

Adapting Community Detection Approaches to Large, Multilayer, and Attributed Networks

Natalie Stanley

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the
Curriculum in Bioinformatics and Computational Biology.

Chapel Hill
2018

Approved by:

Peter Mucha

Marc Niethammer

Jeremy Purvis

Tamara Berg

David Gotz

Laura Miller

©201X
Your M. Name
ALL RIGHTS RESERVED

ABSTRACT

Natalie Stanley: Adapting community detection approaches to large, multilayer, and attributed
networks
(Under the direction of Professor Peter J. Mucha)

Networks have become a common data mining tool to encode relational definitions between a set of entities. Whether studying correlations between genes based on their expression patterns, or communication between individuals in a social network, network analysis tools enable interpretation, prediction, and visualization of patterns in the data. Community detection is well-developed subfield of network analysis, where the objective is to cluster nodes into ‘communities’ based on their connectivity patterns with the rest of the network. There are many well developed and robust approaches for identifying communities in a single moderately-sized network but the ability to work with more challenging types of networks that contain either extra or a large amount of information pose challenges. In this thesis, we address 3 types of challenging network data and how to adapt standard community detection approaches to handle these situations. First, we present a method for identifying communities in multilayer networks, where there exist multiple relational definitions between nodes. Next, we provide a pre-processing technique for reducing the size of large networks, where standard community detection approaches might have inconsistent results or be prohibitively slow. We then introduce an extension to a probabilistic model for community structure to take into account node attribute information. Finally, we show how community detection can be solve applied problems in computational biology and social science.

Dedication...

ACKNOWLEDGEMENTS

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

TABLE OF CONTENTS

LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xix
1 Introduction	1
1.1 Network Notation, Representation, and Summarization	2
1.1.1 Representing relational information	2
1.1.2 Network Summary Statistics	3
1.1.2.1 Case Study: Computing network summary statistics on a network built from single cell data	3
1.1.2.2 Degree Distribution	4
1.1.2.3 Centrality and network hubs.....	6
1.2 Introduction to community detection	8
1.3 Community detection methods	9
1.3.1 Quality function maximization with modularity	9
1.3.2 Identifying communities with probabilistic approaches	10
1.3.2.1 Probabilistic graphical models for statistical inference	11
1.3.2.2 Stochastic Block Model	12
1.3.2.3 Variants to the Classic Stochastic Block Model	15
1.3.2.4 Affiliation model and inference	17
1.3.3 Deep Learning Approaches	18
1.3.4 Spectral community detection methods	19
1.3.5 Higher order network analysis.....	19

1.4	Community detection in computational biology	19
1.4.1	Immunological profiling to establish a pregnancy immune clock	19
1.4.2	Uncovering differences in microbiome community structure in patients with inflammatory bowel disease	20
1.4.3	Community detection for analysis of flow cytometry data	21
1.4.4	Understanding genetic diversity of the malaria parasite genes	22
1.4.5	Analysis of high dimensional single cell data for tumor heterogeneity	24
1.4.6	Identification of virulence factor genes related to antibiotic resistance of uropathogenic <i>E. coli</i>	25
1.5	Challenging problems in community detection	26
1.5.1	Temporal Networks	26
1.5.2	Multilayer networks	26
1.5.3	Network Comparison	26
1.5.4	Large Networks	26
1.5.5	Attributed Networks	26
1.6	Thesis Contribution	26
1.6.1	Relevant Publications	26
1.6.2	Software	27
1.6.3	Thesis Statement	28
2	A multilayer stochastic block model	29
2.1	Introduction to multilayer networks	29
2.2	Comparing network layers based on community structure	30
2.3	Related work in community detection of multilayer networks	32
2.4	A Summary of Novel Contributions of sMLSBM	34
2.5	sMLSBM Model Definition	35
2.6	Inference for learning model parameters of sMLSBM	36
2.7	Synthetic Examples	42
2.7.1	Comparison of sMLSBM to other SBM Approaches	42

2.7.2	Synthetic Experiment with Two Strata	44
2.8	Human Microbiome Project Example	47
2.8.1	Comparison of sMLSBM to multilayer network reducibility	48
2.8.2	Generating samples from the fitted sMLSBM	49
2.9	Concluding remarks for sMLSBM	50
2.10	Detectability in a single stratum	52
2.10.1	Investigating detectability in a multilayer network	53
2.10.2	Studying detectability in two block networks	54
2.10.3	Using random matrix theory to study detectability	54
3	Network compression for community detection with super nodes.....	56
3.1	Super pixel pre-processing of images	56
3.2	Super node pre-processing for networks	57
3.2.1	Problem Formulation	58
3.2.2	An opportunity for super nodes in community detection	58
3.3	Background	59
3.3.1	Related Work	59
3.3.2	Validation metrics for a quality super node representation	60
3.3.2.1	Objectively Comparing Partitions on Possibly Different Scales	62
3.4	Methods	63
3.4.1	Defining seeds	63
3.4.2	Grow Super Nodes Around Seeds	64
3.4.3	Create Network of Super Nodes	65
3.5	Results	65
3.5.1	Overview of experiments	66
3.5.2	Normalized mutual information and under segmentation error	67
3.5.3	Run time Analysis	68
3.5.4	Quantifying variability across algorithm runs	70

3.5.5	Neighborhood agreement	71
3.6	Conclusion and Future Work.....	74
4	An attributed stochastic block model.....	75
4.1	Motivation for studying attributed networks	75
4.2	An Attributed Stochastic Block Model	78
4.2.1	Objective	78
4.2.2	Attribute Likelihood	79
4.2.3	Adjacency Matrix Likelihood	79
4.2.4	Inference	80
4.2.5	Initialization	81
4.3	Synthetic Data Results.....	81
4.3.1	Detectability analysis	83
4.4	Using the fitted attributed SBM for link prediction and collaborative filtering	84
4.4.1	Link Prediction Methods	84
4.4.2	Collaborative Filtering Methods	85
4.5	Applications in Biological Networks	87
4.5.1	Microbiome Subject Similarity Results	87
4.5.1.1	Motivation	87
4.5.1.2	Pre-Processing	87
4.5.1.3	Constructing Node Attributes.....	88
4.5.1.4	Link Prediction Experiments	88
4.5.1.5	Collaborative Filtering Experiments	88
4.5.2	Protein Interaction Network Results	89
4.5.2.1	Data Pre-Processing	92
4.5.2.2	Constructing Node Attributes.....	92
4.5.2.3	Link prediction experiments	93
4.5.2.4	Collaborative filtering experiments	94

4.6	Conclusion and Future Work.....	95
5	Community detection for understanding burn inhalation injury	97
6	Conclusion and future work	98
	BIBLIOGRAPHY	99

LIST OF TABLES

3.1 Network data characteristics.	65
--	----

LIST OF FIGURES

1.1	Toy social network. A small example of a social network, with nodes being users and edges representing connections between users. Image from https://www.phpfox.com	2
1.2	Hairball network. Networks are often noisy data structures and lack an immediate straight forward structural interpretation. Image from https://cs.umd.edu	4
1.3	Network of single cells. We constructed a network from mass cytometry profiling in single cell data. Each node is a single cell and is connected to its 5 nearest neighbors.	4
1.4	Degree distribution for single cell network. We visualize the trends in node degree in the single cell network presented in Figure 1.3. A. We compute a cumulative distribution plot for degree. B. Node degrees can also be visualized with a simple histogram.	5
1.5	Centralities on single cell network. The second order ego network for the highest centrality nodes in the single cell network according to degree, betweenness, and eigenvector in the left, right, and center, respectively.	6
1.6	Assortative Community Structure. Nodes are tightly connected to each other and more sparsely connected to the rest of the network. Each community is outlined with a pink dotted line.	8
1.7	A comparison of k-means and the Louvain algorithm A comparison of the results of clustering the single cell dataset by visualizing the original 50 dimensional data with a 2-dimensional projection with tSNE. Points are colored by their cluster membership under k -means on the original data (left) and Louvain community detection (right) on the constructed 5 nearest neighbor network.	11
1.8	Directed Acyclic Graph. A directed acyclic graph (DAG) is formed based on dependency between random variable and allows for a fully factorized probability distribution.....	12
1.9	SBM Graphical Model. A graphical model is used to model the dependency between the node-to-community assignments, \mathbf{z} and the observed network adjacency matrix, \mathbf{A}	13

2.1	Objective of strata multilayer stochastic block model (sMLSBM). Each of the $L = 9$ networks here represents a layer in a multilayer network. Every network layer has $N = 36$ nodes that are consistent across all layers. There are $S = 3$ strata as indicated by the three rows and the colors of nodes. Clearly, network layers within a stratum exhibit strong similarities in community structure. That is, although each layer follows an SBM with $K = 3$ communities, the SBM parameters are identical for layers within a stratum but differ between layers in different strata. We would like to partition the layers into their appropriate strata and learn their associated SBM parameters, π^s and Z^s	35
2.2	Schematic illustration of our algorithm: Our algorithm for fitting an sMLSBM is broken up into two phases: an initialization phase to cluster layers into strata, and an iterative phase that allows learning of node-to-community and layer-to-strata assignments.	38
2.3	Synthetic experiment comparing sMLSBM to other SBMs. A. We specified a model with $S = 3$ strata and $L = 10$ layers per stratum. A representative layer from each stratum is plotted. Note that nodes in all networks are colored according to their community membership in stratum 1. Each network has $N = 128$ nodes, $K = 4$ communities and mean degree, $c = 20$. The p_{in}^s parameters for $s = 1, 2$ and 3 are $0.6, 0.4$ and 0.25 , respectively. Corresponding values of p_{out}^s were selected to maintain the desired expected mean degree, $c=20$. B. We fit 3 types of models to the 30 network layers: i) single SBM: fitting a single SBM to all of the layers; ii) single-Layer SBM: fitting an individual SBM to each layer; and iii) sMLSBM: identifying strata and fitting an SBMs for each strata. Each model yields an estimate $\bar{\pi}^{s_l}$ for the true SBM of each layer l , which is denoted π^l . Here s_l denotes the inferred strata for layer l . On the vertical axis we plot the mean ℓ_2 norm error $\ \text{vec}(\pi^l) - \text{vec}(\bar{\pi}^{s_l})\ _2$. C. For each of the three models, we computed the normalized mutual information (NMI) between the true node-to-community assignments z^l and the inferred values \bar{z}^{s_l}	43

2.4 Synthetic experiment with two strata. We conducted numerical experiments with multilayer networks with $N = 128$ nodes, mean degree $c = 16$, $S = 2$ strata and $K^1 = K^2 = 4$ communities. The networks contained either $L = 10$ (left column) or $L = 100$ layers (right column), which were divided equally into the two strata. For stratum 1, we fixed the quantity $N(p_{in}^1 - p_{out}^1) = 10$, which fully specifies (p_{in}^1, p_{out}^1) since setting $c = 16$ also constrains these parameters. In contrast, we vary $N(p_{in}^2 - p_{out}^2)$. A. As a function of $N(p_{in}^2 - p_{out}^2)$, we plot the mean NMI to interpret the ability of sMLSBM to recover the true layer-to-strata assignments. We compare the performance of sMLSBM (purple curve) to generic k -means clustering (green symbols) of adjacency matrices. B. We plot the mean number of iterations (NOI) required for Phase II of our algorithm to converge. C. Finally, we measure the quality of node-to-community assignment results by plotting the mean NMI between the true node-to-community assignments and those inferred with sMLSBM in stratum 1 (red symbols) and stratum 2 (blue symbols).	46
2.5 Comparison of sMLSBM on the OTU interaction networks (45) for each of the body sites to a reducibility hierarchy (34). As described in the text, we consider a multiplex network with $L = 18$ layers and $N = 213$ nodes, which we group here into $S = 6$ strata, while the dendrogram was generated by the method employed as the precursor to the reducibility framework. Colored boxes around the leaves of the dendrogram designate the body site to strata assignments obtained with sMLSBM.	49
2.6 Visualization of Strata in SparCC Networks. We visualize the adjacency matrices for SparCC networks that encode microbiome interactions at body sites. In each panel, a colored dot at position (i, j) indicates the existence of an edge (i, j) in the corresponding network layer. The four rows correspond to four different strata. In column 1, we show a sample network generated from the SBM parameters, π^s and \bar{Z}^s , that we inferred for that stratum. In Columns 2 and 3, we show SparCC networks from that particular stratum. Note the strong similarity across each row.	51
3.1 Superpixel pre-processing of an image. An image can be represented by a 1147×1147 grid of pixels (left). Representing the image with 600 super pixels (right), reduces the size of the image and hence the segmentation problem is to partition the set of 600 super pixels.	57
3.2 Defining super nodes. To define the super node representation of a network, we select S seeds and agglomerate local regions around them to create super nodes. This then leads to a new network with weighted edges between the S super nodes upon which community detection can be more efficiently applied.	62

3.3	Choosing seeds in a synthetic network. The identification of 20 seeds with the CoreHD algorithm in a network generated from a stochastic block model with 8 communities. Seeds (black nodes) are well distributed across communities.	64
3.4	Schematic of possible partition comparisons. We outline the types of possible comparisons between partitions generated according to various combinations of network representation and community detection method. According to these comparison rules, we compute normalized mutual information (NMI) between all pairs of networks satisfying the comparison criteria. The colored circles in the schematic represent a single partition generated under the corresponding network representation and community detection algorithm combination. Circles are colored (in each column) by each of the four possible representation/community detection method combinations. In A-C , we outline the types of comparisons we perform in subsequent figures. A. To compare the usefulness of the super node representation in identifying communities retrieved using the full network, we compare pairs of networks with different representations under the same community detection algorithm. B. Due to the stochastic nature of both the Louvain algorithm and SBM fitting, this comparison seeks to quantify partitions generated under the same network representation and method. C. Finally, we consider pairs of partitions generated under the same network representation and different community detection algorithms.	67
3.5	Super Node Quality. We computed normalized mutual information (A.) and under segmentation error (B.) for networks represented by between 100 and 600 super nodes. Line type and color indicate the community detection algorithm applied (Louvain algorithm or SBM fitting). Each curve indicates the mean across 5 super node representations. The shaded area shows standard deviation. A. Normalized mutual information between the full and super node representations of networks [i.e. $\text{NMI}(\mathbf{z}^{\text{Full}}, \mathbf{z}^{SN})$]. A network representation with more super nodes. generally increases the NMI between full network and super node network representations. Horizontal lines indicate the mean pairwise NMI between 10 runs of the Louvain algorithm and SBM result on the full network (pink and gold, respectively). Given the high variability between multiple runs of the same algorithm on the full network, adding more super nodes can only improve the NMI between the full and super node representation to the observed level of similarity observed between algorithm runs. B. The log under segmentation error for super node representations. Defining a super node representation with more super nodes generally decreases the under segmentation error.	69
3.6	Runtimes. We compare community detection runtimes (in seconds) with the Louvain algorithm and by fitting an SBM on the full networks and super node representations for the 9 data sets. (A.) Louvain on the full network. (B.) Louvain on the super nodes. (C.) SBM on the full network. (D.) SBM on the super nodes.	70

3.7 Quantifying partition variability. For each of the 9 networks, we obtained 10 different partitions by the Louvain algorithm and 10 different SBM fits under the default (A.) and matched settings (B.). To assess the similarity between partitions within and between a community detection algorithm in networks under the super node representation, we computed pairwise normalized mutual information (NMI) as a function of the number of super nodes. The pink and blue curves show the mean pairwise normalized mutual information between all pairs of 10 partitions under Louvain and SBM fitting, respectively. The gold curves compare pairs of partitions under different methods. Shaded area denotes standard deviation. Horizontal lines indicates the mean pairwise NMI between partitions under the full network representation for within Louvain and SBM partition comparison (pink and blue, respectively) and between Louvain and SBM partition comparison (gold). Overall, the super node representation is useful for reducing the disparity between the partitions obtained under different methods.	71
3.8 Agreement of community assignments with local connectivity. We study how consistent partitions are within local neighborhood regions of the network by examining how well a node's neighbors (for various order neighborhoods) can be used to predict its community assignment, under some community partition \mathbf{z} . For each community in a partition, we give a binary prediction of whether a node is assigned to that community, based on probabilities we compute for a node from its neighbors. Sweeping the parameter p that sets the probability required for a node to be assigned to a community, we compute ROC curves for each community and report the minimum AUC value observed. Panels A-D show minimum AUC values observed as a function of neighborhood order for communities obtained from the full networks and super node representations by Louvain and by SBM. Line color indicates network and line type indicates communities obtained from the matched and default parameters used by the algorithms on the full networks. Panels E-H visualize the communities obtained in the As22 data on the full network (default parameters) and super node representation (SN) under Louvain and SBM, with node colors indicating community memberships.	73
4.1 Modeling community membership in terms of attributes and connectivity. Node-to-community assignments specified by \mathbf{Z} are determined in terms of adjacency matrix information, \mathbf{A} and attribute matrix information, \mathbf{X} . \mathbf{A} and \mathbf{X} are assumed to be generated from a stochastic block model and a mixture of multivariate Gaussian distributions, parameterized by θ and Ψ , respectively.	78

4.2	Synthetic Example. We generated a synthetic network with $N = 200$ nodes, $K = 4$ communities and an 8-dimensional multivariate Gaussian for each community. A. A visualization of the adjacency matrix for this network where a black dot indicates an edge. We observe that there is an assortative block structure (blocks on the diagonal), but there are also many 'noisy' edges between communities making the true community structure with only a stochastic block model a bit harder to detect. B. We performed PCA on the $N \times p$ dimensional attribute array and plotted each of the N nodes in two dimensions. Points are colored by their true community assignments, \mathbf{z} . Clustering the nodes according to only connectivity, only attributes, and with the attributed SBM, we quantified the partition accuracy with normalized mutual information. This gave results $\text{NMI}(\mathbf{z}, \{\mathbf{z}^{\text{connectivity}}, \mathbf{z}^{\text{attributes}}, \mathbf{z}^{\text{attribute sbm}}\}) = \{0.65, 0.68, 0.83\}$	82
4.3	Detectability Analysis in Synthetic Example. To understand how attribute information can be combined with connectivity to assign nodes to communities accurately, we generated synthetic networks for within-probabilities of p_{in} between 0.05 and 0.3 with corresponding p_{out} or between-community probabilities such that the mean degree of the network was 20. For each of these synthetic networks, we used the attributes from the analysis in figure 2 to fit the attributed SBM. Here, we plot the correctness of the node-to-community assignment with normalized mutual information using the partition obtained from regular SBM (blue) and the partition under the attributed SBM model fit (pink). For each combination of p_{in} and p_{out} , we generated 10 networks and hence the bands around the points denote standard deviation. Incorporating attributes with the attributes stochastic block model shifts the detectability limit slightly to the left.	84
4.4	Microbiome subject similarity network: A visualization of the 121 node microbiome subject similarity network with nodes colored by the partition using the classic (A.) and attributed (B.) stochastic block model. A. Fitting the classic stochastic block model to the network, 7 communities were identified. B. Fitting the attributed stochastic block model to the network with the attributes being the first 5 principle components of each subject's OTU count vector (metagenomic profile), 6 communities were identified. Incorporating attributes in inferring this partition removed some of the noise in the partition on the network, specifically in the mixed purple community in the left of A.	89
4.5	Link Prediction on the microbiome subject similarity network: The results for link prediction on the microbiome subject similarity network for the attributed SBM, Jaccard, Adamic-Adar and preferential attachment methods. The corresponding AUC values for these methods, respectively are, 0.71, 0.69, 0.69, and 0.62.	90

4.6	Collaborative Filtering Accuracy in Microbiome Subject Similarity Network: For each of the 121 nodes, we fit a model to the remaining 120 node network and given the node's closest neighbors (based on network connectivity) sought to predict its 5-dimensional attribute vector. The reported error is the relative error \mathcal{E} between the difference between the true attribute vector (\mathbf{x}_i) and its predicted attribute vector ($\hat{\mathbf{x}}_i$). The mean error in \mathbf{x}_i is 0.21, as opposed to the neighbor average and weighted neighbor averages, having errors of 0.26 and 0.27, respectively.	91
4.7	Protein interaction network. We visualize the 82 node protein interaction network under the classic stochastic block model A , and the attributed stochastic block model B . In both networks, nodes are colored by their community assignment and the node shape indicates whether the modification status increased (square) or decreased. A . Nodes colored according to the community partition under the stochastic block model. Nodes are assigned to one of five communities. B . Nodes are colored to the community partition under one of nine communities.	91
4.8	Community entropies in the protein interaction network. We studied the entropy of the 2 class and 6 class classifications of the nodes in A , and B , respectively under the classic SBM (black) and attributed SBM (purple) partitions. For A . – B . the horizontal axis denotes the community index for the particular partition. Nodes belonged to 1 of 5 communities under the classic SBM and belong to 1 of 9 communities with the attributed SBM. Incorporating attributes under both classifications succeeds in breaking up a high entropy community (5) from the classic SBM partition to lower entropy communities in the attributed SBM partition.	93
4.9	Link Prediction in the protein interaction network. Performing link prediction using the attributed SBM, Jaccard, Adamic Adar, and preferential attachment. The corresponding AUC curves for these methods were 0.61, 0.58, 0.58, and 0.51, respectively.	94
4.10	Collaborative filtering in the protein interaction network. For each of the 82 nodes, we fit a model to the remaining 81 node network and given the node's closest neighbors (based on network connectivity) sought to predict its 6-dimensional attribute vector. The reported error is the relative error \mathcal{E} between the difference between the true attribute vector (\mathbf{x}_i) and its predicted attribute vector ($\hat{\mathbf{x}}_i$). The mean error in \mathbf{x}_i using the attributed SBM is 0.21, as opposed to the neighbor average error where it is 0.48.	95

LIST OF ABBREVIATIONS

ABD	All But Dissertation
I/O	Input/Output
IPC	Inter-Process Communication
IPI	Inter-Processor Interrupt
WSS	Working Set Size
AYO	Add Your Own in alphabetic order...

