomotor behavior characteristics in patients with schizophrenia. *Medical science monitor: international medical journal of experimental and clinical research*, **26**, e922426–1.

176. **Zhao, X.**, **N. Wang**, **H. Shi**, **H. Wan**, **J. Huang**, and **Y. Gao**, Hypergraph learning with cost interval optimization. *In AAAI Conference on Artificial Intelligence*. 2018.

177. **Zhou, D.**, **J. Huang**, and **B. Schölkopf**, Learning with hypergraphs: Clustering, classification, and embedding. *In Advances in Neural Information Processing Systems*. 2007.

178. **Zhou, T.**, **L. Lü**, and **Y.-C. Zhang** (2009). Predicting missing links via local information. *The European Physical Journal B*, **71**(4), 623–630.

179. **Zitnik, M.**, **S. Rok Sosic**, and **J. Leskovec** (2018). Biosnap datasets: Stanford biomedical network dataset collection. *Note: http://snap. stanford. edu/biodata Cited by*, **5**(1).

LIST OF PAPERS BASED ON THESIS

1. **REFEREED JOURNALS BASED ON THE THESIS**

    (a) “Effect of Inter-layer Coupling on Multilayer Network Centrality Measures.” Journal of the Indian Institute of Science, Kumar T., Narayanan M., and Ravindran B., (2019), Vol 99, pages 237 - 246, DOI: 10.1007/s41745-019-0103-y

    (b) “Hypergraph Clustering by Iteratively Reweighted Modularity Maximization." Applied Network Science, Kumar T., Sankaran V., Harini A., Parthasarathy S., and Ravindran B. (2020), Vol 5, DOI: 10.1007/s41109-020-00300-3

    (c) “MultiCens: Multilayer network centrality measures to uncover molecular mediators of tissue-tissue communication.” PLOS Computational Biology, Kumar, T., Sethuraman, R., Mitra, S., Ravindran, B., and Narayanan, M. (2023). bioRxiv. DOI: 10.1101/2022.05.15.492007

2. **CONFERENCE PROCEEDINGS**

    (a) “A new measure of modularity in hypergraphs: Theoretical insights and implications for effective clustering.” In International Conference on Complex Networks and Their Applications (pp. 286-297), Kumar, T.*, Vaidyanathan, S.*, Ananthapadmanabhan, H., Parthasarathy, S., and Ravindran, B. (2019, December). Springer, Cham. DOI: 10.1007/978-3-030-36687-2_24

    (b) “HPRA: Hyperedge prediction using resource allocation.” In 12th ACM conference on web science (pp. 135-143). Kumar, T.*, Darwin, K.*, Parthasarathy, S., and Ravindran, B. (2020, July). DOI: 10.1145/3394231.3397903

CURRICULUM VITAE

1. NAME : Tarun Kumar

2. DATE OF BIRTH : 25 Aug 1991

3. EDUCATIONAL QUALIFICATIONS

2013 Bachelor of Technology (B.Tech)

Institution : Punjab Technical University

Specialization : Computer Science and Engineering

2015 Master of Technology (M.Tech)

Institution : Shiv Nadar University

Specialization : Computer Science and Engineering

Doctor of Philosophy (Ph.D.)

Institution : Indian Institute of Technology Madras

Specialization : Computer Science & Engineering

Registration Date : 6 Jan 2016

DOCTORAL COMITTEE

CHAIRPERSON : Dr. NARAYANASWAMY N S

Professor

Department of Computer Science & Engineering

GUIDES : Dr. Balaraman Ravindran

Professor

Department of Computer Science & Engineering

Dr. Manikandan Narayanan

Associate Professor

Department of Computer Science & Engineering

MEMBERS : Dr. Meghana Nasre

Associate Professor

Department of Computer Science & Engineering

Dr. Rupesh Nasre

Associate Professor

Department of Computer Science & Engineering

Dr. Venkatesh Ramaiyan

Assistant Professor

Department of Electrical Engineering