CHAPTER 1

Introduction

Network data appears widely across fields as a data structure for modeling relational information between a set of entities. In recent years, networks have become an indispensable data mining tool, as they allow for tasks such as, data visualization, clustering, and predictive modeling. Motivated by problems in fields, such as, biology, medicine, neuroscience, social science, and epidemiology, the field of network analysis has gained popularity and seeks to develop tools for understanding the associated network data. The main objectives in creating tools for the analysis of network data is to enable effective modeling, prediction, and date interpretation. In this thesis, we present three new methods that enable a more thorough understanding of the structural organization patterns in network through community detection. The objective of community detection is to partition the network nodes into *communities*, such that members of a community have similar connectivity patterns. With an increasing amount of more challenging types of network data, such as those containing multiple relational definitions between a set of nodes can not always be properly addressed using standard community detection approaches. In this thesis, we will look in depth at how to handle communities in networks that are *multilayer*, *large*, and *attributed*. We then present several case studies in each of the developed methods in applications such as, microbiome analysis, protein interaction network understanding, and mining in social networks. We show that the successful identification of communities in these types of networks allows the network to be used for tasks such as, visualization, prediction, and classification.

In this introduction, we first present notation, terminology, and useful concepts for talking about networks. We then provide a detailed discussion about community detection, highlighting not only the main methods studied in this thesis, but also novel and state-of-the-art approaches. Next, we provide several examples of how community detection is used as an important tool in computational



Figure 1.1: **Toy social network.** A small example of a social network, with nodes being users and edges representing connections between users. Image from <https://www.phpfox.com>

biology, as it assists in tasks such as, clustering and biological interpretation. Finally, we put the work in this thesis in the context of important open problems in community detection.

1.1 Network Notation, Representation, and Summarization

1.1.1 Representing relational information

Humans frequently benefit from network applications for tasks such as, viewing relevant queries from a google search, enjoying a suggested movie on Netflix, or interacting on a social network platform. The basic building blocks of networks are nodes, representing entities in a systems, and edges, encoding connections their physical or inferred connection or similarity. Figure 1.1 shows a social network between 7 users and edges between them denoting whether they interact.

Such a network with edges simply representing whether or not a pair of nodes interact is an example of an *undirected, unweighted* network. We will use an undirected network to introduce two forms of representations for networks. For a set of N nodes, we define the $N \times N$ network adjacency matrix, $\mathbf{A} = \{a_{ij}\}$. For a pair of nodes i and j , its corresponding adjacency matrix entry a_{ij} is defined as follows,

$$\begin{cases} a_{ij} = 1 & \text{if node } i \text{ and node } j \text{ are connected} \\ a_{ij} = 0 & \text{otherwise.} \end{cases}$$

Undirected networks can also be *weighted*, where the weight of an edge between a node pair encodes their extent of similarity. These edge weights are some real number and are frequently quantities such as correlation or pairwise similarity. A simple extension of \mathbf{A} to an undirected, weighted network where w is the edge weight between nodes i and j computes the adjacency matrix entry a_{ij} as,

$$\begin{cases} a_{ij} = w & \text{if node } i \text{ and node } j \text{ are connected with weight } w \\ a_{ij} = 0 & \text{otherwise.} \end{cases}$$

Alternatively, the assumption of a symmetric relationship between a pair of nodes that node i connects to node j and node j connects to node i may be unrealistic. For example, on twitter, user i can follow user j , but user j does not necessarily need to follow user i . This type of network is known as a *directed* network. While directed are frequently discussed in the network science literature, we will not introduce them here.

1.1.2 Network Summary Statistics

Given a network, there are fundamental tasks of interest that allow for a more clear interpretation and understanding of the data. Some of these objectives include, quantifying node importance, quantifying edge density, identifying connected components ,clustering nodes, and predicting links. Networks in textbooks often look deceptively clean and well-structure. In reality, most network data is described as being a hairball. This term refers to the difficulty of discerning structure or interpreting meaning from the network based on the connectivity patterns. An example of a typical hairball is shown in figure 1.2

1.1.2.1 Case Study: Computing network summary statistics on a network built from single cell data

Such a challenging representation of the data requires breaking the network down into smaller pieces that can be further analyzed. In this section, we will describe several summary statistics and analyses that can be performed and will be seen throughout the rest of this thesis.

To illustrate these quantities, we will compute them on the network shown in Figure 1.3. This

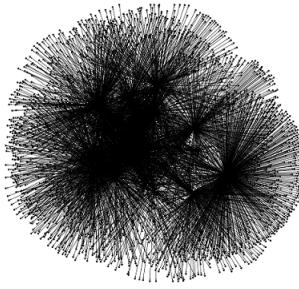


Figure 1.2: **Hairball network.** Networks are often noisy data structures and lack an immediate straight forward structural interpretation. Image from <https://cs.umd.edu>

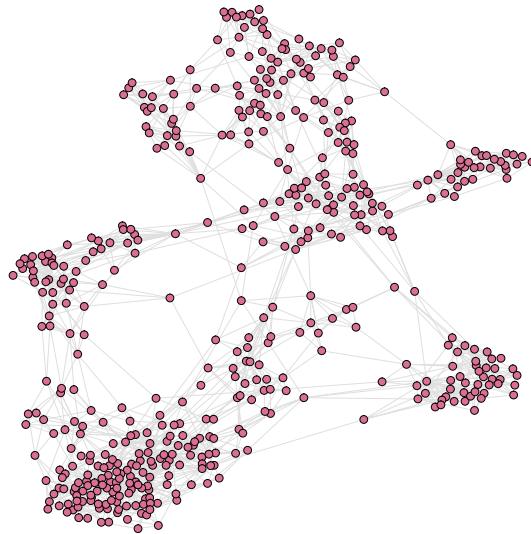


Figure 1.3: **Network of single cells.** We constructed a network from mass cytometry profiling in single cell data. Each node is a single cell and is connected to its 5 nearest neighbors.

network is constructed from a single cell mass cytometry dataset released publicly in the Cytofkit R package (26). Each node represents a single cell and is represented with 50 features profiled with mass cytometry. We created a network from this dataset by selecting 500 cells and building a k -nearest neighbor network with $k = 5$. This means that for a node i , we found its 5 nearest neighbors according to Euclidean distance, and connected them all to node i .

1.1.2.2 Degree Distribution

The first most basic summary statistic is known as *degree*. Here, we will define a variety of summary statistics and quantities that can be computed on a network that give insight into the network's

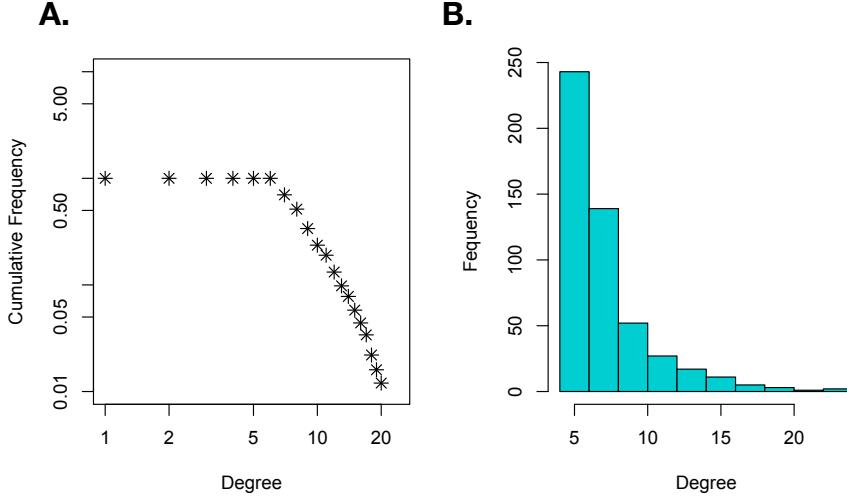


Figure 1.4: Degree distribution for single cell network. We visualize the trends in node degree in the single cell network presented in Figure 1.3. **A.** We compute a cumulative distribution plot for degree. **B.** Node degrees can also be visualized with a simple histogram.

structure. Given the adjacency matrix for an undirected network, **A**, the degree of node i , $\text{degree}(i)$ is computed as,

$$\text{degree}(i) = \sum_j a_{ij} \quad (1.1)$$

We visualize degree distribution in 1.4 using a cumulative distribution plot (A.) and a simple histogram (B.). Since this network was constructed with a 5-nearest neighbor rule, we see this reflected in the degree distribution, with all nodes having degree 5 or more. A few nodes have significantly higher degree (> 10) and represent single cells who is a nearest neighbor to many of the other cells in the original 50 dimensional space.

In the case of an undirected, unweighted network, the degree of node i counts its number of neighbors, while in the undirected, weighted context, degree encodes the total edge weight incident to node i . Collectively examining the distribution of degrees for a network is known as the *degree distribution*. Understanding the degree distribution provides insight into the network type and structural organization. A node's degree is often highly related to its importance in the network, which provides a nice transition to the next set of summary statistics: centrality and hub scores.

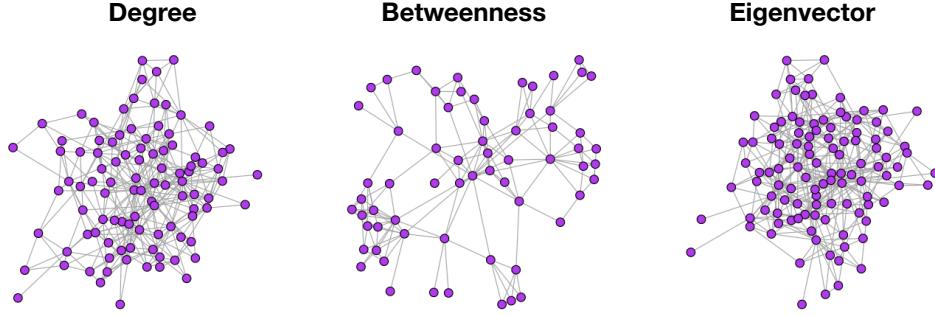


Figure 1.5: **Centralities on single cell network.** The second order ego network for the highest centrality nodes in the single cell network according to degree, betweenness, and eigenvector in the left, right, and center, respectively.

1.1.2.3 Centrality and network hubs

To compute the importance of a node in the network it is common to compute a centrality score. There are many definitions of centrality, and we will only present a small subsets of these definitions here. We all benefit from the idea of high centrality nodes, when we do a Google search and have a relevant page of returned search results. In this section, we introduce, degree centrality, betweenness centrality, and eigenvector centrality. Given that each of these measures is computed in a different way, each is measured so capture something different about the network.

Degree centrality

Degree centrality is the most simple centrality measure because it is just simply a node's degree. This means that under this measure, the most important nodes in the network are nodes with high degree. This centrality is attractive because it is easy to compute, having complexity in a sparse network of $O(E)$ (where E is the number of edges). We define degree centrality of node i , $\mathcal{D}(i)$ as,

$$\mathcal{D}(i) = \sum_j a_{ij} \quad (1.2)$$

Betweenness centrality

Betweenness centrality quantifies node importance, based on how many shortest paths go through a node. So, if a node appears on many of the shortest paths between node pairs, then it is considered to be an important node. We define betweenness centrality for a node i , $\mathcal{B}(i)$ as,

$$\mathcal{B}(i) = \sum_{i \neq j \neq t} \frac{\sigma_{jt}(i)}{\sigma_{jt}}, \quad (1.3)$$

where σ_{jt} is the total number of shortest paths between a pair of nodes, s and t that pass through i .

Eigenvector centrality

The idea behind eigenvector centrality is that a node should be prioritized not only based on its degree, but the degree of the nodes it connects to. That is, a node connected to other ‘important’ or high degree nodes should be ranked higher than one connected to many low degree nodes. The eigenvector centrality for node i , can be computed using the spectra of the adjacency matrix, \mathbf{A} . In particular, the vector of centralities, \mathbf{x} is the one satisfying the eigenvector equation,

$$\mathbf{Ax} = \lambda \mathbf{x}. \quad (1.4)$$

Because centralities are non-zero, the solution must be an eigenvector with all positive entries. Since multiple eigenvalues (λ) correspond to non-zero eigenvectors, the eigenvector corresponding to the largest eigenvector is used and the centrality scores for each node reflect its relative importance in comparison to the rest of the nodes. Moreover, the i -th entry of \mathbf{x} gives the eigenvector centrality for node i .

We visualized the results of each of these three presented centralities on the single cell network data in figure 1.5. We selected the highest centrality nodes, according to degree, betweenness, and eigenvector in the left, middle, and right panels respectively. From these highest centrality nodes, we visualized the order 2 ego graph for these nodes. An ego network for node i is simple the subgraph of all nodes within a path length of 2 from node i . This visualization gives a sense of what kinds of connectivity patterns each centrality measure favors. For example, we see that degree and eigenvector centrality have similar ego networks, as they are capturing nodes with a lot of connections. However, the ego network of the high betweenness centrality node is serving as more as a bridge between densely connected parts of the network.

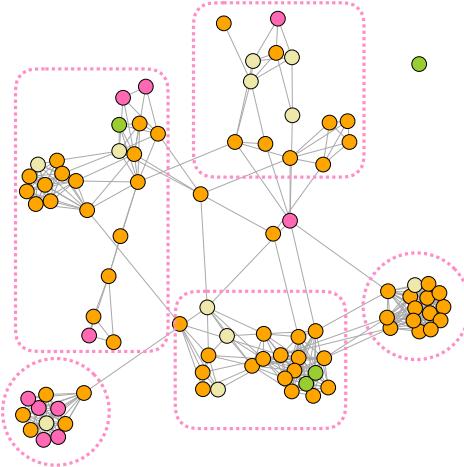


Figure 1.6: **Assortative Community Structure.** Nodes are tightly connected to each other and more sparsely connected to the rest of the network. Each community is outlined with a pink dotted line.

1.2 Introduction to community detection

A community in a network is broadly defined as a set of who share something in common in terms of their connectivity patterns in the network. One can think of a community as a clustering problem on networks, where the objective is to identify a set of nodes that are highly similar. The most basic type of community to understand is a network with assortative community structure. In this case, nodes are tightly connected to each other but more sparsely connected to the rest of the network. An example of a network with assortative community structure is shown in ?? Communities in the network are outlined with pink dotted lines.

Alternatively, networks can have a dissasortative structure where the between community edge density exceeds the within-community density. Finally, a core periphery structure can arise when there is a central core in the network that connects to the rest of the network and a set of peripheral nodes that connect to the core, but not to each other.

Community detection is a well-studied sub-domain of network science. The interested reader can refer to one of the comprehensive review articles (67; 44; 109)

1.3 Community detection methods

When performing community detection on a network, the objective is to segment nodes into one of K communities. This K can be known apriori or estimated through some kind of model selection or quality function computations. There are many optimization approaches that can be used to approach network community detection. In this section, we will introduce the current state-of-the-art approaches characterized as quality function maximization, deep learning, higher order clustering, probabilistic, and spectral methods. These methods are discussed based on their ability to handle networks of non-trivial size with diverse structures. We particularly elaborate on the stochastic block model and modularity techniques, as those are the the approaches built on in this thesis.

1.3.1 Quality function maximization with modularity

For quality function optimization, one writes down a quantity to optimize that seeks to identify a partition of the network into nodes that is representative of the network structure. The most common quality function for this task is known as modularity (85). Intuitively, modularity defines a null model for network that doesn't have prominent organizational structure. In particular, this null model is a random graph model, known as the configuration model (19). To generate an N -node network from the configuration model, one first specifies a fixed degree sequence, $D = \{k_1, k_2, \dots, k_N\}$. From this sequence, nodes are connected with k_i stubs that will ultimately be connected together. Finally, the graph is constructed by randomly choosing pairs of the created stubs and joining them. Based on how this network was generated, it is easy to specify the probability that an edge exists between a pair of nodes, i and j , or $p(a_{ij} = 1)$.

$$p(a_{ij} = 1) = \frac{k_i k_j}{2M}. \quad (1.5)$$

Here, k_i and k_j represent the number of edges for nodes i and j , respectively, and M is the total number of edges in the network.

Modularity was introduced in 2004 by Newman and Girvan (88). We define the modularity quality function, Q as,

$$Q = \frac{1}{2M} \sum_{i,j} \left[a_{ij} - \gamma \frac{k_i k_j}{2M} \right] \delta(z_i, z_j) \quad (1.6)$$

Here, γ is a resolution parameter (104) that controls the scale of community size. Large values of γ favor more small communities while smaller value enforce for fewer large communities.

In order to determine \mathbf{z} , the most computationally efficient approach is known as the Louvain algorithm (20). The Louvain algorithm is an agglomerative heuristic, which initially starts with each node in its own community and in the first match merges pairs of nodes if their merge leads to an increase in modularity. Each group of nodes assembled after this first pass becomes a new node in the network and a new weighted network is created between the set of new nodes. The weight on the edges of the new network are the number of edges from the original network that go between the sets of merged nodes. This process is continues iteratively until the modularity no longer increases. The reason that this approach is so computationally tractable is because the gain in modularity, ΔQ of merging two groups of nodes can be explicitly computed in closed form.

Modularity has shown to be effective in applications from neuroscience (78) to image segmentation (25). It has also shown to be effective in clustering high dimensional data that has been used to create a network. In 1.7, we used tSNE ?? to project the 50-dimensional single cell data into 2 dimensions. Points are colored by their cluster assignment according to k . We first performed k -means on the original 50 dimensional data (left) and Louvain community detection on the 5 nearest neighbor network representation (right). One benefit of the Louvain algorithm is that it does not require specifying the number of cluster. Moreover, in this example, the Louvain algorithm maximized modularity by partitioning the network into 10 clusters. Moreover, we also clustered the original data with 10 clusters. From these two partitions, we observe that creating a network representation of the data before clustering assists in identifying smaller, less prominent clusters.

1.3.2 Identifying communities with probabilistic approaches

Probabilistic community detection methods aim to find a partition of the network through likelihood optimization. Intuitively, the goal is to study the generative process of the node edges in terms of inferred community assignments. For example, given nodes i and j , one may model $P(a_{ij} = 1)$ as $g(z_i, z_j)$, where $g(\cdot)$ is some rule based on the node-to-community assignments. Two common

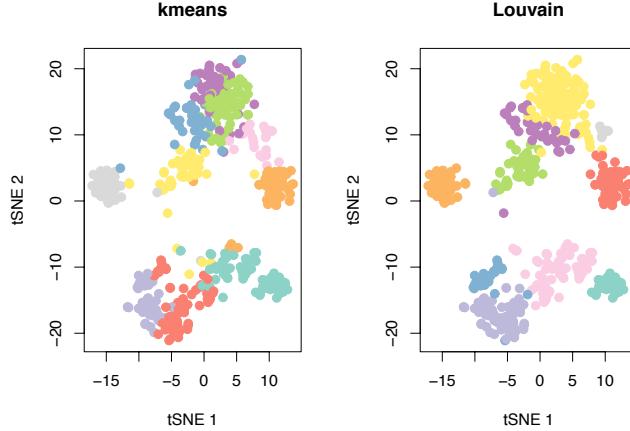


Figure 1.7: A comparison of k -means and the Louvain algorithm A comparison of the results of clustering the single cell dataset by visualizing the original 50 dimensional data with a 2-dimensional projection with tSNE. Points are colored by their cluster membership under k -means on the original data (left) and Louvain community detection (right) on the constructed 5 nearest neighbor network.

probabilistic community detection models are the stochastic block model (111) and the affiliation model (128). The definition and description of these models and inference techniques are described in depth in chapter ???. To facilitate working with probabilistic models, we first introduce some notation and background on inference techniques.

1.3.2.1 Probabilistic graphical models for statistical inference

Probabilistic network models are one approach to community detection that seek to model edge existence based on the node-to-community assignments. In doing so, the objective is to learn the node-to-community assignments that make the structure of the observed network the most likely. In this section, we will define some useful notation and concepts. To fit a probabilistic network model to data, we will define some useful notation and concepts that help simplify writing down and interpreting the likelihood.

Probabilistic graphical models enable efficient specification and manipulation of large probability distributions through semantic structures. Given a set of random variables, $\{A, B, C, D, E, F\}$, we seek to compute the joint distribution, $P(A, B, C, D, E, F)$. This joint distribution can be expressed with a directed acyclic graph (DAG), whose structure encodes dependencies between random variables. The DAG allows for the representation of the joint distribution in a factorized way, which

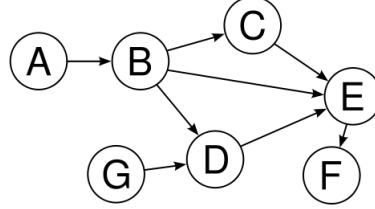


Figure 1.8: **Directed Acyclic Graph.** A directed acyclic graph (DAG) is formed based on dependency between random variable and allows for a fully factorized probability distribution.

is computationally useful. A DAG between the set of random variables, $\{A, B, C, D, E, F\}$ is shown in 1.8.

To translate a DAG between a set of N random variables, $\mathbf{X} = \mathbf{X} = \{X_1, X_2, \dots, X_N\}$ to its joint distribution, we rely on the Factorization theorem, which specifies that a DAG factors according to its parent/child relationships with,

$$P(\mathbf{X}) = \prod_{i=1:N} P(X_i | \mathbf{X}_{\pi_i}). \quad (1.7)$$

Here, π_i denotes the set of parents for node i . Using this information, we can write down the joint distribution for figure 1.8 as,

$$P(A, B, C, D, E, F) = P(A)P(B | A)P(C | B)P(D | B, G)P(E | D, B, C)P(F | E). \quad (1.8)$$

This introduced idea will help in subsequent sections to express a model graphically, write down the model likelihood, and use the likelihood to optimize for the most appropriate model parameters.

1.3.2.2 Stochastic Block Model

For an undirected, unweighted network \mathcal{G} with adjacency matrix, \mathbf{A} , we seek to partition each of the N nodes into one of K communities. We denote the node-to-community assignments as \mathbf{z} , with z_i specifying the community assignment of node i . Here, \mathbf{z} is a latent variable, with each entry taking on 1 of K states, or one of K community assignments. Figure 1.9 shows the

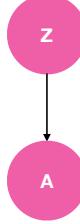


Figure 1.9: **SBM Graphical Model.** A graphical model is used to model the dependency between the node-to-community assignments, \mathbf{z} and the observed network adjacency matrix, \mathbf{A} .

dependency relationship between the node-to-community assignments. Here, the node-to-community assignments are treated as a latent variables because we seek to identify the \mathbf{z} that makes the observed adjacency matrix, \mathbf{A} the most likely. The crucial assumption of the stochastic block model is that nodes within a community are connected to nodes within their community and to other communities in a characteristic way. To this end, the model fitting procedure requires learning a set of within and between community connection probabilities. Under this approach, edges are treated as independent and identically distributed and deciding whether or node an edge exists between a pair of nodes is the learned connection probability between the communities to which each of the nodes belong.

Using the factorization rules described in section 1.3.2.1, we can specify the complete data log likelihood between \mathbf{z} and \mathbf{A} as,

$$\log P(\mathbf{z}, \mathbf{A}) = \log(P(\mathbf{A} | \mathbf{z})) + \log(P(\mathbf{z})) \quad (1.9)$$

To further specify these communities, we will define additional notation. First, let $\Pi_{K \times K} = \{\pi_{ij}\}$ be the matrix that specifies the within and between community edge probabilities. Using this information, we can model the probability of an edge existing between nodes i and j as,

$$P(A_{ij} = 1) \sim \text{Bernoulli}(\Pi_{z_i, z_j}) \quad (1.10)$$

We let $Z_i = \{Z_{i1}, Z_{i2}, \dots, Z_{ik}\}$ be a collection of binary indicators where Z_{ik} is 1 if i belongs to community k and 0 otherwise. We also let α_k be the probability that a node belongs to community k . With all of this information, we can write down each term of the complete data likelihood.

First,

$$\log(P(\mathbf{Z})) = \sum_i \sum_k Z_{ik} \log(\alpha_k). \quad (1.11)$$

Next,

$$\log(P(\mathbf{A} | \mathbf{Z})) = \sum_{i \neq j} \sum_{k < l} Z_{ik} Z_{il} [a_{ij} \log(\Pi_{kl}) + (1 - a_{ij}) \log(1 - \Pi_{kl})] \quad (1.12)$$

Optimizing the parameters of this incomplete data log likelihood requires computing the posterior $P(\mathbf{z} | \mathbf{A})$ but as shown by (32) is intractable. To address this issue, the posterior can be recast using a factorized approximation. This is accomplished by optimizing a lower bound of $\mathcal{L}(\mathbf{A})$. We let \mathcal{R}_A be an approximation of the posterior, $P(\mathbf{z} | \mathbf{A})$. To optimize the lower bound of $\log \mathcal{A}$, we seek the \mathcal{R}_A that is as close as possible to $P(\mathbf{z} | \mathbf{A})$. In other words, we define the lower bound of $\mathcal{L}(\mathbf{A})$ as $\mathcal{T}(\mathcal{R}_A)$, with,

$$\mathcal{T}(\mathcal{R}_A) = \log \mathcal{L}(\mathbf{A}) - \text{KL}[\mathcal{R}_A(\mathbf{z}), \mathbf{P}(\mathbf{z} | \mathbf{A})]. \quad (1.13)$$

Here KL denoted the Kullback-Leibler divergence (KL divergence) and the best approximation will be the value that makes the KL divergence the smallest. Jaakkola *et al.*, present a mean field approximation for the posterior distribution (58) as,

$$\mathcal{R}_A(\mathbf{z}) = \prod_i h(Z_i; \boldsymbol{\tau}_i). \quad (1.14)$$

Here $\boldsymbol{\tau} = (\tau_{i1}, \dots, \tau_{iK})$ and τ_{ik} is the approximation that node i belongs to community k , or $P(Z_{ik} = 1 | \mathbf{A})$. Furthermore, $h(\cdot; \boldsymbol{\tau}_i)$ denotes the multinomial distribution with parameter $\boldsymbol{\tau}$.

Daudin (32) *et al.*, show that the optimal estimate for τ_{ik} denoted $\hat{\tau}_{ik}$ satisfies

$$\hat{\tau}_{ik} \propto \alpha_k \prod_{j \neq i} \prod_l [\theta_{z_i, z_j}^{a_{ij}} (1 - \theta_{z_i, z_j})^{1-a_{ij}}]^{\hat{\tau}_{ik}}. \quad (1.15)$$

Here, α_k notes the probability that a node belongs to community k . Furthermore, after computing the set of variational parameters, the updates for $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ that maximize $\mathcal{T}(\mathcal{R}_A)$ are also shown by Daudin *et al.*, (32) to be,

$$\hat{\alpha}_k = \frac{1}{n} \sum_i \hat{\tau}_{ik} \quad \theta_{ql} = \sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{jl} a_{ij} / \sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{jl} \quad (1.16)$$

We have presented this variational approach for performing SBM parameter inference and likelihood optimization because this approach was appropriate for the work presented in this thesis. Variational inference is just one approach that can be applied to learn model parameters and was but a study by Zhang *et al.* (135) also show that belief propagation is very effective for this task (81). Briefly, belief propagation is a message passing algorithm for parameter inference in probabilistic graphical models. Given that parameter learning offer requires computing marginal distributions for a set of variables with a very large number of possible configurations, belief propagation uses the graphical model to reduce the complexity of the problem. Using the belief propagation to infer latent node-to-community assignments and update the model parameters was shown to perform superior to the variational approximation

This formulation of the problem and parameter optimization procedure works well and converges quickly for networks that have assortative community structures and a homogenous degree distribution. We will now explore how this classic formulation of the SBM can be modified to enable a broader application for a variety of networks.

1.3.2.3 Variants to the Classic Stochastic Block Model

The introduced stochastic block model is the most vanilla version in that it makes the assumption that the network is unweighted, each node is assigned to only one community. The introduced model also does not account for issues that may arise from degree heterogeneity (i.e. a large disparity in node degree in sets of nodes). Here, we will briefly discuss the approaches that adapt the stochastic block model to handle these issues and assumptions.

Edge Weights

The majority of the stochastic block model literature considers unweighted networks simply because describing a probabilistic model to handle both edge existence and edge weight is a challenging task. In the classic stochastic block model, we are simply modeling whether an edge exists based on the inferred community memberships of the edge stubs. Since edge weights can come in a

variety of forms (real-valued, count, etc.), it is difficult to immediately decide what distribution the edge weights should follow. In the past few years, this issue has been tackled in two papers (9; 98).

First, Aicher *et al.* developed a model and associated inference technique, for the weighted stochastic block model. Here, edge weights can be modeled by any exponential family distribution. The authors use a mixing parameter that allows for the control of the use of edge existence versus edge weights when learning node-to-community assignments. This method requires having an estimate of the number of communities, K , but the paper provides an approach to use Bayes' factors between two competing values of K to determine which model is a better fit. The inference for fitting this model is performed through a variational bayes approach (13).

To avoid having intuition about K , Peixoto (98) developed a non parametric bayesian approaches that is capable of inferring K with no prior knowledge. The assumption of the model is also slightly different and assumes a hierarchical structure between communities. The inference is achieved through MCMC sampling.

Degree Heterogeneity

Based on the variety of network structures and types, the assumption that the classic stochastic block model is an appropriate model for the data is often invalid. That is, for some networks, the fitted model may not actually be a good fit for the data. Work by Karrer *et al.*, introduced a simple extension to the classic stochastic block model, known as the degree corrected stochastic block model, that is informed by degree distribution as a proxy for the network structure. In networks where there is a high disparity between node degree (i.e. many high degree nodes and many low degree nodes), stochastic block models inference tends to partition the nodes into communities of high degree and low degree nodes. The approach for adapting the SBM to this setting is to learn a $K \times K$ matrix, θ , describing the number of edges between each pair of communities. These counts are modeled as poisson random variables. The likelihood of the observed network under this poisson assumption takes into account node degrees.

The restriction of single community membership

As it is often observed in social networks, the assumption that every node belongs to only a single community is restrictive. To address this issue, approaches have been developed to allow nodes to participate in a mixture of communities (12) or to overlapping groups (70). Airolidi *et al.*, pioneered the development of the mixed membership stochastic block model (12), where instead of

modeling a node's membership in each community in a binary manner, the authors allow a node to belong to multiple communities. The generative process for this approach for modeling the existence of an edge between nodes p and q in a network with K possible communities and θ representing the between community connection probabilities.

- For each node p , draw a mixed membership vector $\pi_p \sim \text{Dirchelet}(\alpha)$
- Then for each pair of nodes (p, q) , draw $\mathbf{z}_{p \rightarrow q} \sim \text{Multinomial}(\pi_p)$, $\mathbf{z}_{q \rightarrow p} \sim \text{Multinomial}(\pi_q)$
- Sample the edge between p and q as, A_{pq} , where $A_{pq} \sim \text{Bernoulli}(\mathbf{z}_{q \rightarrow p}^T \theta \mathbf{z}_{q \rightarrow p})$

Following the development of the mixed membership stochastic block model, Latocuhe *et al.* (70) addressed an important limitation of (12). Since the probability of an edge between a pair of nodes p and q depends on a single draw of $\mathbf{z}_{p \rightarrow q}$ and $\mathbf{z}_{q \rightarrow p}$, the class memberships of nodes p and q towards other nodes in the network are ignored. Moreover, this model adapts the mixed membership stochastic block model to incorporate more structures of the network.

1.3.2.4 Affiliation model and inference

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer

1.3.3 Deep Learning Approaches

In recent years, deep learning has begun to revolutionize many fields, including network analysis. Perozzi *et al.*, pioneered the use of deep learning in community detection with the development of DEEPWALK (100) to learn a latent space representation of nodes in a lower dimensional space (i.e. an embedding). Once the network is embedded in a lower dimensional space, simple clustering techniques, such as k -means (54) can be used to partition the network into communities. The approach to learn an embedding for the network is based on random walks on the network (91; 49). A random walk on a network involves choosing a starting node and traversing the network by hopping between adjacent nodes. The DEEPWALK approach seeks to learn an embedding of the nodes that preserves the sets of nodes traversed in a random walk. To do this, the authors used Word2Vec, a tool from natural language understanding that allow for the specification of a node embedding that enable accurate prediction of a word's context, given the word (79). To adapt this context to networks, a random walk is treated as a sentence and nodes are treated as a word within the sentence. Moreover, the analogous task to the problem in text data to a network is to accurately assign a probability predict a set of nodes likely to be seen with the node of interest. Moreover, this problem is solved using the same optimization approach as Word2Vec

Based on the success of DEEPWALK, the method was followed up with Node2Vec in 2016 (51). While node2vec also uses the random walk framework to specify the optimization problem, they modify how the random walk is performed to enable an embedding that captures different aspects of a potential network community. For example, one may describe a community by a set of nodes located close to each other in the network with many common neighbors and connections to common neighbors. This assumption is known as network homophily (65). Alternatively, perhaps a good definition of a community is a set of networks that have similar roles in the network. This idea is known as structural equivalence (75). For example, a grouping of nodes that take into account their degree, with the community assignments being highly related to node degree. To modify the random walk so that it leads to a model that gives flexibility in the nature of retrieved communities, the authors introduced a search bias term, which controls whether the random walk is performed in a breadth-first or depth-first search parameter. If on a random walk, the path is traversed in a depth-first search, favoring the exploration of a larger area of the network far from the source, the

resulting community aligns with the homophily hypothesis. A random walk performed in a breadth first manner that restricts the path to nodes neighboring the source and tends to capture nodes based on structural equivalence (i.e. a hub, or highly connected node).

1.3.4 Spectral community detection methods

1.3.5 Higher order network analysis

1.4 Community detection in computational biology

A community approach to network analysis has shown to be fruitful in particular, in the analysis of biological and brain connectivity applications. In this section, we will describe examples of analyses where the identification of communities provided insight and understanding for a scientific problem.

Multiple experimental modalities exist that enable the collection and analysis of biological data. Understanding protein expression, gene expression, microbiome composition, metabolomic profiles, genomic mutations, and immune profiling are just a few of examples of biological data that is studied routinely for insight into human health. With most experimental platforms producing high dimensional data, it is crucial to have good tools for interpretation, visualization, and prediction. Machine learning techniques in computational biology have revolutionized prediction in healthcare and medicine. Here, we outline particular examples of how community detection lead to important biological understanding and predictive ability.

1.4.1 Immunological profiling to establish a pregnancy immune clock

A study lead by Aghaeepour *et al.*, demonstrated that there is a typical timing of immunological events in a healthy, term, human pregnancy (8). Immunological profiling was performed on a training cohort of 18 women, using a technology called mass cytometry (18) was used to quantify various features of the immune system, such as, cell type abundances, signaling activity. From this set of measured immune features, a correlation network from the training cohort to identify which immune features were potentially related or working together. Simultaneously, a regression model was trained to identify immune features associated with increased gestational age. When communities were identified in the network of immune features, there were two important observations. First,

immune features of the same type (i.e. cell signaling vs. cell frequency) were aligned with community labels. Second, sets of features associated with a particular gestation age often fell in the same community, indicating their synchronous activity during the pregnancy. Finally, after identifying influential nodes in their ability to predict stage in pregnancy, according to the regression model, the communities of these nodes were more closely examined to uncover further insight into the immunological mechanisms occurring throughout the pregnancy time course.

1.4.2 Uncovering differences in microbiome community structure in patients with inflammatory bowel disease

The microbiome refers to the collection of bacterial species that populate an organism's gut. Microbiome analysis has recently gained attention, as its biological implications are large for health and disease. A 2017 review article presented the idea that the development of network analysis approaches for microbiome data is under explored and has great potential for advancing biological understanding and interpretation of these data (71). A network in this context is typically constructed based on some notion of co-occurrence or correlation between microbial species, profiled across samples. A recent example where community detection played a key role in the biological understanding was introduced in 2017 and assessed the interplay between microbial co-occurrence structural organization patterns between patients with and without inflammatory bowel disease (14). Communities were identified in the healthy and diseased networks, using classic modularity maximization (48). After identifying a community structure for each network, the similarity of these partitions was quantified with the Rand index (119), which showed to be statistically significant under a permutation test. This observation allowed the authors to understand that the core structure from a healthy microbiome was conserved even in diseased patients, but allowed for more careful probing of the subtle differences. First, the functional roles of the members of each community were interrogated. Some interesting co-occurrence relationships within communities were identified, such as the loss of strong clustering, or association propensity between pro and anti-inflammatory species within the diseased networks. This interplay between pro and anti inflammatory species is thought to play a pivotal role in the maintenance of a healthy gut microbiome.

Next, the authors used the community structure of each network to study the differences in node roles (i.e. importance) between the healthy and IBD networks. Within the neuroscience community,

there have been numerous efforts to characterize nodes, in terms of the role they play connecting communities or as an important node within a community (124). Nodes have the potential to be *connectors*, where they have high ‘participation’ or connections with many nodes across numerous communities. Alternatively, a node can be an intramodular hub, where it serves as a high degree node, connecting to many members of its community. After assessing the role of each node in the healthy versus IBD network, the roles of many nodes were not consistent between the two networks. Most notably, the most prominent community-connector nodes in the healthy network were lost in the IBD network. Further, there were some nodes with few intermodule connections in the healthy network, that increased their role as a connector node in the IBD case. The interrogation of nodes with a dramatic change in their role are good candidates for follow-up investigation.

Overall, the partitioning of each network into communities allowed for a systematic comparison between the healthy and disease network and to prioritize specific species (nodes) and co-occurrence patterns for further investigation.

1.4.3 Community detection for analysis of flow cytometry data

Flow cytometry allows for the simultaneous quantitative analysis of a large population of cells within a biological sample. Typically, cells are strained with fluorochrome-conjugated antibodies which emit light upon encountering laser beams in the flow cytometry machine. This emitted light is measured and reported as a quantitative measurement of the cell. An important analysis of flow cytometry data is the ability to automatically group cells based on their similarities in light emission and quantification. While this process was historically performed manually, there has been a significant amount of work to develop computational methods that can successfully segment cell populations, automatically (7). A network-based approach to this problem, known as SamSPECTRAL was introduced in 2010 by Zare *et al.* (133). In this approach, the authors seek to segment a population of cells into distinct populations of cells, through the construction of a similarity network and identification of communities. In this network, the nodes are comprised of the cell types in a sample, and edges between nodes indicate the similarity between a pair of cells, based on the quantification of their emitted light. Because a high throughput biological sample could contain as many as biological points, this approach seeks to first create a smaller representation of the data, build a network on this smaller version, and ultimately segment the data this way.

To create a network of the flow cytometry data, a large subset of data points (cells) are first sampled and denoted as ‘registered’ nodes. The next step is to look at the collection of ‘unregistered nodes’ and ultimately assign them to their closed registered node neighbor. Iteratively, for each registered node, denoting one of these registered nodes as P the set of unregistered nodes within some defined distance h become registered to p . The set of unregistered nodes that were newly assigned to be registered are removed from the set of unregistered nodes. This process is repeated until there are no more unregistered nodes. Each set of nodes registered with the same label are denoted as a community (an inconvenient label, given a network will be constructed and communities will be identified). A weighted network is constructed between these registered communities with edge weights quantifying the similarity in the quantitative features (as measured with the flow cytometry machine) between a pair of a communities. Once this weighted graph is created, a spectral community detection method (127) is applied to segment the network into 1 of K network communities. These is one final post-processing step, motivated by previous work in computational flow cytometry methods, to combine the agglomerate a pair of network communities if members if the community show similarity greater than a predefined threshold (in terms, again, of their measured flow cytometry properties). The usefulness of this approach is that it exhibited outstanding performance on datasets containing clusters of challenging shapes. For example, overlapping clusters, non-elliptical shaped clusters, or low-density clusters. To summarize, the SamSPECTRAL method shows how network communities can be used to automate a challenging computational task and enables biologists to better study and characterize their data.

1.4.4 Understanding genetic diversity of the malaria parasite genes

Rich genetic diversity in the *var* genes of the human malaria parasite has been shown to contribute to the complexity of the epidemiology of the infection and disease. The parasite can change which of the *var* genes are expressed at any given time on the infected red blood cell, which prevents the antibody from recognizing and resisting the new protein. One diversity-generating mechanism is recombination, which is the exchange and shuffling of genetic information during mitosis and meiosis (16). The ability to understand genetic diversity is complicated by inadequate tools to uncover the phylogeny, or genetic relationship between sequences resulting from recombination events, in a scalable and statistically rigorous way. The typical analyses for evolutionary data assume a tree-like

relationship between events, which is unrealistic for recombination data. To address this challenge, (69) use a novel approach: they cast their problem in terms of a collection of networks. Then, they apply community detection to each of the networks and use the properties of the communities to generate hypotheses of the mechanisms behind the recombination process. To investigate the heterogeneity and the corresponding possible patterns in recombination events across a set of 307 sequences from the *var* gene, the authors restricted their analyses to 9 particular “highly variable regions” (HVR) within each of the 307 sequences. Then for each HVR, they constructed a network, where the nodes represented the 307 sequences and an edge was placed between a pair of nodes if they had evidence of a recombinant relationship, based on a notion of sequence similarity within the particular HVR. Communities were then identified in each of the 9 networks using a degree-corrected stochastic block model (SBM) approach (62).

After identifying communities within each HVR network, the authors used two summary statistics to formulate their biological hypothesis. First, the variation of information (107) was used to compare the community assignments of nodes (i.e. each of the 307 sequences) across the 9 HVR networks. They observed that each network had a prominent community structure (i.e. far from random) and that the community assignments between networks were quite distinct. These observations motivated the hypothesis that recombination events occur in constrained ways, leading to a strong community structure, and that one should analyze HVR networks individually instead of building a consensus network that aggregates the HVR networks. Next, they used *assortativity* (84) to overlay the network structure with various known biological features of the sequences, such as *var* gene length. Specifically, assortativity quantifies the tendency of nodes of the same type (e.g. same gene length) to be connected in the network. They observed that three HVR networks had community structure correlating strongly with two biological features (i.e. nodes of the same biological label tend to group together), while three other HVR networks with highly heterogenous community structure were unaligned with any of the known biology. These observations allowed for the formulation of the hypothesis that the HVRs that are unrelated to each other also promote recombination under unrelated constraints and are responsible for fostering genetic diversity to avoid immune evasion.

Given the ability to find communities within each HVR network and the lack of similarity in community structure between HVR networks, (69) were able to formulate and test hypotheses for the diversity-generating mechanisms of *var* genes, and this would have been difficult using standard phylogenetic approaches or without adopting a community-based perspective. The application of the stochastic block model to this task provided a statistically grounded approach for testing the plausibility of the model.

1.4.5 Analysis of high dimensional single cell data for tumor heterogeneity

A very beautiful application of community detection is the development of a network and community detection based method, called PhenoGraph for the analysis of single cell data, presented by Levin *et al.*, in 2015 (73). Single cell technologies allow for the profiling of cells individually within a sample. Recent attention and methods development have focused on the use of RNA sequencing and mass cytometry to give a high dimensional profile for a single cell. Single cell technologies have enabled for an advancement in the understanding of the pathobiology of cancer in that cells within a tumor have been shown to exhibit a large amount of heterogeneity at the single cell level. Furthermore, this heterogeneity has important functional and clinical significance (77). The data produced by these single cell technologies profiles millions of cells, based on multiple features (whether those be genetic, immunological, or signaling response). Moreover, a key challenge is to accurately separate individual cells into biologically meaningful subpopulations or cell phenotypes. While we will mostly profile the community detection based method used for this task, the implications of this work lead to the identification of a cellular phenotype and a corresponding gene expression signature which was highly correlated with accurate prediction of patient survival rates.

Unsupervised analysis of cell types is a challenging problem as there are millions of cells, with each cell being a point in d -dimensional space. Traditional clustering algorithms are too slow, or require assumptions about the number of clusters, or the shape of the data in high-dimensional space. One benefit of community detection on networks is that many methods do not require specification of the number of clusters and are agnostic to the shape of the data in high dimensions. The first step of PhenoGraph is to build a k -nearest neighbor network between pairs of cells. To do this, each cell is connected to its k -nearest neighbors. The second step of the algorithm refined the k -nearest neighbor network to prioritize keeping the most similar pairs of nodes connected in the network and

removing extraneous connections. This is done by creating a new weighted network between the cells based on the Jaccard similarity measure. In this context, the Jaccard similarity between a pair of nodes reflects the similarity of their neighbors in the network. With this refined network, modularity based community detection was applied and each of the resulting communities corresponds to a distinct cellular phenotype. When this method was applied to a manually gated (i.e. cells were manually separated dataset), PhenoGraph showed very strong performance for multiple values of k . The authors specifically tried, $k = \{15, 30, 45, 60\}$. The authors also provide an approach to add supervision to the problem, which uses partially labeled data set. In this context, this means that some of the cells have a classification. Moreover, given that the network contains N nodes, with T labeled nodes ($T < N$), the objective is to label the remaining $N - T$ nodes. Based on a concern that network-based classification methods operating on a majority vote rule for a node's neighbors, the authors sought to develop an approach that would not suffer in circumstances where a node's closest neighbors were a small subset of the available labeled data. This issue is mediated through the use of label information on the whole network through a random walk. Conceptually, starting from an unlabeled node, the random walker can move through the network, taking into account edge weight information at each step. The random walk classification scheme from an unlabeled node is therefore the probability of its random walk ultimately arriving at a node from each of the classes. The probability of an unlabeled node reaching a node in each of the labeled classes can be computed in a straightforward way, using the graph laplacian (118). Overall, the findings of this paper use community detection to allow for the analysis and understanding of tumor heterogeneity data that was not possible with standard high dimensional data analysis techniques. The authors suggest that this method is useful in characterizing primitive cancer cells and for the identification of cell biology features that define particular biological states and clinical outcomes.

1.4.6 Identification of virulence factor genes related to antibiotic resistance of uropathogenic *E. coli*

Urinary tract infections are primarily caused by uropathogenic *E. coli* (UPEC). In their study Parker *et al.*, seek to better understand UPEC antibiotic resistance, which prevents patients from being treated for urinary tract infections. Using a cohort of 337 *E. coli* patient isolates, the authors looked closely at the virulence factor genes of these patients. Virulence factors are non conserved or are

carried on mobile genetic elements and elicit biological functions that relate to uropathogenesis (i.e. the onset of a patient getting a UTI). The biological function of virulence factors are known and allow for the development of therapeutic agents. In the analysis, the presence or absence for each of 16 virulence factors was determined. A network was constructed between the 337 patient isolates, with each edge reflecting the pairwise similarity in their virulence factor profiles. Modularity based community detection was then applied to this network and partitioned it into 4 different communities. Most remarkably, each of the 4 communities was characterized by clinical isolated described by either a single or pair of virulence factor markers. These pairs of related virulence factors were then probed further to investigate their role in antibiotic resistance. This approach offers a new way to integrate genomic and individual patient information to determine which types of antibiotics might be most effective.

1.5 Challenging problems in community detection

1.5.1 Temporal Networks

1.5.2 Multilayer networks

1.5.3 Network Comparison

1.5.4 Large Networks

1.5.5 Attributed Networks

1.6 Thesis Contribution

In this thesis, we focus on 3 of the discussed challenging problems of adapting community detection approaches to multilayer, large, and attributed networks.

1.6.1 Relevant Publications

The work addressed in this thesis can be found in the following publications. note that we have organized the publications by the theme they address.

Community detection in multilayer networks

1. *Clustering Network Layers with the Strata Multilayer Stochastic Block Model.* **N. Stanley, S. Shai, D. Taylor, P.J. Mucha.** IEEE Transactions on Network Science and Engineering. 2016.
<http://ieeexplore.ieee.org/abstract/document/7442167/>
2. *Enhanced Detectability of Community Structure in Multilayer Networks Through Layer Aggregation.* **D. Taylor, S. Shai, N. Stanley, P.J. Mucha.** Physical Review Letters. 2016. <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.116.228301>

Community Detection (General)

1. *Case Studies in Network Community Detection.* **S. Shai, N. Stanley, C. Granell, D. Taylor, P.J. Mucha.** Appears as a chapter in the Oxford Handbook of Social Networks. 2017.
<https://arxiv.org/abs/1705.02305>

Pre-Processing for Community Detection in Large Networks

1. *Compressing Networks with Super Nodes.* **N. Stanley, R. Kwitt, M. Niethammer, P.J. Mucha.** Under Review. 2018. <https://arxiv.org/abs/1706.04110>

Community Detection for Attributed Networks

1. *Stochastic Block Models with Multiple Continuous Attributes.* **N. Stanley, T. Bonacci, R. Kwitt, M. Niethammer, P.J. Mucha.** In preparation. 2018.

1.6.2 Software

The three developed methods described in this thesis are available and maintained in github.

1. **sMLSBM: Fitting a multilayer stochastic block model.** <https://github.com/stanleyN/sMLSBM>
2. **SuperNode: For compressing a large network.** <https://github.com/stanleyN/SuperNode>
3. **Attributed SBM: For fitting an SBM with multiple continuous attributes.** <https://github.com/stanleyN/AttributedSBM>

1.6.3 Thesis Statement

CHAPTER 2

A multilayer stochastic block model

In this chapter we present the strata multilayer stochastic block model (sMLSBM). The sMLSBM method and inference described here is described in *Clustering Network Layer with the Strata Multilayer Stochastic Block Model* (115). The goal in developing this method is two-fold. First, we seek to develop an approach to cluster network layers within a multilayer network. Second, we wish to develop a novel extension to the stochastic block model to handle the information contained across network layers and determine which subsets of the network layers are likely to be samples from the same stochastic block model.

2.1 Introduction to multilayer networks

Currently, we are relatively comfortable working with a single network of nodes and edges, capturing one type of relational definition. We have seen this numerous times thus far in this thesis, from modeling similarity of immune features in women during pregnancy to profiling microbiome species co-occurrence patterns in patients with IBS. With the consistently improving ability to generate and analyze large amount of biological data, there is often the opportunity to generate multiple relational definitions between a set of objects. This could be simply the desire to compare a gene co-expression network across multiple tissues (136), or the desire to study multiple microbial co-occurrence networks in different sites of the body (121). Multilayer networks provide a framework to do this, in that each relational definition leads to a new layer in the network (64; 21; 35). Such data and corresponding networks have shown to be useful in many contexts, such as, in the comparison of genetic and protein-protein interactions in a cell (29), in understanding underlying relationships and community structure across social networks (50), and in the analysis of temporal networks (80). Furthermore, recent advances in the mathematical foundations for multilayer networks have made

analysis of these types of data more feasible. In particular, (35) has introduced a mathematical formalism with tensors. Doing so allows for the calculation of important network quantities, such as centrality and clustering coefficients, as well as modularity (80). Thus, given the inherent multiplexity of network data across fields as well as recent theoretical developments for handling these types of data, there exists a need for the development of appropriate tools that can leverage information from all layers to elucidate structural patterns.

Inspired by the ideas in (34) that groups of layers often provide redundant information, we seek to further explore this idea to identify sets of layers, which we denote as “strata”, with each stratum described by a single probabilistic model based on community structure. This effectively amounts to defining *local* probabilistic network models, and is analogous to biclustering (76) or co-clustering (40) problems. Moreover, our method can be regarded as a joint clustering procedure, in which the nodes and layers of networks are clustered simultaneously. Just as in (40), where the objective is to jointly cluster words and documents such that joint word-document subgroups correspond to particular topics, our objective is to cluster network layers such that each stratum is a set of layers with a characteristic community structure. To achieve this goal, we have developed the strata multilayer stochastic block model (sMLSBM). We additionally emphasize that by collectively utilizing similar layers in a principled way, we can achieve more robust community detection and parameter inference for the probabilistic community detection models that describe each stratum.

2.2 Comparing network layers based on community structure

The problem of aggregating layers in a multilayer network is closely related to the problem of clustering networks. That is, given an ensemble of networks, one aims to identify sets such that networks within a set have similar characteristics. These characteristics, or “features” in this context, can describe any of the following: micro-scale structural properties such as subgraph motifs (122; 120); multiscale properties such as community structure (92; 90; 57), the spectra of network-related matrices (24) and by defining latent roles (23). Although clustering layers in a multilayer network is closely related to clustering networks in an ensemble, these are distinct problems with different difficulties and nuances. We focus on the prior pursuit; however, we expect for certain network ensembles that it will be beneficial to modify and apply our methods to the clustering of

networks.

In this work, we analyze and compare layers in a multilayer network based on their community structure. Community detection in single-layer networks is an essential tool for understanding the organization and functional relatedness between nodes in a network (101; 43). Although there are many definitions for what constitutes a “community” (106), one often assumes an “assortative community” in which there is a prevalence of edges between nodes in the same community as compared to the amount of edges connecting these nodes to the remaining network. In seeking to identify such communities, numerous approaches have been proposed, including those based on maximizing a modularity measure (86) and fitting a generative probabilistic model (59). Because each of these approaches present computational challenges for efficiently detecting communities, numerous heuristics exist for developing practical algorithms (102; 43; 72; 27; 89).

While our approach is to define a probabilistic model for multilayer community structure, we note that there have previously been approaches to understand similarities in network ensembles that are grounded in exploiting similarities in community structure between networks. In (90), the authors seek to partition a group of networks into subgroups through construction of a network of networks (NoN). Communities in the NoN are chosen such that the networks representing the nodes are sufficiently similar in their underlying community structure. In one significant application of this method, the authors clustered gene co-expression networks and found an increased number of significant functional enrichment categories for biological processes. Similarly, in (57), the authors explore mesoscopic similarity between layers using an informational theoretic approach. While they have designed their method to handle any feature of network architecture, they highlight their ability to quantify similarity between network layers based on node-to-community assignments in the layers.

In seeking a statistically-grounded approach for studying communities in multilayer networks, we consider the stochastic block model (SBM) (112), a popular generative model for community structure in networks. The assumption of the SBM is that nodes in a particular community are related to nodes within and between communities in the same way, thus allowing SBMs to describe several types of communities (e.g., assortative, disassortative, core-periphery, etc. (106; 10)). There are many other appealing aspects of stochastic block models; for example, a model-based approach allows for the denoising of networks through the removal of false edges and the addition of missing edges

(59; 52). As we introduced in chapter 2, the inference procedure for fitting SBMs to an undirected network with N nodes and K communities involves learning the two parameters, π and \mathbf{Z} . Parameter π is a $K \times K$ symmetric matrix, where π_{mn} gives the probability of an edge existing between a given node in community m and another node in community n . Matrix \mathbf{Z} is an $N \times K$ indicator matrix, wherein each binary entry Z_{im} indicates whether or not node i is in community m . Each row of \mathbf{Z} is constrained such that $\sum_{m=1}^K Z_{im} = 1$, i.e. each node only belongs to 1 community. We also define vector \mathbf{z} , which has entries $z_i = \text{argmax}_m\{Z_{im}\}$ that indicate the community to which node i belongs. For a given network, these parameters are often inferred through a maximum likelihood approach, and once learned, they provide information about the within and between community relatedness.

2.3 Related work in community detection of multilayer networks

Due to the ubiquity of network data with multiple network layers, community detection in multilayer networks constitutes an important body of research. Important directions include generalizing the modularity measure (80) and studying dynamics (33) for this more general setting.

Given the usefulness of SBMs for the understanding of node organization in single-layer networks, it is important to extend SBMs to the multilayer framework, and indeed this direction of research is receiving growing attention (53; 93; 15; 123; 97). In this context, the general assumption is that there are shared patterns in community structure across the layers of a multilayer network, and the goal is to define and identify a stochastic block model that captures this structure. These works have explored many types of applications that can arise involving multilayer networks, and have therefore given rise to several complementary models for multilayer stochastic block models (MLSBMs). We now briefly summarize this previous work that is very related, but notably different, from the model we study herein.

In Refs. (53; 93; 15), the authors studied situations in which many layers follow from a single SBM. In these instances, it is possible to obtain improved inference of the SBM parameters by incorporating multiple samples from a single model. For example, in Ref. (53) the authors considered an increasing number of layers, L , and explored asymptotic properties of the estimated SBM parameters. Specifically, they fit an SBM to each individual layer in a way that utilizes the information

from all layers, and they showed convergence of these estimators to their true values as $L \rightarrow \infty$. For a network with L layers and K communities in each layer, their approach requires an estimate of the community assignment matrix \mathbf{Z}^l and probability matrix π^l for each layer l , the latter of which involves learning $K(K + 1)L/2$ parameters. To this end, the authors extended the variational approximation for approximating the maximum likelihood estimates of SBM parameters introduced in single-layer SBMs introduced in (?) to the multilayer setting.

Ref. (53) was followed up by Ref. (93), wherein the authors addressed issues that can arise for the model when K and/or L is large, or if the network is sparse. They proposed a modified model called the restricted multilayer stochastic block model (rMLSBM). In this model, instead of learning a set of L independent parameters, π_{mn}^l , for each pair, (m, n) , each entry in π is fully layer-dependent so as to produce a reduction in the number of free parameters. Specifically, to determine the probability of an edge between a node from community m and a node from community n in layer l , they use a logistic link function and model the probability as $\text{logit}(\pi_{mn}^l) = \pi_{mn} + \beta_l$. The β_l is an offset parameter representing the particular layer or type of edge. In this model, it is necessary to learn $K(K + 1)/2 + L$ total parameters. Thus, the maximum likelihood estimate for an rMLSBM is a regularized estimator.

Consistent with the theme of fitting a single block model to a collection of layers, Ref. (15) is similar to Refs. (53) and (93) in that the authors seek to leverage information from all layers by considering the joint distribution of layers. Using this, they estimated quantities such as the marginal probabilities of node assignments to communities and the edge probabilities within and between groups. An interesting aspect of their approach is that they introduce a covariate capturing the coupling between pairs of nodes. For a network with K communities and L layers, this requires the estimation of $(2^L - 1)K^2 + (K - 1)$ parameters.

We summarize Refs. (123) and (97), which provide techniques to determine whether a single layer network is the result of an aggregation procedure in a multilayer network. In Ref. (123), the authors defined a version of multilayer stochastic block model and an inference procedure for assessing whether or not a single-layer network was actually obtained from an aggregation of layers in a multilayer network; they considered the aggregation of layers using boolean rules. Ref. (97) describes two possible generative processes for multilayer networks: the *edge-covariate* and *independent-layer* models. In the edge-covariate model, an aggregated network is defined in which a given edge (i, j)

only appears in a single layer. Aggregating the layers in a multilayer network into a single network representation combines all of the edges from each of the layers. Thus, the translation of this idea into a generative model involves choosing a layer membership for each edge and sampling edges with a probability conditioned on adjacent nodes. In the independent-layer model, layers are generated independently from each other and the only constraint is that group membership of the nodes are the same across all layers.

While motivation to pursue this problem originated from (34), we point out that our approach does not provide a method for aggregating layers or reducing the number of layers in the network. Instead, it can in a sense compress the network in that the learned stochastic block model parameters for each stratum can be used to generate a sample network to serve as a consensus for that stratum.

2.4 A Summary of Novel Contributions of sMLSBM

While the literature on MLSBMs has recently grown quickly, there is still a need for a probabilistic generative model that allows for the layers in a multilayer network to be described by multiple SBMs. To this end, we developed a novel multilayer stochastic block model, sMLSBM, that assigns network layers into disjoint sets that we call strata, where a collection of layers in a given stratum are assumed to be samples from the same underlying generative model. Our method can be viewed as a joint clustering procedure, where we seek to group layers into strata and nodes into communities. That is, we seek to simultaneously find layer-to-strata and node-to-community assignments.

In order to address practical applications that can involve multilayer networks with several strata, layers, communities and nodes, we introduce an algorithm that effectively partitions layers into strata and an inference procedure to learn the SBM parameters for each stratum. Importantly, these two steps—assigning nodes to communities and layers to strata—are combined in an iterative algorithm so that an improvement in community detection can lead to an improvement in the clustering of layers into strata, which can iteratively lead to further improvement in community detection, and so on.

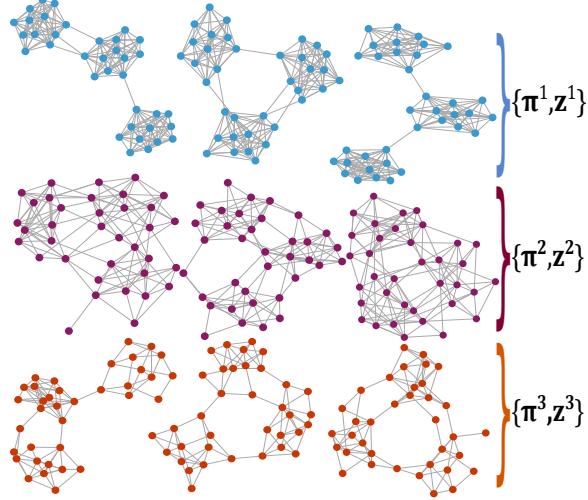


Figure 2.1: **Objective of strata multilayer stochastic block model (sMLSBM).** Each of the $L = 9$ networks here represents a layer in a multilayer network. Every network layer has $N = 36$ nodes that are consistent across all layers. There are $S = 3$ strata as indicated by the three rows and the colors of nodes. Clearly, network layers within a stratum exhibit strong similarities in community structure. That is, although each layer follows an SBM with $K = 3$ communities, the SBM parameters are identical for layers within a stratum but differ between layers in different strata. We would like to partition the layers into their appropriate strata and learn their associated SBM parameters, π^s and Z^s .

2.5 sMLSBM Model Definition

Under the sMLSBM, the network layers, $G^l(N, \mathcal{E}^l)$ are assumed to be generated by a set of S stochastic block models, where the layers in stratum $s \in \{1, 2, \dots, S\}$, are parameterized by π^s and Z^s (or equivalently, vector z^s , which has entries $z_i^s = \text{argmax}_m \{Z_{im}^s\}$). Note that the parameters π^s and Z^s for a single stratum are analogous in meaning to their respective parameters in the single-layer SBM case. For each stratum s , we let $\mathcal{L}^s \subseteq \mathcal{L}$ denote the set of layers corresponding to s , so that $\mathcal{L} = \bigcup_s \mathcal{L}^s$ and $\emptyset = \mathcal{L}^s \cap \mathcal{L}^t$ for all $s, t \in \{1, \dots, S\}$, $s \neq t$. We let $L^s = |\mathcal{L}^s|$ denote the number of layers in strata s so that $\sum_s L^s = L$. Finally, we allow the number of communities, K^s , to vary across the strata.

For a given multilayer network, our objective during inference is to identify the stratum assignment of each layer and to learn the collection of strata parameters, $\Pi = \{\pi^1, \pi^2, \dots, \pi^S\}$ and $\mathcal{Z} = \{Z^1, Z^2, \dots, Z^S\}$. The learned SBM parameters for a stratum represent a consensus for the associated layers, and so in that sense can be interpreted as reducing the effective number of layers

(34). However, strata can also be interpreted as a way to simply identify layers with similarities in community structure. Figure 1 shows a toy example of a multilayer network with $S = 3$ strata, where each layer has $N = 36$ nodes and $K = 3$ communities. Each individual network in this figure represents a layer in the network. The nodes in the layers belonging to each stratum are colored according to their stratum membership; moreover, it is easy to see that layers of a stratum exhibit high similarities in community structure.

As part of our procedure, we specify another parameter that we refer to as the adjacency probability matrix, $\boldsymbol{\theta}^s$, which can be computed from $\boldsymbol{\pi}^s$ and \mathbf{Z}^s . Specifically, $\boldsymbol{\theta}^s$ is an $N \times N$ matrix such that θ_{ij}^s gives the probability of an edge between nodes i and j in stratum s . That is, $\theta_{ij}^s = \pi_{z_i^s z_j^s}^s$, where z_i^s specifies the community number for node i in stratum s . Finally, we define the matrix \mathbf{Y} of size $L \times S$, wherein an entry Y_{ls} is a binary indicator of whether or not layer l is assigned to stratum s . Note that $\sum_s Y_{ls} = 1$. We also define a vector \mathbf{y} , which has entries $y_l = \text{argmax}_s \{Y_{ls}\}$ to indicate the strata to which layer l belongs.

2.6 Inference for learning model parameters of sMLSBM

The procedure for fitting an sMLSBM to a given network requires finding the layer-to-strata memberships and node-to-community memberships that best describe the multilayer network. For notational convenience, we introduce hat notation to represent the learned parameter estimate from the inference procedure. We can write down the marginal likelihood for the collection of network layers, \mathcal{G} , as,

$$p(\mathcal{G} \mid \boldsymbol{\Pi}) = \sum_{\mathcal{Z}} \sum_{\mathbf{Y}} p(\mathcal{G}, \mathcal{Z}, \mathbf{Y} \mid \boldsymbol{\Pi}). \quad (2.1)$$

We assume the probability of an edge between two nodes in layer l belonging to stratum s can be modeled as a Bernoulli random variable, based on the community membership of the nodes. In particular, $p(A_{ij}^l = 1) \sim \text{Bernoulli}(\pi_{z_i^s z_j^s}^s)$.

Since \mathbf{Y} and \mathcal{Z} are both latent quantities, searching over all possible values quickly becomes intractable. To tackle this issue, we develop a two-phase algorithm that incorporates a clustering algorithm for choosing the best \mathbf{Y} . This greedy approach leads to a significant reduction for the size of the search space since only \mathcal{Z} must be statistically inferred. Specifically, during Phase I, we

infer an SBM for each layer in isolation, and we cluster together sets of layers that have similar SBM parameters. Using these results as an initial condition in Phase II, we develop an iterative method that jointly identifies layer-to-stratum and node-to-community assignments as well as the SBM parameters for each stratum. We provide a schematic of the algorithm in Fig. 2.2, and below we present the two-phase algorithm in detail.

Phase I. Phase I is comprised of two parts. First, we fit an SBM to each individual layer $l \in \{1, \dots, L\}$, which yields inferred SBM parameters $\hat{\pi}^l$ and node-to-community memberships $\hat{\mathbf{Z}}^l$. Then we cluster the layers based on the similarities of $\hat{\pi}^l$ and $\hat{\mathbf{Z}}^l$. To infer $\hat{\pi}^l$ and $\hat{\mathbf{Z}}^l$, we use the inference method described in (?). Here, the authors used a variational inference technique to approximate the maximum likelihood estimates for the stochastic block model parameters. For the set of L layers, this produces sets of SBM parameters for each layer, which we denote by $\hat{\Pi} = \{\hat{\pi}^1, \hat{\pi}^2, \dots, \hat{\pi}^L\}$ and $\hat{\mathcal{Z}} = \{\hat{\mathbf{Z}}^1, \hat{\mathbf{Z}}^2, \dots, \hat{\mathbf{Z}}^L\}$ (that is, at this stage of the procedure, each layer is temporarily treated as its own stratum). Note also that each $\hat{\mathbf{Z}}^l$ can be equivalently represented by vector $\hat{\mathbf{z}}^l$. Using the estimates $\hat{\pi}^l$ and $\hat{\mathbf{Z}}^l$ for a given layer, l , we can construct the corresponding adjacency probability matrix, $\hat{\theta}^l$, which is defined entry-wise by $\hat{\theta}_{ij}^l = \hat{\pi}_{\hat{z}_i, \hat{z}_j}^l$. Doing this for each layer results in a collection of adjacency probability matrices, $\hat{\Theta} = \{\hat{\theta}^1, \hat{\theta}^2, \dots, \hat{\theta}^L\}$.

Now, we seek an initial partition of layers into strata based on $\hat{\Theta}$. The goal is to identify S sets \mathcal{L}^s so that the matrices $\{\hat{\theta}^l\}$ with $l \in \mathcal{L}^s$ are close to one another, but they are distant from the remaining matrices, $\{\hat{\theta}^l\}$ with $l \in \mathcal{L} \setminus \mathcal{L}^s$. This is accomplished by treating each $\hat{\theta}^l$ as a feature vector and applying k -means clustering with S centers so as to identify S strata, \mathcal{L}^s . Note that S can be selected *a priori*, or approximated with a measure such as the gap statistic (117). This gives us an initial estimate $\hat{\mathbf{Y}}$ for \mathbf{Y} . Note that this procedure initially treats each layer as a separate stratum, but provides a principled agglomeration of layers into $S \leq L$ strata.

Phase II. After a first-pass approach for assigning layers to strata, we initialize an iterative phase to more effectively estimate layer-to-strata assignments as well as the model parameters. Specifically, we would like to find the consensus SBM for each strata—that is, the $K^s \times K^s$ matrix π^s and the $N \times K^s$ matrix \mathbf{Z}^s that maximize the likelihood of the observed layers in each stratum. We let $\mathcal{A}^s = \{\mathbf{A}^l\}$ for $l \in \mathcal{L}^s$ denote the collection of adjacency matrices corresponding to the L^s layers in stratum s .

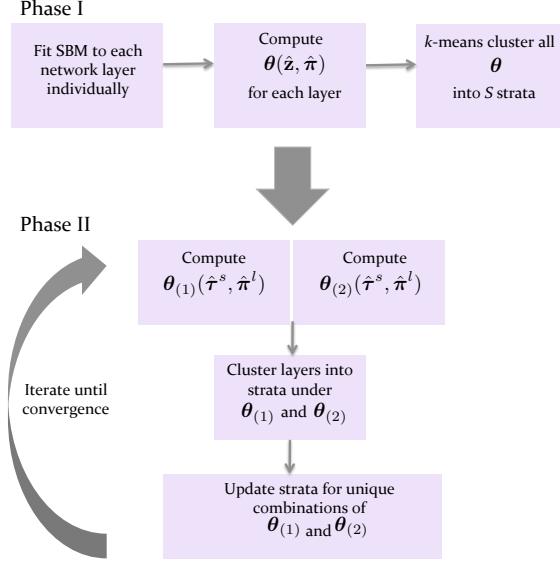


Figure 2.2: Schematic illustration of our algorithm: Our algorithm for fitting an sMLSBM is broken up into two phases: an initialization phase to cluster layers into strata, and an iterative phase that allows learning of node-to-community and layer-to-strata assignments.

We now proceed to maximize the likelihood in each stratum, by extending the framework of Ref. (32) to a multilayer context. Note that this is similar to Ref. (53), except that we are not aiming to infer an SBM probability matrix for each layer, individually. In particular, the complete-data log-likelihood for stratum s can be written as,

$$p(\mathcal{A}^s, \mathbf{Z}^s) = p(\mathcal{A}^s | \mathbf{Z}^s)p(\mathbf{Z}^s), \quad (2.2)$$

where

$$p(\mathcal{A}^s | \mathbf{Z}^s) = \prod_{l \in \mathcal{L}^s} \prod_{i < j} \prod_{mn} \pi_{mn}^{s A_{ij}^l} (1 - \pi_{mn}^s)^{(1 - A_{ij}^l)}. \quad (2.3)$$

To write $p(\mathbf{Z}^s)$, it is helpful to introduce a new parameter α_m^s that represents the probability that a randomly-selected node in stratum s belongs to community m , i.e. $\alpha_m^s = p(Z_{im}^s = 1)$.

Note that $\sum_m \alpha_m^s = 1$. Using this parameter, we can write

$$p(\mathbf{Z}^s) = \prod_i \prod_m \alpha_m^s (Z_{im}^s). \quad (2.4)$$

It follows that the complete-data log-likelihood for the adjacency matrices representing the layers in stratum s can be expressed as,

$$\begin{aligned}
\log P(\mathcal{A}^s, \mathbf{Z}^s) &= \log(P(\mathbf{Z}^s)) + \log(P(\mathcal{A}^s | \mathbf{Z}^s)) \\
&= \sum_i \sum_m Z_{im}^s \log(\alpha_m^s) \\
&\quad + \sum_{l \in \mathcal{L}^s} \sum_{i < j} \sum_{mn} A_{ij}^l \log(\pi_{mn}^s) \\
&\quad + \sum_{l \in \mathcal{L}^s} \sum_{i < j} \sum_{mn} (1 - A_{ij}^l) \log(1 - \pi_{mn}^s).
\end{aligned} \tag{2.5}$$

Problems of this variety that involve the need to compute maximum likelihood estimates with incomplete data are typically addressed with the expectation maximization (EM) framework (38). Doing so requires the ability to compute $P(\mathbf{Z}^s | \mathcal{A}^s)$; however, Ref. (32) showed that it is intractable to calculate the conditional distribution for the single-layer network case. To address this challenge, we use a variational approximation, analogous to approaches in (53; 15; 32). In general, a variational approximation seeks to optimize a lower bound on the log-likelihood. To do this, we first approximate the conditional distribution, $P(\mathbf{Z}^s | \mathcal{A}^s) \approx R_{\mathcal{A}^s}$, where

$$R_{\mathcal{A}^s}(\mathbf{Z}^s) = \prod_i h(\mathbf{Z}_{i\cdot}^s; \boldsymbol{\tau}_{i\cdot}). \tag{2.6}$$

Here, matrix $\boldsymbol{\tau}^s$ contains entries τ_{im}^s that approximate the probability that node i belongs to community m in stratum s . Further, function $h(\cdot)$ represents the multinomial distribution, with parameters, $\{\boldsymbol{\tau}_{im}^s\}$ for $m \in \{1, \dots, K^s\}$. Using this, we define the variational approximation as

$$\mathcal{J}(R_{\mathcal{A}^s}) = \ell\ell(\mathcal{A}^s) - \text{KL}(R_{\mathcal{A}^s}(\mathbf{Z}^s), P(\mathbf{Z}^s | \mathcal{A}^s)), \tag{2.7}$$

where $\ell\ell$ is log likelihood and KL is the Kullback-Leibler divergence.

Through maximizing $\mathcal{J}(R_{\mathcal{A}^s})$, we minimize the KL divergence between the true conditional distribution, $P(\mathbf{Z}^s | \mathcal{A}^s)$, and its approximation, $R_{\mathcal{A}^s}(\mathbf{Z}^s)$. Moreover, we follow the derivation in Ref. (?) and rewrite $\mathcal{J}(R_{\mathcal{A}^s})$ as

$$\begin{aligned}
\mathcal{J}(R_{\mathcal{A}^s}) &= \sum_i \sum_m \tau_{im}^s \log(\alpha_m^s) \\
&+ \sum_{l \in \mathcal{L}^s} \sum_{i < j} \sum_{mn} \tau_{im}^s \tau_{jn}^s [A_{ij}^l \log(\pi_{mn}^s)] \\
&+ \sum_{l \in \mathcal{L}^s} \sum_{i < j} \sum_{mn} \tau_{im}^s \tau_{jn}^s [(1 - A_{ij}^l) \log(1 - \pi_{mn}^s)] \\
&- \sum_i \sum_m \tau_{im}^s \log(\tau_{im}^s).
\end{aligned} \tag{2.8}$$

We can now differentiate $\mathcal{J}(R_{\mathcal{A}^s})$ with respect to each parameter—while using Lagrange multipliers to enforce constraints (i.e. probabilities summing to 1)—to compute the updates. Doing so yields the following, where the hat notation symbolizes the current best estimate for the given parameter:

$$\hat{\alpha}_m^s = \sum_i \hat{\tau}_{im}^s / N, \tag{2.9}$$

$$\hat{\pi}_{qt}^s = \frac{\sum_{l \in \mathcal{L}^s} \sum_{i < j} \hat{\tau}_{im}^s \hat{\tau}_{jn}^s A_{ij}^l}{\sum_{l \in \mathcal{L}^s} \sum_{i < j} \hat{\tau}_{im}^s \hat{\tau}_{jn}^s}, \tag{2.10}$$

$$\hat{\tau}_{im}^s \propto \hat{\alpha}_m^s \prod_{l \in \mathcal{L}^s} \prod_{i < j} \prod_n [\hat{\pi}_{mn}^{s,A_{ij}^l} (1 - \hat{\pi}_{mn}^s)^{1 - A_{ij}^l}]^{\hat{\tau}_{jn}^s}. \tag{2.11}$$

To find the best estimates for $\hat{\tau}^s$ and $\hat{\pi}^s$, we alternate between updating $\hat{\tau}^s$ and $\hat{\pi}^s$ until convergence. When convergence has occurred, we refer to the resulting estimates as the consensus $\bar{\tau}^s$ and $\bar{\pi}^s$ for stratum s . Similarly, \bar{Z}^s represents the consensus indicator matrix of node-to-community assignments computed from $\bar{\tau}^s$. Note that we use the bar notation to reflect that the particular parameter estimate is for a stratum, rather than for an individual layer.

Since $\bar{\tau}^s$ and $\bar{\pi}^s$ are computed in terms of each other, we can use one of the consensus parameters to compute the other parameter in individual layers. In particular, using the fixed node-to-community assignments from $\bar{\tau}^s$, we compute the maximum-likelihood SBM parameters for a particular layer l , which we denote with a tilde and hence, $\tilde{\pi}^l$ and $\tilde{\tau}^l$. Similarly, for fixed $\bar{\pi}^s$, we compute the node-to-community assignments $\tilde{\tau}^l$. Such estimates allow us to determine whether or not the stratum consensus estimates are accurate estimates for the SBMs of individual layers of the stratum. More importantly, as we shall now describe, these layer-specific estimates allow us to design an iterative

algorithm that allows for alternating between learning the node-to-community and layer-to-stratum assignments.

To this end, we represent each layer by the adjacency probability matrix, which we compute two different ways: letting $\theta(\tau, \pi)$ represent the adjacency probability matrix specified by τ and π , we define

$$\theta_{(1)}^l = \theta^l(\bar{\tau}^s, \tilde{\pi}^l), \quad (2.12)$$

$$\theta_{(2)}^l = \theta^l(\tilde{\tau}^l, \bar{\pi}^s) \quad (2.13)$$

Note that the first definition uses the strata-consensus estimate for τ^s and a layer-specific estimate for π^s , whereas the latter uses a layer-specific estimate for τ^s and the strata-consensus estimate for π^s .

During Phase I, we identified strata by clustering the adjacency probability matrices for the L layers using the k -means algorithm. We employ a similar procedure here, but instead of clustering L matrices, we now cluster $2L$ matrices, since each layer is represented in two different ways. Moreover, clustering these $2L$ matrices yields two cluster assignments for each layer. Typically, both representations of a particular layer will receive identical cluster assignments—that is, for a given l , $\theta_{(1)}^l$ and $\theta_{(2)}^l$ are assigned to the same cluster, or strata. However, an interesting case arises when the two representations induce different stratum assignments for a given layer, because this implies that there is disagreement between $\theta_{(1)}^l$ and $\theta_{(2)}^l$, which implies uncertainty in the strata assignment of that particular layer l . Because our iterative algorithm requires each layer to be assigned to a single stratum (i.e., we do not allow for mixed membership of layers into strata), layers with mixed membership according to $\theta_{(1)}^l$ and $\theta_{(2)}^l$ must be dealt with in some way. To account for these situations, we define additional strata for each combination of membership that arises. For example, if there are several layers $\{l\}$ that are clustered into stratum 1 according to $\theta_{(1)}^l$ and stratum 2 according to $\theta_{(2)}^l$, then we define a new stratum that contains only these layers. We note that there exists a variety of options for handling layers with such mixed membership after applying k -means clustering to $\theta_{(1)}^l$ and $\theta_{(2)}^l$ (e.g., one could assign such a layer to a stratum at random); however, we leave open for future work the exploration of these other options.

After a single pass of Phase II, which requires layer-to-strata assignments (which can be encoded by vector y) as input, the algorithm yields (ideally) improved layer-to-strata assignments (as well as consensus estimates for the SBM parameters of the strata, $\bar{\tau}^s$ and $\bar{\pi}^s$). Therefore, Phase II involves

iterating the above procedure until the layer-to-strata assignments do not change. We note that in principle, it is possible for new strata to arise in each iteration (i.e., because we create strata to avoid mixed membership of layers), and this can allow the number of strata to grow with each iteration; however, we did not observe this issue in any of our synthetic or real data experiments. As we will show in the following section, convergence is typically observed after just a few iterations (e.g., see, for example, the second row of Fig. 4). If such an issue arises, it may be helpful to bound the number of iterations in Phase II.

2.7 Synthetic Examples

Here, we demonstrate the performance of sMLSBM on synthetic networks.

2.7.1 Comparison of sMLSBM to other SBM Approaches

To demonstrate a situation where the sMLSBM framework has a clear advantage over other models, we designed a synthetic experiment and compared the results to two different SBM approaches: i) fitting a single SBM to all of the layers (denoted “single SBM”), and ii). fitting a stochastic block model to each layer individually (denoted “single-layer SBM”). We generated a multilayer network, where each layer has $N = 128$ nodes, $K = 4$ communities and an expected mean degree of $c = 20$ (i.e., every network layer is expected to contain $cN/2 = 1280$ undirected edges). We specified an sMLSBM with $S = 3$ strata and 10 layers per strata, which resulted in $L = 30$ total layers. We defined π^s for each stratum s in terms of two parameters, p_{in}^s and p_{out}^s , which give the within-community edge probabilities and between-community edge probabilities, respectively. That is, we define $\pi_{mn}^s = p_{in}^s$ when $m = n$ and $\pi_{mn}^s = p_{out}^s$ when $m \neq n$. It follows that the expected mean degree is given by $c = N(p_{in}^s + (K - 1)p_{out}^s)/K$. In our experiment, we select the following SBM parameters: $(p_{in}^1, p_{out}^1) = (0.6, 0.0083)$; $(p_{in}^2, p_{out}^2) = (0.4, 0.075)$; and $(p_{in}^3, p_{out}^3) = (0.125, 0.167)$. In Fig. 3(A), we show an example network layer from each strata. Nodes are colored by their community assignments in stratum 1. Note that the node-to-community assignments are different in each stratum and that the extent of block structure decreases from stratum 1 to stratum 3.

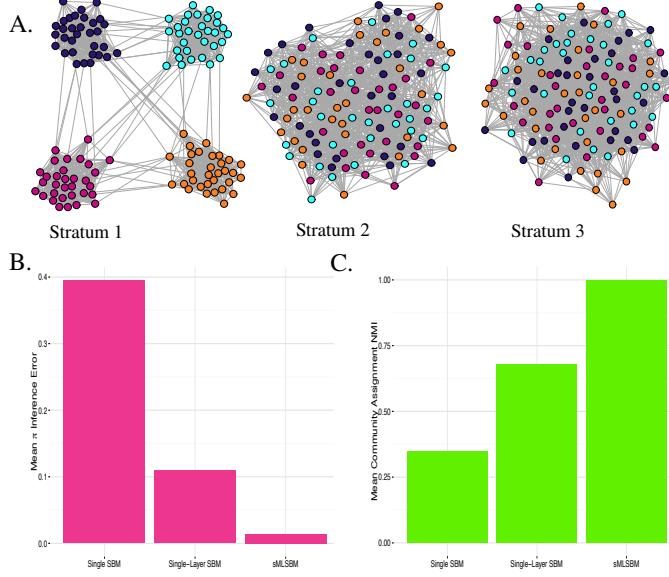


Figure 2.3: Synthetic experiment comparing sMLSBM to other SBMs. **A.** We specified a model with $S = 3$ strata and $L = 10$ layers per stratum. A representative layer from each stratum is plotted. Note that nodes in all networks are colored according to their community membership in stratum 1. Each network has $N = 128$ nodes, $K = 4$ communities and mean degree, $c = 20$. The p_{in}^s parameters for $s = 1, 2$ and 3 are $0.6, 0.4$ and 0.25 , respectively. Corresponding values of p_{out}^s were selected to maintain the desired expected mean degree, $c=20$. **B.** We fit 3 types of models to the 30 network layers: i) single SBM: fitting a single SBM to all of the layers; ii) single-Layer SBM: fitting an individual SBM to each layer; and iii) sMLSBM: identifying strata and fitting an SBMs for each strata. Each model yields an estimate $\bar{\pi}^{sl}$ for the true SBM of each layer l , which is denoted π^l . Here sl denotes the inferred strata for layer l . On the vertical axis we plot the mean ℓ_2 norm error $\|\text{vec}(\pi^l) - \text{vec}(\bar{\pi}^{sl})\|_2$. **C.** For each of the three models, we computed the normalized mutual information (NMI) between the true node-to-community assignments \mathbf{z}^l and the inferred values $\bar{\mathbf{z}}^{sl}$.

In order to compare the accuracy of fit for the three models—single-layer SBM, single SBM and sMLSBM—we quantify the inference accuracy of the SBM parameters, $\bar{\pi}^{y_l}$, and community assignments, \bar{Z}^{s_l} . First, for each layer and each model, we quantified the error (ℓ^2 norm) between $\text{vec}(\bar{\pi}^{y_l})$ and its true value, $\text{vec}(\pi^l)$. Note that $\text{vec}(\mathbf{X})$ is the $\frac{K(K+1)}{2}$ length vector representing the lower triangle of the matrix \mathbf{X} . Moreover, to quantify error, we compute $\|\text{vec}(\pi^l) - \text{vec}(\bar{\pi}^{s_l})\|_2$. We note that this error is well-defined because we identify $K = 4$ communities for all layers and all models. The mean error across layers under each model are shown in Fig. 3(B). In this example, sMLSBM outperforms the two other models. Second, we computed for each layer the mean normalized mutual information (NMI) (30) between the true node-to-community assignments, \mathbf{z}^l , and the inferred values, $\bar{\mathbf{z}}^{y_l}$, under each model. In other words, for each layer, we compute, $\text{NMI}(\mathbf{z}^l, \bar{\mathbf{z}}^{y_l})$. Figure 3(C) shows the mean NMI for community assignments across layers. Indeed, the effects of fitting an incorrect model to a collection of layers in terms of ability to effectively estimate SBM parameters and community assignments is apparent. In particular, fitting a single SBM model results in both larger mean inference and community assignment error, compared to fitting single-layer SBMs and 3 strata sMLSBM. In other words, sMLSBM provides an efficient clustering into strata only when the layers are indeed related (i.e. generated from the same SBM), otherwise each layer is a stratum on its own.

2.7.2 Synthetic Experiment with Two Strata

Next, we further explored the performance of our algorithm (see Sec. ??) for inferring an sMLSBM under various situations: 1) in comparison to baseline clustering methods; 2) in response to an increase in the number of layers; and 3) under variations in levels of detectability. Specifically, we designed synthetic experiments in which we generated multilayer networks with either $L = 10$ or $L = 100$ layers. Every multilayer network contained $S = 2$ strata (each having $K^1 = K^2 = 4$ communities), and in each layer there were $N = 128$ nodes (each having an expected mean degree of $c = 16$). Note that in this example both strata have the same node-to-community assignments. The strata were fixed to be the same size, $L^1 = L^2 = L/2$. Similar to the experiment described in Sec. 2.7.1, the SBM parameters were constructed using p_{in}^s and p_{out}^s . Since we have already specified the expected mean degree, these parameters must satisfy the constraint $c = N(p_{in}^s + p_{out}^s)/2$ for both strata. In all simulations, we fixed the SBM parameters of the first strata as $(p_{in}^1, p_{out}^1) = (.1836, .1055)$. It

is also convenient to define the quantity, $N(p_{in}^1 - p_{out}^1) = 10$, which relates to the detectability of communities (36). For example, the ability to detect community structure in a given layer and/or strata is, in general, expected to improve with increasing $N(p_{in}^s - p_{out}^s)$. For the second strata, we allow $N(p_{in}^2 - p_{out}^2)$ to vary.

We present results for this experiment in Fig. 4, wherein the left and right columns give results for $L = 10$ and $L = 100$, respectively.

Symbols in each plot represent the mean over 50 multilayer networks, and error bars show standard error. In each plot, the vertical dotted line indicates $N(p_{in}^2 - p_{out}^2) = 10$, which represents the point where the two strata are indistinguishable since $(p_{in}^1, p_{out}^1) = (p_{in}^2, p_{out}^2)$. In Fig. 4(A), we show the NMI between the true layer-to-strata assignments and those inferred by sMLSBM, or $\text{NMI}(\mathbf{y}, \hat{\mathbf{y}})$. As a baseline, we compare sMLSBM results to directly clustering the layers' adjacency matrices using the k -means algorithm with $K = 2$. We consistently observe higher NMI as a result of sMLSBM compared to k -means. More interestingly is the case with $L = 100$, where both k -means and sMLSBM perform at least moderately well at partitioning layers into strata before the point where the strata are indistinguishable. In Fig. 4(B), we plot the number of iterations (NOI) required for Phase II of our algorithm to converge. We observe that as the number of layers in the network increases, so does the number of required sMLSBM iterations. Moreover, the peaks in panel B. correspond to the sudden jumps in strata NMI.

Finally, in Fig. 4(C) we show the quality of node-to-community assignments by plotting the NMI between the true and inferred node-to-community assignments as described in Sec. 2.7.1. Note that stratum 1 here represents the stratum where the majority of layers were generated from model S^1 and analogously for stratum 2. Therefore, when the strata NMI is low (panel A.), we see poorer community detection results than expected, as layers get incorrectly mixed. As the strata NMI increases, layers from the same model are assigned together and the communities NMI stabilizes. Finally, by comparing the results for $L = 100$ to those for $L = 10$, we observe an increase in number of layers, L , generally leads to an improvement in community detection and strata identification.

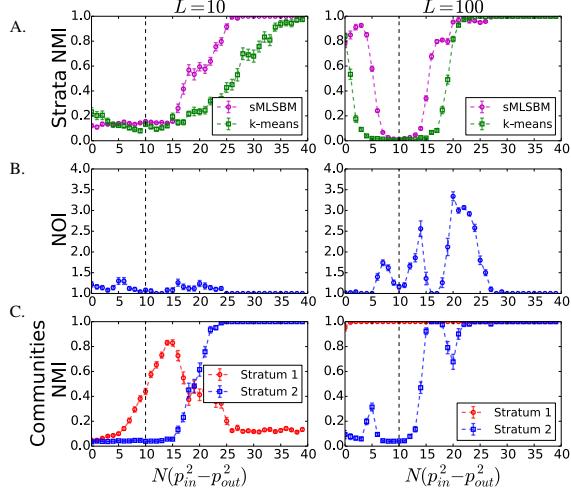


Figure 2.4: Synthetic experiment with two strata. We conducted numerical experiments with multilayer networks with $N = 128$ nodes, mean degree $c = 16$, $S = 2$ strata and $K^1 = K^2 = 4$ communities. The networks contained either $L = 10$ (left column) or $L = 100$ layers (right column), which were divided equally into the two strata. For stratum 1, we fixed the quantity $N(p_{in}^1 - p_{out}^1) = 10$, which fully specifies (p_{in}^1, p_{out}^1) since setting $c = 16$ also constrains these parameters. In contrast, we vary $N(p_{in}^2 - p_{out}^2)$. **A.** As a function of $N(p_{in}^2 - p_{out}^2)$, we plot the mean NMI to interpret the ability of sMLSBM to recover the true layer-to-strata assignments. We compare the performance of sMLSBM (purple curve) to generic k -means clustering (green symbols) of adjacency matrices. **B.** We plot the mean number of iterations (NOI) required for Phase II of our algorithm to converge. **C.** Finally, we measure the quality of node-to-community assignment results by plotting the mean NMI between the true node-to-community assignments and those inferred with sMLSBM in stratum 1 (red symbols) and stratum 2 (blue symbols).

2.8 Human Microbiome Project Example

As an application of sMLSBM, we consider correlation networks constructed from data from the Human Microbiome Project (121). For various sites on the body, the human microbiome project has successfully collected multiple human samples in order to better understand interactions between bacterial species. In this context, network inference is particularly interesting, as such methods aim to capture the relationships between various organisms. Microorganisms exhibit intricate ecologies within the gut of their human host and particular body sites have been shown to possess characteristic interactions. Further, certain interactions between microbes can often be associated with particular health and disease states (42). Microbiome data is typically collected through metagenomic sequencing and reads are further binned into groups, known as operational taxonomic units (OTUs), to represent particular organisms. The nature of this count-based sequencing data makes network inference challenging, and is thus an interesting field in itself. To demonstrate the potential use for sMLSBM in the context of the human microbiome, we applied our algorithm for learning sMLSBMs to multilayer networks constructed from the SparCC (45) network inference method.

SparCC is a correlation network inference method that aims to approximate the linear Pearson correlation between components in a system. This method performs favorably, as it accounts for the extent of diversity in the microbial community, which plays a significant role in detecting valid interactions. Furthermore, networks are constructed with the assumptions that the number of components in the system (e.g. OTUs) is large and that the correlation network should be sparse. As supplemental data in Ref. (45), the authors provided their inferred microbial interaction networks for 18 sites in the human body, using the sparse, SparCC framework. The edges in these networks have positive and negative real-valued weights, based on the results of SparCC inference. In this analysis, we converted the SparCC networks into binary adjacency matrices by allowing a link only if the SparCC edge-weight between two OTUs was at least 0.15 (chosen as a value close to 0.2, given in Ref (45)). To convert the 18 single-layer networks corresponding to species interactions in 18 body sites, we identified the collection of nodes (OTUs) that participated in at least two layers in terms of having at least one connecting edge weight value in the layer above the 0.15 threshold. This resulted in $N = 213$ unique OTUs (nodes) for our multilayer network analysis. We emphasize that restricting attention to nodes that participate in multiple layers was a choice we made in our

focus on identifying common community structures across layers, to demonstrate the accuracy in the algorithm and inference procedures of sMLSBM. A more biologically-relevant treatment of this dataset should of course consider domain-specific expertise in formulating a network representation appropriate to the question at hand.

We inferred an sMLSBM for the multilayer network and chose to show results for $S = 6$ strata. That is, this selection leads us to find 6 clusters of body sites such that the microbiomes are similar between sites in the same cluster but differ from microbiomes at sites in the remaining clusters. We indicate these 6 strata with colored boxes in Fig. 5. We note that due to the stochasticity of k-means in our algorithm, the communities and strata fit by sMLSBM can vary from one realization to the next. The shown strata assignments reflect those observed to yield the highest log-likelihood.

2.8.1 Comparison of sMLSBM to multilayer network reducibility

To gauge the performance of our method, we compared our strata membership results to the hierarchy obtained as part of the reducibility method developed in (34). To do this, we followed the following steps:

1. Compute the normalized Laplacian matrices for each of the 18 body site networks;
2. Compute the eigenvalues for each normalized Laplacian matrix;
3. Use these eigenvalues to compute the Von Neumann entropies for individual layers and pairs of layers;
4. Use the Von Neumann entropies to compute Jensen-Shannon distances between pairs of networks; and
5. Perform hierarchical clustering using the Jensen-Shannon distances and Ward linkage.

We show the results of this hierarchical clustering with a dendrogram in Fig. 5, which are in very good agreement with the sMLSBM results. However, as expected, we observe slight differences, since these methods cluster layers based on different criteria; in particular, sMLSBM partitioning reflects similarity only in community structure.

The results of both methods are relatively faithful to body regions in terms of groups of body

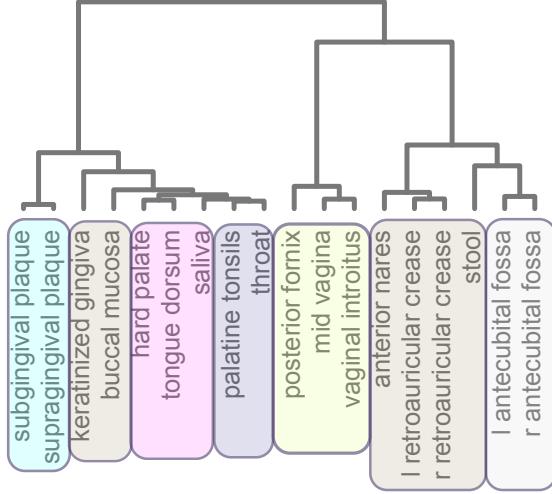


Figure 2.5: Comparison of sMLSBM on the OTU interaction networks (45) for each of the body sites to a reducibility hierarchy (34). As described in the text, we consider a multiplex network with $L = 18$ layers and $N = 213$ nodes, which we group here into $S = 6$ strata, while the dendrogram was generated by the method employed as the precursor to the reducibility framework. Colored boxes around the leaves of the dendrogram designate the body site to strata assignments obtained with sMLSBM.

sites that are spatially proximal. The only exception to this observation is the brown-colored stratum in Fig. 5, which is comprised of some seemingly unrelated body sites. While this grouping may not be intuitive, there is biological evidence to explain its plausibility. Specifically, Ref. (41) offers a state-of-the-art clustering of body sites based on biological expertise. Here, the authors have advanced understanding of microbial community composition through the application of a multinomial mixture model to define community types to characterize body sites. In particular, each sample collected through the Human Microbiome Project was assigned to 1 of 4 community types. They then quantified relationships between body sites using the p-value from a Fisher exact test on the membership of samples to community types. Similar to what we observe in the brown-colored stratum, the authors of (41) found a surprising correlation between samples from stool and oral cavity, which is reflected in our result.

2.8.2 Generating samples from the fitted sMLSBM

In Fig. 6, we illustrate network layers for 4 of the 6 strata that we identify to highlight one advantage of having a probabilistic generative model for microbial composition shared in subsets of body sites.

Specifically, each row provides information about the network layers and their fitted sMLSBM model for a particular stratum. Each grid in the figure represents the binary adjacency matrix encoding interactions between OTUs: a colored dot at position (i, j) indicates the existence of an edge (i, j) in the corresponding network layer. In the first column of each row is a sample network generated with the learned SBM parameters of that stratum, $\bar{\pi}^s$ and \bar{Z}^s . Columns 2 and 3 show two representative network layers within the stratum. Note that while some strata have more than two members, for illustrative purposes we only show two example layers. It is easy to see the very similar block structure between all networks in a given row, corroborating the usefulness of the sMLSBM approach. Finally, we highlight the usefulness of fitting sMLSBM to this multilayer network as each stratum elucidates a mechanistic understanding of the relationship between groups of OTUs, which could inspire further biological understanding or inquiry.

2.9 Concluding remarks for sMLSBM

We developed a novel model for multilayer stochastic block models (MLSBMs) and an associated algorithm to jointly partition layers into strata and nodes into communities. Our model assumes that layers belonging to a stratum have community structure following the same underlying SBM. To fit sMLSBM to a multilayer network, and more-specifically, a multiplex network, we iteratively alternate between rearranging layer-to-strata assignments and updating the model parameters for each stratum. Having multiple networks within a stratum—hence multiple realizations from some underlying model—helps to make inference more accurate. Particularly, more accurate assignments of nodes-to-communities within a stratum leads to improved estimation of SBM probability parameters, and vice versa. We have shown for multiplex networks with several strata (e.g., see Fig. 3) that inaccuracies can arise if one attempts to fit a single SBM to the network or study the network layers in isolation. In contrast, our model allows for an understanding of the similarities between layers in a network, in terms of their community structure.

The ability to identify strata within collections of network layers holds promise in numerous applications. One motivating application is network reducibility, whereby one compresses a multilayer network by aggregating similar layers (34). We stress that although reducibility is a closely related pursuit, it is fundamentally different from our co-clustering pursuit of simultaneously identifying

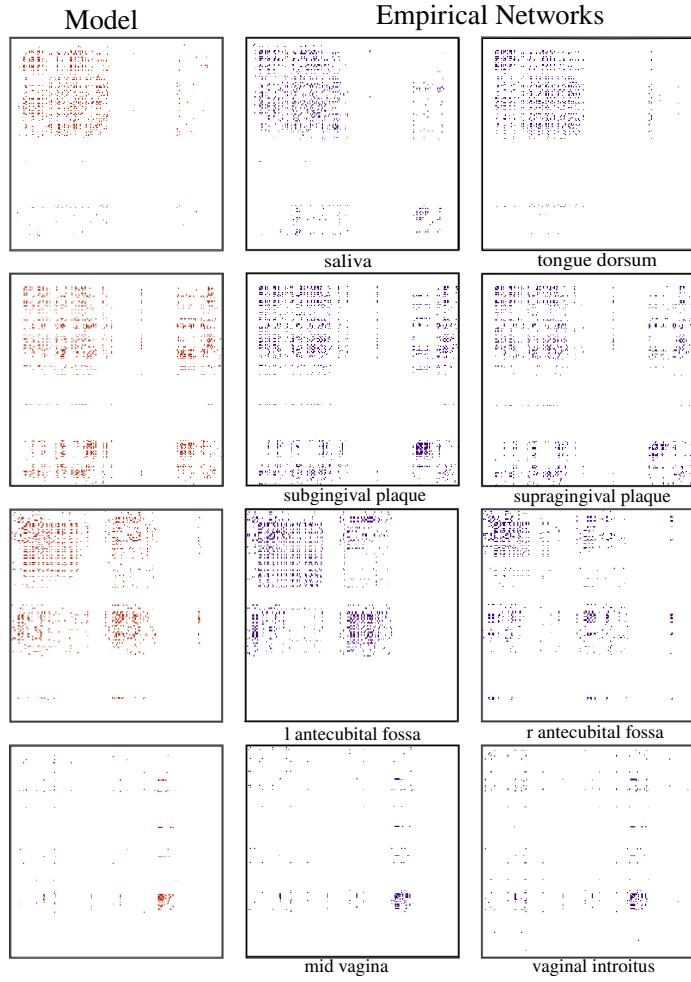


Figure 2.6: Visualization of Strata in SparCC Networks. We visualize the adjacency matrices for SparCC networks that encode microbiome interactions at body sites. In each panel, a colored dot at position (i, j) indicates the existence of an edge (i, j) in the corresponding network layer. The four rows correspond to four different strata. In column 1, we show a sample network generated from the SBM parameters, $\bar{\pi}^s$ and \bar{Z}^s , that we inferred for that stratum. In Columns 2 and 3, we show SparCC networks from that particular stratum. Note the strong similarity across each row.

communities and strata. In particular, our approach does not provide a method for aggregating layers. Instead, sMLSBM compresses the network information in the sense that the learned SBM parameters represent a consensus for each stratum, and those consensus parameters can be used to generate a representative sample network for that stratum. For applications in which layer aggregation is sought, there are a variety of ways to aggregate layers in a strata. See, for example, Ref. (116), where the authors explore the effects on community structure for different aggregation methods. We highlight that the sMLSBM modeling approach is appropriate in situations where one seeks a generative model for community structure, and it may be particularly appropriate when application-specific evidence suggests that subsets of networks have characteristic differences in community structure.

Our comparison of sMLSBM to the reducibility method of Ref. (34) (see Fig. 5) for the application of studying microbial interaction networks reveals several extensions to sMLSBM that could make the approach more accurate and applicable to a wider range of applications. First, the reducibility method (34) does not require networks to be undirected and unweighted, and it could be quite useful to extend the sMLSBM framework to weighted and directed networks following the extensions for single-layer SBMs, as developed in (11) and (126), respectively. It would also be useful to extend to degree-corrected and overlapping (i.e., mixed-membership) communities (61), as well as mixed membership of layers into strata. Additionally, the Human Microbiome example reveals some interesting biological questions that could facilitate the development of more advanced network tools. To construct the multilayer network, negative edges were thresholded away; however, antagonistic relationships between microbes are known to be important (132). Thus, it would be useful to develop a signed version of sMLSBM that allows edges to be either positive or negative.

The rise of a greater number of multilayer network datasets is providing the need for additional tools for the construction and analysis of such networks. The sMLSBM provides a new method to find signal in inherently noisy and complex network data.

2.10 Detectability in a single stratum

The development of sMLSBM motivated the analysis for how multiple layers can be collectively used to more accurately learn SBM model parameters in the single stratum case. That is, given a collection of sparse networks from a multilayer stochastic block model with one stratum, how can

the layers most accurately be combined to give the most accurate definition of community structure. We investigate these questions in *Enhanced detectability of community structure in multilayer networks through layer aggregation* (116). In particular, we studied the detectability limitations of the stochastic block model for a multilayer network with 1 stratum using random matrix theory techniques.

2.10.1 Investigating detectability in a multilayer network

Community structure detectability has gained considerable attention (68; 105; 56; 37; 82; 5) with a hope of being able to identify properties of networks and their corresponding adjacency matrices that reveal how prominent or easy-to-find the community structure is. A network with detectable community structure is thought to be one where multiple community detection algorithms would agree on common groups, and that nodes are not just being assigned to communities randomly, but instead exhibit straight-forward clustering patterns. Applying a community detection algorithm to a network with undetectable community structure might be dramatically different between algorithms, or may assign nodes to the biggest community or even all to the same community. It is particularly interesting to investigate this question in relation to a multilayer stochastic block model because we can generate samples from various models with different parameters and see if the community partition of the network agrees with the specified model. Previous work has previously been explored in networks with degree heterogeneity (103), hierarchical structure (95; 108), and in temporal networks (46), but not characterized in multilayer networks.

To study this in multilayer networks, we use random matrix theory to study a multilayer network generated from a stochastic block model, and enumerate ways that these layers can be *aggregated* or combined to most improve community structure. We show that the detectability limit vanishes with an increasing number of layers, L , and decays as $O(L^{-1/2})$ when we aggregate the the network layers, by taking the sum of their adjacency matrices. Further, we also explore the detectability limits of this aggregated summation of adjacency matrices that are thresholded to a binary adjacency matrix according to some value, \tilde{L} .

2.10.2 Studying detectability in two block networks

In this work, we study a 2 block multilayer stochastic block model. As seen in previous sections, each network layer has the same set of N nodes and parameterized by an N -length vector, \mathbf{z} specifying the node-to-community assignments and a 2×2 community probability connectivity matrix, $\boldsymbol{\theta}$. Further, we assume that the between probability connection probability is denoted by p_{out} , and that $\pi_{1,2} - \pi_{2,1} = p_{out}$. Similarly, we denote the within-community probability as p_{in} , so that $\pi_{1,1} - \pi_{2,2} = p_{in}$. Previous work has shown that for the large network limit, as $N \rightarrow \infty$, there is a solution to the detectability limit (37; 82), characterized by the solution curve (Δ^*, ρ) to

$$N\Delta = \sqrt{4N\rho}, \quad (2.14)$$

where $\Delta = p_{in} - p_{out}$ is the difference in probability and $\rho = (p_{in} + p_{out})/2$ is the mean edge probability. For a given value of ρ , the communities are only detectable (or correctly characterized) if $\Delta > \Delta^*$. Equation 2.14 was derived for sparse networks (i.e. constant ρN so that $\rho = O(N^{-1})$) and was obtained using both a Bayesian analysis (37) and random matrix theory (82).

In this work , we study the behavior of Δ^* for two methods of aggregating layers within a multilayer network of L layers, which we denote \mathcal{L} . We define the *summation* network, $\bar{\mathbf{A}} = \sum_{l \in \mathcal{L}} \mathbf{A}^l$. Note that, \mathbf{A}^l gives the adjacency matrix for network layer, l . We also define a family of *thresholded* networks, with unweighted adjacency matrices $\{\hat{\mathbf{A}}^{\tilde{L}}\}$ that are obtained by applying a threshold $\tilde{L} = \{1, \dots, L\}$ to the entries of $\bar{\mathbf{A}}$. Under this thresholding rule, $\hat{A}_{ij}^{\tilde{L}} = 1$ if $\bar{A}_{ij} \geq \tilde{L}$ and is 0 otherwise. We are particularly interested in the limiting cases when $\tilde{L} = L$ and when $\tilde{L} = 1$, which correspond to applying logical AND and OR operations to the original multilayer data $\{A_{ij}\}$, for a fixed pair of nodes (i, j) . We refer to these thresholded networks as the AND and OR networks, respectively.

2.10.3 Using random matrix theory to study detectability

Since node-to-community assignments, \mathbf{z} can be inferred with spectral method, random matrix theory (17; 83) is a useful approach for studying partitioning and phase transitions in detectability (i.e. node-to-community assignment accuracy) (82; 95; 108). Using this approach, phase transition in detectability correspond to the disappearance of gaps between eigenvalues (whose corresponding

eigenvectors reflect community structure) and bulk eigenvalues [which arise due to stochasticity and whose $N \rightarrow \infty$ limiting distribution is given by a spectral density $P(\lambda)$. The theory we develop in this work is based on the modularity matrix, $\bar{B}_{ij} = \bar{A}_{ij} - \rho L$ (88).

We first study Δ^* for the summation network. We analyze the distribution of real eigenvalues $\{\lambda_i\}$ of $\bar{\mathbf{B}}$ (in descending order). First, we describe the statistical properties of entries $\{\bar{A}_{ij}\}$, which are independent random variables following a binomial distribution with $P(\bar{A}_{ij} = A) = f(a; L, \pi_{z_i, z_j})$, where

$$f(a; L, p) = \binom{L}{a} p^a (1-p)^{L-a} \quad (2.15)$$

has mean Lp and variance $Lp(1-p)$. With sufficiently large variance in the edge probabilities, we find that the limiting $N \rightarrow \infty$ distribution of bulk eigenvalues for $\bar{\mathbf{B}}$ is given by a semicircle distribution,

$$P(\lambda) = \frac{\sqrt{\lambda_2^2 - \lambda^2}}{\pi \lambda_2^2 / 2} \quad (2.16)$$

CHAPTER 3

Network compression for community detection with super nodes

In practice, social and biological networks are quite large with hundreds of thousands or millions of nodes. For example, the SNAP (Stanford Network Analysis Project) network repository ((?)) houses various types of social and technological networks. For example, the amazon co-purchasing network has 548,000 nodes and 1,788,725 edges. Working with a network this size is overwhelming and makes computations slow and variable. In particular, we will show that community detection algorithms produce highly variable outputs on large networks or take a long time to run. In this chapter, we will introduce a pre-processing technique for networks to take the originally large network and reduce it to a smaller size, which enables analysis on a small network with a user-defined number of nodes or ‘super nodes’. This work is from our paper *Compressing Networks with Super Nodees* (114).

3.1 Super pixel pre-processing of images

Much of our motivation for this network pre-processing step for community detection is inspired by the image analysis literature. The identification of communities in networks is in some ways similar to multi-label image segmentation, which aims to partition a grid of pixels into contiguous regions corresponding to objects in the image. In this sense, each segmented region can be viewed as a community (25). To speed up segmentation for large images, a popular approach is to avoid computing segmentations at the pixel level and instead reformulate the segmentation problem based on larger-scale image primitives that are likely part of the same partition. Specifically, this can be accomplished by *super pixels* that aggregate pixels together in a way that faithfully adheres to image boundaries, maintaining or improving segmentation accuracy (6). The SLIC super pixel method



Figure 3.1: Superpixel pre-processing of an image. An image can be represented by a 1147×1147 grid of pixels (left). Representing the image with 600 super pixels (right), reduces the size of the image and hence the segmentation problem is to partition the set of 600 super pixels.

(6) chooses seed pixels across the image’s pixel grid to serve as the super pixel centers and then iteratively grows out and recomputes based on aggregation with neighboring pixels with similar visual features. This is shown in Figure 3.1, where if our task is to separate the grass from the dog, the input to the segmentation algorithm would be a 1147×1147 grid of pixels (left image). By representing the image with 600 super pixels (right image), we instead use this 600 super pixel representation as the input to a segmentation algorithm, which performs segmentation by partitioning the super pixels. From this segmentation on the super pixel representation, the labels can be mapped onto the full 1147×1147 image grid based on their super pixel assignment. We seek to perform the analogous task on networks, which is pre-processing the network into ‘super nodes’ before applying the community detection algorithm. Similar to a quality super pixel representation, we would like our super node representation to adhere to boundaries and produce accurate results. In this case, we refer to boundaries as communities that would have been obtained using the full network. Similarly, we gauge accuracy based on how similar the result obtained on the super node representation is to that on the full network. This task is more straightforward on an image because it is simply a grid of pixels, so it is easy to account for the spatial information, or the idea that neighboring pixels should be agglomerated into the same super pixel.

3.2 Super node pre-processing for networks

In this work, we seek to extend the super pixel methodology to large networks. In doing so, we designed experiments and validation metrics to measure the quality of our results. In images, there is

a natural notion of a ground truth segmentation that can be specified by a human. Because community detection is an inherently unsupervised analysis, we had to think carefully about quantities that we could compute to measure the quality of a super node representation.

3.2.1 Problem Formulation

For a network with N nodes that we will split into K communities, we seek to find a representation of the network with S super nodes optimizing the following two quantities. First, given the set $\mathbf{s} = \{s_1, s_2, \dots, s_S\}$ of S super nodes and K communities, $\mathbf{k} = \{k_1, k_2, \dots, k_K\}$ identified with the full network, we wish to minimize the under segmentation error,

$$U = \frac{1}{K} \sum_{k_i=1:K} \frac{[\sum_{s_j | s_j \cap k_i \neq \emptyset} |s_j|] - |k_i|}{|k_i|}, \quad (3.1)$$

where $|\cdot|$ represents the count or number of nodes in the indicated set.

We let \mathbf{z}^{Full} and \mathbf{z}^{SN} denote the node-to-community assignments for the full network and super node network representations, respectively. To compute the similarity between \mathbf{z}^{Full} and \mathbf{z}^{SN} , we use Normalized Mutual Information (NMI). (31) That is, for partitions \mathbf{z}^{Full} and \mathbf{z}^{SN} with p and q communities, respectively, with N the $R \times C$ contingency table matrix where N_{ij} gives the count of the number of shared nodes in community i in \mathbf{z}^{Full} and community j in \mathbf{z}^{SN} , the NMI between the two partitions is

$$\text{NMI}(\mathbf{z}^{\text{Full}}, \mathbf{z}^{\text{SN}}) = \frac{-2 \sum_i \sum_j N_{ij} \log \frac{N_{ij}N}{N_{i\cdot}N_{\cdot j}}}{\sum_i N_{i\cdot} \log \frac{N_{i\cdot}}{N} + \sum_j N_{\cdot j} \log \frac{N_{\cdot j}}{N}}, \quad (3.2)$$

where $N_{i\cdot}$ and $N_{\cdot j}$ are the marginal sums over the corresponding rows and columns and $N = \sum_i N_{i\cdot} = \sum_j N_{\cdot j} = \sum_{ij} N_{ij}$.

3.2.2 An opportunity for super nodes in community detection

While there are a variety of approaches to identify communities, in this paper we specifically examine how the compressed version of a network can be used in modularity maximization and likelihood maximization (through a stochastic block model). These agglomerative heuristics for both modularity and likelihood maximization simplify a computationally challenging task but can still be time consuming for large networks and often give rise to large variability in the partitions returned across

multiple runs of the algorithms. We seek to explore how a compressed network representation can improve these issues. Motivated by how this problem is approached for super pixels in images, we wish to define seed nodes in networks that can be used as a starting point to grow out ‘super nodes’ to define a new, smaller network upon which we apply standard community detection algorithms. Creating a direct analog of super pixels in networks is challenging because the inherent geometry of a network can be quite different from the grid layout of an image (where simple neighborhood structures such as 4- or 8-neighborhoods are typically used), and we need to ensure seeds are well distributed across the network. Further, while super pixels are largely constrained by the structure of the pixel grid (i.e. proximity between pixel pairs matter), their definition also incorporates extra image features to refine members of a super pixel set, whereas in network community detection we typically only have the edges of the network to work with. Finally, the performance of a super pixel representation of an image can be objectively validated from the quality of the corresponding segmentation result, with reference to human-specified objects in images; in contrast, community detection is typically an unsupervised, exploratory data analysis technique with limited available notions of ‘ground truth’ (129). As such, we must develop measures that can be used to validate the quality of the super node representation.

3.3 Background

3.3.1 Related Work

Our objective to define a smaller network of super nodes is a form of network compression. Several references have explored useful ways to compress networks,(74; 131; 47; 99) with Yang *et al.*, (131) and Peng *et al.* (99), using graph compression in the context of community detection. These compression approaches can either be classified as *network pre-processing* or *network size reduction*. Under these definitions, pre-processing refers to a method that uses all of the nodes to pre-partition the network or agglomerate nodes to form a smaller network of pre-agglomerated nodes or ‘super nodes’. Creating a super node representation of the network can assist in visualization, gives control over how many nodes to originally split the network into, and allows for the input of a pre-processed network into standard network analysis tools. Alternatively, in network size reduction, nodes are systematically removed and further analysis is performed on a smaller subnetwork. Such an approach

may be useful if one has prior knowledge of unimportant or redundant nodes. Two network pre-processing methods that define super nodes are explored by Yang *et al.* (131), and Lisewski *et al.* (74); but these approaches differ from our proposal in that they seek to define super nodes along with additional side information about relationships between node pairs. First, Lisewski *et al.*(74), describes ‘super genomic network compression’ to reduce the number of edges in a large protein interaction network. To do this, the authors identify ‘clusters of orthologous groups’ of proteins, or proteins that give rise to similar functions in different species and originated from a common ancestor. Members of an orthologous group are connected as a star network, with the center node as one member of the orthologous group. Furthermore, edges between orthologous groups are replaced by a single weighted link reflecting the pairwise group evolutionary similarity. Next, Yang *et al.* (131), defines super nodes by defining ‘must link’ and ‘cannot link’ constraints between pairs of nodes, agglomerating as many nodes as possible sharing must link constraints while being cautious about agglomerating nodes that cannot link. Alternatively, two approaches that perform network compression through network size reduction were presented in two works by Gilbert *et al.* (47), and Peng *et al.* (99). Gilbert *et al.*, introduce the ‘KeepAll’ method, (47) which seeks to prioritize a set of nodes according to their importance in the network and retain only the smallest set of additional nodes required for the induced subgraph of prioritized nodes to be connected. Results in this paper highlight its ability to remove redundant and noisy nodes that allow for clearer analysis of the original set of prioritized nodes. Finally, Peng *et al.*, (99) extract a smaller network through a k -core decomposition, and perform community detection on the subnetwork. While we also seek to perform community detection on a smaller version of the network, we seek to do this in the network pre-processing manner so that all nodes are effectively included as the input to the community detection algorithm, with flexibility to choose the number of super nodes or size to represent the network with. Given that the number of nodes in the k -core of a network decreases dramatically with an increasing k , there is not much flexibility in the scale or size of the network representation.

3.3.2 Validation metrics for a quality super node representation

The aims of this work are twofold: *First*, we seek an effective way to define a super node representation of a network that can then be used in standard community detection algorithms, such that the representation minimizes our defined under segmentation error and maximizes NMI with the

partition that would have been obtained using the full network. *Second*, we wish to highlight several benefits of using such a compressed representation of network in community detection. In particular, we show that a super node representation of the network accomplishes the following.

1. **Decreased runtime for community detection:** Even though recently developed heuristics for maximizing modularity (20) and fitting SBMs (96) are highly efficient relative to previous approaches for performing the same computational optimizations, these methods can still be time consuming for large networks. We aim to reduce runtime for large networks, moving most of the computational cost in practice from tasks scaling with the size of the network to alternatives scaling with the (much smaller) size of the super node representation.
2. **Decreased stochastic variability of community detection algorithm output:** In large networks, there is often significant variability across multiple runs of the same algorithm (employing computational heuristics to solve NP-Complete optimizations), as well as differences between various community detection algorithms. We expect applying community detection to a well-chosen super node representation to decrease the observed variability.
3. **High local agreement:** In defining super nodes, we inherently assume that the identified communities should agree with the local network connectivity in that members of a neighborhood should be more likely to have the same community assignment, provided that the super nodes were constructed to minimize aggregation across community boundaries.
4. **Consistent with communities found using the full network:** Despite the differences in line with the above features, the identified community structure should still be relatively similar to the distributions of results that would have been obtained through applying community detection to the full network.

To define the super node representation of an unweighted N -node network, we first select $S \ll N$ seed nodes through a 2-core decomposition (discussed further in Methods). We then agglomerate the remaining $N - S$ nodes with the seeds to create super nodes, and specify the network between these super nodes. Community detection can then be applied to the S -node network representation. Figure 3.2 visualizes this approach, with details provided in the Methods.

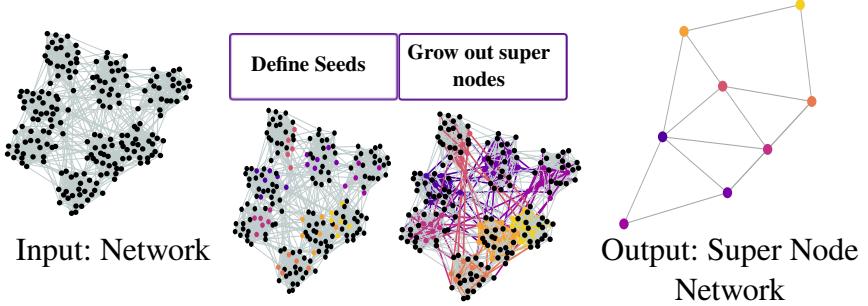


Figure 3.2: **Defining super nodes.** To define the super node representation of a network, we select S seeds and agglomerate local regions around them to create super nodes. This then leads to a new network with weighted edges between the S super nodes upon which community detection can be more efficiently applied.

3.3.2.1 Objectively Comparing Partitions on Possibly Different Scales

A challenge in directly comparing the community partitions on the full and super node network representations is the difference in scales between the partitions. For example, using the full network typically produces significantly more communities than under the super node representation. In an attempt to compare community partitions with similar size distributions in the subsequent experiments, we can adapt the scales obtained from the Louvain algorithm and SBM fitting.

In community detection with the Louvain algorithm, we identified comparable resolution parameters (controlling community size) to apply to the full network that would produce a size distribution agreeing as much as possible with the community partition in the super node network. We compute experimental results using both the default resolution parameter and the ‘matched’ parameter. While the default resolution parameter is $\gamma = 1$, in our analyses we computed partitions of the full network using several different $\gamma \in [0.05, 2.5]$. To choose the matched resolution parameter on the full network, we first find the community partition using the super node representation. For each partition, we then order nodes based on the sizes of the communities to which they belong. With this approach, all nodes from the same community are at the same position in the ordering. For each partition of the full network (at different resolution parameters), we then consider the similarity of this ranking with that from the super node communities, measuring this similarity by Kendall’s tau correlation. We identify the resolution parameter producing the highest Kendall’s tau correlation, referring to this resolution parameter as the ‘matched parameter’ in the remainder of the text, while we refer to the standard $\gamma = 1$ as the ‘default’ resolution parameter.

In fitting SBMs, we chose to fit a model with the same number of blocks that was found in the super node representation using the standard optimization and model selection strategies discussed in Ref. 96. We refer to the ‘matched’ version as that using the number of blocks identified by the model selection on the super node representation, while the ‘default’ result is obtained using the model selection strategy on the full network. In subsequent experiments, we compare both the ‘matched’ and ‘default’ versions to ensure our results are not artificially influenced by the scale of the community sizes.

3.4 Methods

We create a super node representation through three steps, outlined in Figure 3.2. First, we define seeds. Next, we ‘grow’ super nodes by assigning the remaining nodes to seeds. Finally, the network of super nodes is defined by agglomerating edges, and can then be used in a community detection algorithm.

3.4.1 Defining seeds

To define S seeds, we aim to identify a set of nodes S^* that are individually central to the network and to their communities, and that are well separated from one another. Such problems are related to influence maximization, where one identifies a small number of nodes in the network from which to effectively spread influence or diffuse messages across the network (60; 63). The most naive approach is to select nodes with highest degree, and this might be perfectly reasonable under various circumstances. Importantly, the selection of nodes with highest degree is computationally fast, requiring $O(M)$ operations, summing over the M edges to calculate the degrees of the N nodes.

At slightly higher computational cost, we employ the CoreHD algorithm, which nearly optimally identifies nodes in network decycling and dismantling (134). CoreHD recursively identifies the highest degree node in the 2-core (the maximal connected subgraph in which all nodes have degree at least 2 within the subgraph, obtained efficiently by recursively pruning away all nodes of degree less than 2). At each iteration, the removed node is added to our seed set and the now smaller 2-core and their degrees are recomputed. The difference between selecting highest degree nodes and CoreHD for our present task may be small, both in terms of result and computational cost. In particular,

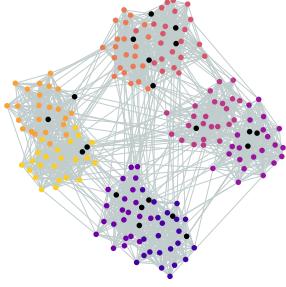


Figure 3.3: **Choosing seeds in a synthetic network.** The identification of 20 seeds with the CoreHD algorithm in a network generated from a stochastic block model with 8 communities. Seeds (black nodes) are well distributed across communities.

because we will only select $S \ll N$ seeds, there is reduced opportunity for the removals to lead to subgraphs with substantial differences between degree order in the graph and its 2-core [cf. selection of $O(N)$ removals in network dismantling]. Indeed, in our experience, simply selecting the highest degree nodes as the seeds often works well in practice. Because of the minimal extra computational cost for computing the 2-core, we use CoreHD for all of our results shown here. In Figure 3.3, we plot a sample network generated from the stochastic block model, with nodes colored according to community assignments and black nodes indicating seeds, showing the seeds are distributed across all 8 communities.

3.4.2 Grow Super Nodes Around Seeds

Once the set of seeds, S^* , is defined, we ‘grow’ them out agglomerating nearby nodes to build the super nodes. We formally define a super node as a subset of one or more nodes from the original network, \mathcal{X} . To do this, seeds are grown out to engulf nodes in increasing neighborhood orders until either all nodes are assigned to a super node center or until a user-defined number of neighborhood orders has been considered. The maximum order, o_{max} , can be specified to control the maximum order neighborhood to consider in building the super nodes. If after o_{max} , there are still unassigned nodes, the unassigned nodes are not used to build the new super node network and are all ultimately assigned to the same periphery community as they are not considered relevant to the network core. Depending on the number of chosen super nodes, S , the degree distribution of the original network and the quality of the chosen seeds at collapsing the network, different networks will require repeating the agglomeration process for different neighborhood orders if one wants to ensure every node is

Dataset (* indicates subgraphs)	# Nodes	# Edges
As22* (Internet)	22,801	48,270
Enron*	32,374	178,195
CMatter* (Condensed matter 2003 collab.)	17,816	83,337
Dblp* (com-DBLP)	150,801	639,330
Amazon* (com-Amazon)	77,463	209,887
Email (email-EuAll)	265,214	420,045
Stanford (web-Stanford)	281,903	2,312,497
Notre Dame (web-Notre Dame)	325,729	1,497,134
BrightKite (loc-BrightKite)	58,228	214,078

Table 3.1: Network data characteristics.

assigned to a super node. The output is a vector, \mathbf{s} of length N , which gives the node-to-super-node assignments for the nodes in the original network.

3.4.3 Create Network of Super Nodes

Finally, after growing the super nodes, we create a new network representation of the super nodes. To do this, we create a weighted network, \mathcal{W} , where each super node is a node and the weight of the edge between a pair of distinct super nodes is the total weight of edges in the original network \mathcal{X} between pairs of nodes assigned to the respective super nodes. For pairs of super nodes whose members have no edges between them in \mathcal{X} , there is no corresponding edge in \mathcal{W} . By definition, we construct \mathcal{W} with no self loops. Moreover, the produced super node network representation produces a weighted network where the edge weights are counts.

After applying community detection to the super nodes, their community assignments are mapped back to their constituent N nodes of the original network, \mathcal{X} . We denote this final N -length matched community assignment as \mathbf{z} . In experiments in the Results section, we consider the node-to-community assignment \mathbf{z}^{Full} obtained by applying community detection to the full network and the mapped result \mathbf{z}^{SN} obtained by applying community detection to the super node representation.

3.5 Results

To demonstrate the effectiveness of super nodes, we performed experiments to analyze the runtime, partition variability, alignment of communities with local network connectivity, and agreement with the communities obtained from the full network. We considered 9 unweighted network data sets (see Table 3.1) from the Stanford Network Analysis Project database(sna) (Enron, Amazon, Dblp, Email,

BrightKite, Stanford, Notre Dame) and Newman’s collection(new) (As22, CMatter). We treat all networks as undirected. To explore a range of network sizes, in some of these cases we used large subgraphs defined by the union of all nodes of degree ≥ 2 , their neighbors, and next nearest neighbors. We use the Louvain algorithm(tra) for modularity maximization (20) and the stochastic block model (SBM) inference(tia) described in Ref. 96. Since the super node representation ultimately produces a weighted network, where the edge weights are counts computed based on the original network, both of these community detection are able to accommodate these kinds of edge weights.

3.5.1 Overview of experiments

Because we consider a variety of comparisons between partitions under different community detection methods and network representations, we provide a schematic in Figure 3.4 of the comparisons we performed in Figures 3.5 and 3.7. In these comparisons, we use normalized mutual information (NMI) to quantify the similarity between a pair of partitions. In general, there are four possible combinations of community detection method/ network representation that can be applied to identify communities. First, there are two choices of community detection algorithm, Louvain algorithm or stochastic block model fitting. There are also two choices for network representation, which is to use either the full network or super node network representation. In Figure 3.4, we represent a single partition of the network by a circle. Due to the stochastic nature of the community detection methods, it is useful to consider multiple runs (i.e. partitions) of the algorithm. The circles in the diagram are colored by one of the four network/method combinations. We assume that under each network/method combination, we generate multiple partitions. Amongst all partitions, we perform our analyses on all pairs of network partitions generated that satisfy our comparison criteria. The first partition comparison we consider is shown in Figure 3.4 A., and seeks to quantify the similarity between a set of partitions generated with different network representations under the same community detection algorithm. The comparison computes $NMI(\mathbf{z}^{Full}, \mathbf{z}^{SN})$. For example, we could consider the pairwise similarity between the full network and super node representations with the stochastic block model. This comparison helps to understand how well the super node representation can be used to estimate the node-to-community assignments that could have been obtained using the full network. Next, we explore the inherent variability of community detection algorithms, which seems to especially arise among partitions of a large network. Figure 3.4 B. considers pairs of partitions under the same

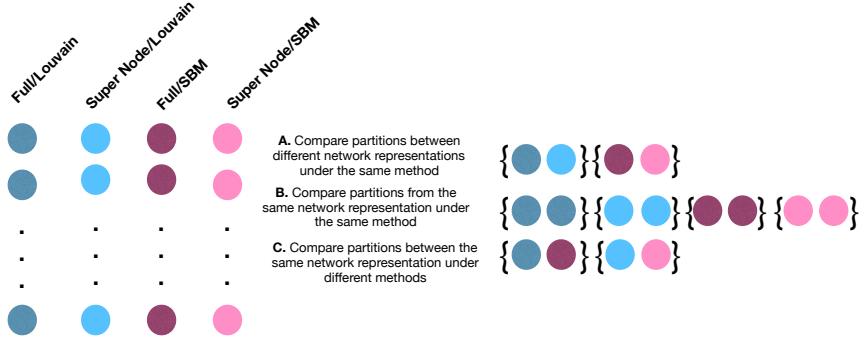


Figure 3.4: Schematic of possible partition comparisons. We outline the types of possible comparisons between partitions generated according to various combinations of network representation and community detection method. According to these comparison rules, we compute normalized mutual information (NMI) between all pairs of networks satisfying the comparison criteria. The colored circles in the schematic represent a single partition generated under the corresponding network representation and community detection algorithm combination. Circles are colored (in each column) by each of the four possible representation/community detection method combinations. In **A-C**, we outline the types of comparisons we perform in subsequent figures. **A.** To compare the usefulness of the super node representation in identifying communities retrieved using the full network, we compare pairs of networks with different representations under the same community detection algorithm. **B.** Due to the stochastic nature of both the Louvain algorithm and SBM fitting, this comparison seeks to quantify partitions generated under the same network representation and method. **C.** Finally, we consider pairs of partitions generated under the same network representation and different community detection algorithms.

network representation and community detection algorithm. For example, we may run the Louvain algorithm on the super node representation of the network multiple times and compare all pairs of partitions. Finally, in Figure 3.4 C., we consider pairs of partitions generated under the same network representation but with different methods. In one example, we may compare partitions of the full network representation, where one partition used the Louvain algorithm and the other is obtained by fitting an SBM. Since these methods are different by design, they should still capture sufficiently common structures.

3.5.2 Normalized mutual information and under segmentation error

First we measure the quality of the super node representation in terms of NMI and under segmentation error that were defined in equations 3.1 and 3.2. In Figure 3.5, we vary the number of super nodes and examine the normalized mutual information (A.) (equation 3.2) and log under segmentation error (B.) (equation 3.1) in each of the 9 networks. The curves represent the mean NMI (A.) and

mean under segmentation error (B.) over 5 super node representations, with the shaded area denoting standard deviation. Varying the number of super nodes between 100 and 600, the results generally indicate that as the number of super nodes increase, the network has 1) lower under segmentation error and 2) higher NMI (i.e. similarity) with the partition obtained using the full network. Each curve and line type in Figure 3.5 specifies whether community detection was performed using the Louvain algorithm or by fitting an SBM. To give some intuition about what value of NMI is considered good, we put our results in the context of the partition variability among 10 runs of the same community detection algorithm. In Figure 3.5 A., we indicate the mean pairwise NMI between multiple runs of the Louvain algorithm and SBM fitting with horizontal pink and gold lines, respectively. This comparison is that described in Figure 3.4 B. Since in most cases, the pairwise NMI between partitions is not 1, by increasingly adding super nodes, we can only expect to asymptotically approach the mean pairwise NMIs. between multiple runs of the same algorithm on the original full network representation. Randomly permuting the node-to-community assignment obtained under the super node representation 1000 times and computing the NMI with the full network (i.e. $NMI(\mathbf{z}^{Full}, \mathbf{z}^{SN})$) gives a mean NMI on the order of 0.01.

3.5.3 Run time Analysis

In practice, one of the most desirable properties of a super node representation of the network is the decrease in the run-time of community detection algorithms in comparison to using the full network. For each of the 9 networks, we recorded the runtime required to identify communities with the Louvain algorithm and stochastic block model inference procedure under the full a 500 node super node network representations (Figure 3.6). The Louvain algorithm is fast and scales well, at $O(M)$ per iteration for M edges, with its relative speed and high modularity values contributing to its popularity. While the reported runtimes may seem quite modest, in practice it is common to run many realizations of the algorithm (hundreds, thousands, or even more for large networks) to explore resolution parameters and stochastic variation due to pseudorandom node order in the heuristic. We note a large increase in runtime for the full Stanford network, with over 2 million edges. As also observed in the figure, fitting a stochastic block model, at $O(N \ln^2 N)$ for sparse networks in this implementation,(96) becomes significantly slower on the full networks with more than 200,000 edges.

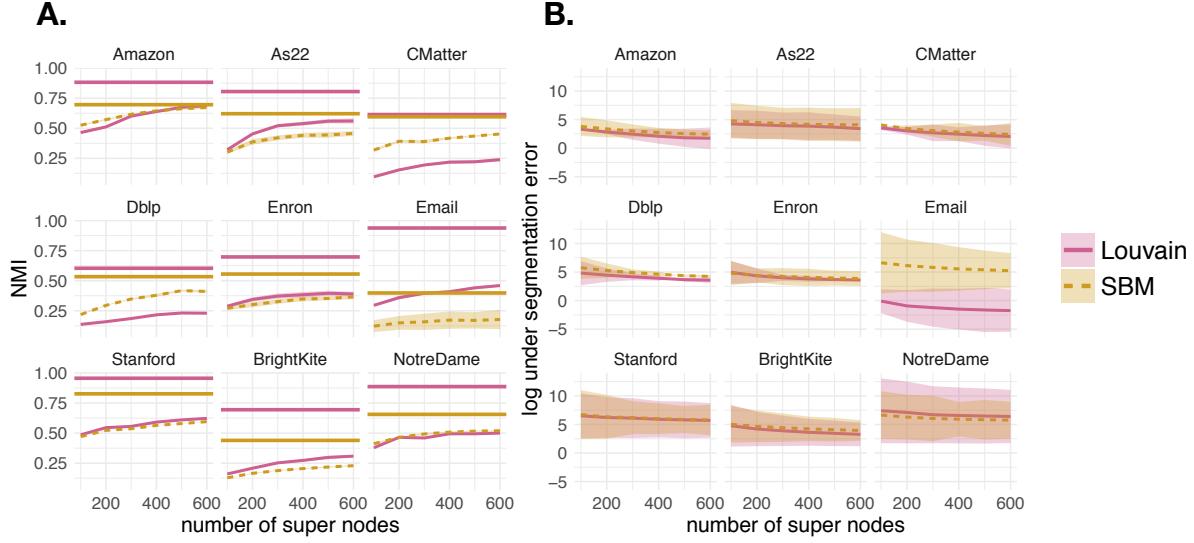


Figure 3.5: Super Node Quality. We computed normalized mutual information (**A.**) and under segmentation error (**B.**) for networks represented by between 100 and 600 super nodes. Line type and color indicate the community detection algorithm applied (Louvain algorithm or SBM fitting). Each curve indicates the mean across 5 super node representations. The shaded area shows standard deviation. **A.** Normalized mutual information between the full and super node representations of networks [i.e. $\text{NMI}(\mathbf{z}^{\text{Full}}, \mathbf{z}^{SN})$]. A network representation with more super nodes. generally increases the NMI between full network and super node network representations. Horizontal lines indicate the mean pairwise NMI between 10 runs of the Louvain algorithm and SBM result on the full network (pink and gold, respectively). Given the high variability between multiple runs of the same algorithm on the full network, adding more super nodes can only improve the NMI between the full and super node representation to the observed level of similarity observed between algorithm runs. **B.** The log under segmentation error for super node representations. Defining a super node representation with more super nodes generally decreases the under segmentation error.

While we see a significant improvement in community detection runtime from using super nodes for both methods, the benefit in the SBM fitting is particularly large, especially for the bigger networks (DBLP, Stanford, Email). In moving to the super node representation, we traded out large-coefficient scaling-with- N community detection computations for those scaling with $S \ll N$ (with possible increases due to the increased density of the super node representation), at the cost of constructing the super node representation. In particular, we observe the SBM runtimes on super nodes appear to be relatively independent of M . We note that each of the three steps building our super-node representation is $O(M)$, so in the large graph limit the expected gain of our approach may be only a constant factor over Louvain iterations and the SBM fitting (up to logarithmic factors). In the present calculations, we have not endeavored to optimize the runtime to build our super node representations; even so, the three steps building the $S = 500$ super node representation of the

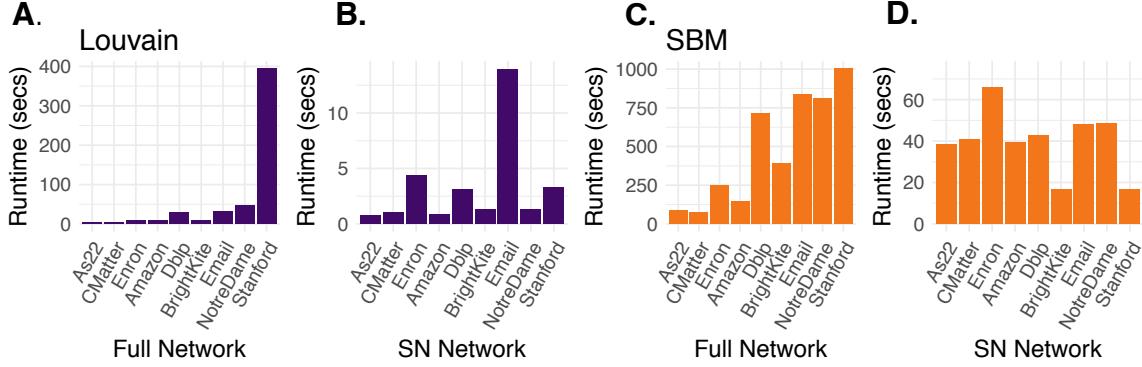


Figure 3.6: Runtimes. We compare community detection runtimes (in seconds) with the Louvain algorithm and by fitting an SBM on the full networks and super node representations for the 9 data sets. **(A.)** Louvain on the full network. **(B.)** Louvain on the super nodes. **(C.)** SBM on the full network. **(D.)** SBM on the super nodes.

Stanford network in our current implementation together take ~ 350 sec with CoreHD and ~ 200 sec using highest-degree nodes. While this alone might not seem like a large improvement compared to a single realization of running Louvain or fitting an SBM, the computational gains compared to generating multiple community partitions can be quite significant.

3.5.4 Quantifying variability across algorithm runs

As mentioned previously, there is variability in the partitions generated by multiple runs of the same community detection algorithm on the same network representation. We sought to quantify how the variability or pairwise similarity between multiple runs of the same algorithm changes as a function of the number of super nodes in each network representation under the default and matched algorithm parameters (Figures 3.7 A. and B.), respectively. Note that A. and B. show the same analysis with the only difference being the parameters input to the community detection algorithm. Curves represent the mean pairwise NMI between all pairs of 10 computed partitions under the super node network representation and shading shows standard deviation. The pink and blue curves show the within-method comparisons on the super node network representation for the Louvain algorithm and stochastic block model, respectively. That is, these are the comparisons shown in Figure 3.4 B. We were also interested in the variability of the partitions obtained between partitions of the super node representation found with different algorithms. This result is shown in the gold curve and labeled ‘Louvain+SBM’. Note that this analysis is described in Figure 3.4 C. The horizontal

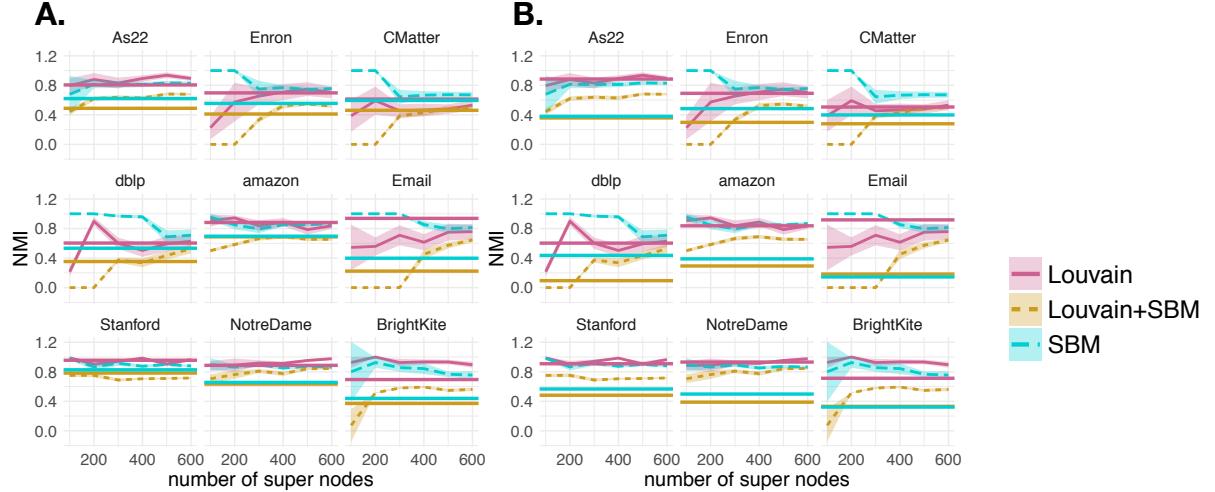


Figure 3.7: **Quantifying partition variability.** For each of the 9 networks, we obtained 10 different partitions by the Louvain algorithm and 10 different SBM fits under the default (**A.**) and matched settings (**B.**). To assess the similarity between partitions within and between a community detection algorithm in networks under the the super node representation, we computed pairwise normalized mutual information (NMI) as a function of the number of super nodes. The pink and blue curves show the mean pairwise normalized mutual information between all pairs of 10 partitions under Louvain and SBM fitting, respectively. The gold curves compare pairs of partitions under different methods. Shaded area denotes standard deviation. Horizontal lines indicates the mean pairwise NMI between partitions under the full network representation for within Louvain and SBM partition comparison (pink and blue, respectively) and between Louvain and SBM partition comparison (gold). Overall, the super node representation is useful for reducing the disparity between the partitions obtained under different methods.

lines show the mean pairwise similarity observed between all 10 runs of the Louvain algorithm and stochastic block model (pink and blue, respectively). The most significant improvement we observe under the super node representation is between runs of the Louvain algorithm and SBM fitting, suggesting that the new compressed representation of the network has prominent structural features that are robustly identified with both approaches. A high normalized mutual information between a pair of partitions indicates that the algorithms identified similar community structures. The Louvain algorithm is generally less variable than fitting an SBM, but we also observed decreased variability in the fitting of stochastic block models on the super node representation.

3.5.5 Neighborhood agreement

While we have emphasized benefits in the mechanics and usability of running standard community detection algorithms, we now seek to address whether the communities that we find using the super

node representation align with local network connectivity so that neighbors are more likely to have similar community assignments and how this alignment compares with what we would have found by community detection on the full network. While we visualize this qualitatively for the As22 network in Figure 3.8E-H, we also designed a prediction task to quantify this alignment. In this prediction task, we seek to take a node-to-community partition (from either the full or super node network representations \mathbf{z}^{Full} or \mathbf{z}^{SN} , respectively) and the full network \mathcal{X} to see how accurately we can predict members of a community for different neighborhood sizes. For network \mathcal{X} with node-to-community assignments \mathbf{z} , we assign a probability distribution to each node over all of the communities under \mathbf{z} . For a neighborhood order o (x-axis in Figure 3.8), we say that node i has probability of being in community k , based on what fraction of its neighbors belong to that community under \mathbf{z} . Then for each community in \mathbf{z} , we perform a binary prediction task for whether each node of \mathcal{X} should be assigned to that community, according to the computed probability distributions for all nodes with respect to that community.

We sweep the probability parameter, p , representing the required threshold probability for a node to achieve in order to be assigned to a community in this binary classification task. By sweeping p for each of the communities, we compute an ROC curve for each community and the corresponding areas under the curve (AUC). Finally, we use the minimum AUC value as our summary statistic of this task, with a high AUC value indicating that the neighboring regions of a node were strong predictors of community assignments, as shown in Figure 3.8A-D. All experiments are performed on 5 networks (As22, Enron, CMatter, Dblp, Amazon) and for both the matched and default parameters (indicated by line type) in the full network. (Recall from section 3.1 that the matched parameters for the full network were chosen based on the super node partition results under default settings; hence, there is no corresponding ‘matched’ set for the super nodes in these plots.) We observe in most cases using the super node representation improves the minimum AUC value, indicating that communities obtained from this representation have higher agreement with local connectivity by this measure. To qualitatively evaluate how the super node representation is able to ultimately partition the network into communities that are locally relevant, we visualize the As22 network, with nodes colored by communities identified under the full network (E,G) and super node representation (F,H) under Louvain (E,F) and SBM (G,H). Consistent with the quantitative results in Figure 3.8A-D, we

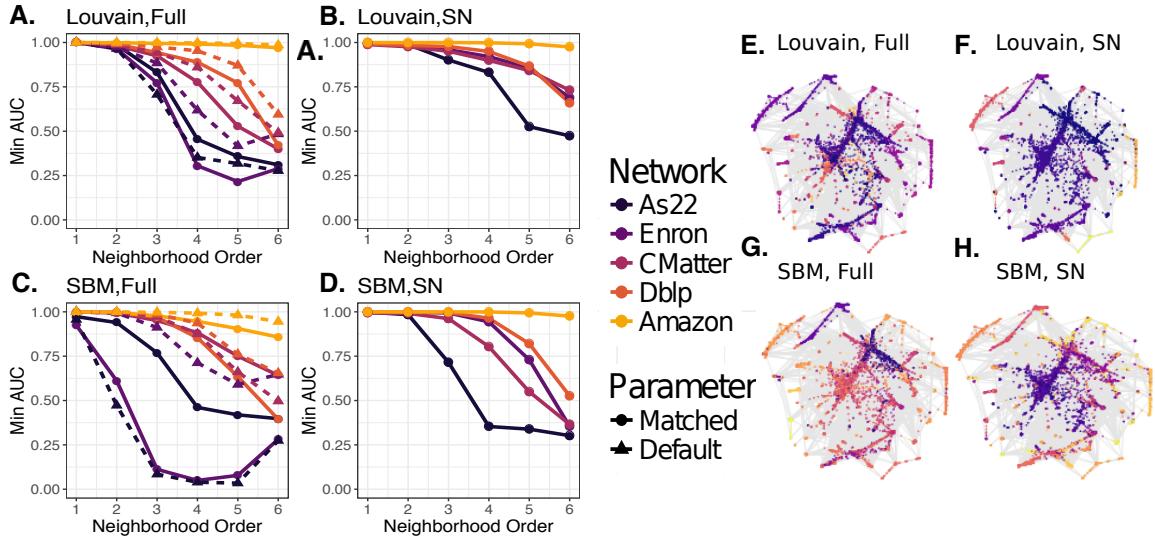


Figure 3.8: Agreement of community assignments with local connectivity. We study how consistent partitions are within local neighborhood regions of the network by examining how well a node’s neighbors (for various order neighborhoods) can be used to predict its community assignment, under some community partition z . For each community in a partition, we give a binary prediction of whether a node is assigned to that community, based on probabilities we compute for a node from its neighbors. Sweeping the parameter p that sets the probability required for a node to be assigned to a community, we compute ROC curves for each community and report the minimum AUC value observed. Panels **A-D** show minimum AUC values observed as a function of neighborhood order for communities obtained from the full networks and super node representations by Louvain and by SBM. Line color indicates network and line type indicates communities obtained from the matched and default parameters used by the algorithms on the full networks. Panels **E-H** visualize the communities obtained in the As22 data on the full network (default parameters) and super node representation (SN) under Louvain and SBM, with node colors indicating community memberships.

observe that the super node representation leads to an effective coarse-graining, with the community labels appearing to be qualitatively consistent across large regions of the network.

3.6 Conclusion and Future Work

We developed an approach for compressing a network into a super node representation that can be used in standard community detection algorithms. Using the smaller super node network reduces runtime and the variability between multiple runs of the same community detection algorithm. Our results also demonstrate that the communities in the super node network are better aligned with local network neighborhoods in a predictive sense, while still being in relatively good alignment with the partitions obtained using the full network.

Super nodes may be useful in a variety of contexts where large datasets are otherwise difficult to mine and interpret. For example, one might visualize the super node version of the network rather than the entire network, or use the members of a super node to identify redundant information in the network. Future work on super node representations could include the extension of this method to directed, signed, attributed, or bipartite networks. Additionally, one might consider a probabilistic model framework, attempting to infer latent super node assignments. Future work could also examine graph theoretic properties of super node representations in terms of how it aligns with the original network.

CHAPTER 4

An attributed stochastic block model

While community detection is an inherently unsupervised problem, there are often circumstances where there is extra information about the nodes, or node attributes. In this case, the clustering problem of grouping nodes into communities becomes a problem of integrating two different types of data. There have recently been numerous discussions as to what extent it makes sense to assume that members of a community according to connectivity should also have similar attributes. In particular, we address the case where each node has multiple continuous attributes. To build a model that takes into account both sources of information, we extend the classic stochastic block model to take into account continuous attribute information, when inferring node-to-community assignments. In the process of assigning nodes to communities, we learn probabilistic models describe both the attributes and connectivity. We then use these learned models to perform link prediction and collaborative filtering tasks.

4.1 Motivation for studying attributed networks

In certain applications, each node in a network is equipped with additional information (or particular attributes) that was not implicitly taken into account in the construction of the network. For example, one could consider a collection of attributes measured for individuals in a social network (i.e., age, interest, level of education). The interplay between the connectivity-based (or structural) community organization of the network and attribute information has gained attention, as it is often unclear whether it is valid to assume that a *structural* community should necessarily correlate with an attribute-based *functional* community (55; 94; 129). While these studies suggest that extreme caution should be taken in assuming a correlation between structural and functional communities, we limit our focus in this work to the assumption that a node's connectivity and attribute patterns can be

jointly modeled based on its community assignment. In other words, we seek to develop an approach to assign node to communities based jointly on both sources of information, such that a community is defined as a group of nodes with similar connectivity and attribute patterns. Moreover, our objectives are two-fold. First, we seek to develop a probabilistic approach to jointly model connectivity and attributes. Second, we wish to ensure that our model can handle multiple, continuous attributes.

Recently, there have been numerous efforts to incorporate attribute information into the community detection problem (130; 87; 28; 55; 94). Two classes of community detection methods are quality function-based and probabilistic. Quality function based methods define a quantity of interest that an ideal partition would satisfy, while probabilistic methods identify communities through likelihood optimization and focus on the underlying statistical distribution for the observed network. A recent quality function-based method to handle multiple attributes is I-louvain (28). This method approaches the problem as an extension to the Louvain algorithm, which is the state-of-the-art scalable quality function modularity-based community detection method (20). The traditional modularity-based approach to community detection defines a null model for community structure under the assumption that there is not substantial structural organization in the network and seeks to identify a partition maximally different from this model through optimizing a quality function (modularity). The I-louvain method modifies the standard modularity quality function as to what they call 'inertia-based modularity'. This quality function incorporates the Euclidean distance between nodes based on their attributes. This work demonstrates in multiple examples how incorporating connectivity and attributes allows for a partition of nodes to communities that aligns with ground truth better than the results obtained based on connectivity or attributes in isolation.

Alternatively, there a variety of probabilistic approaches to handling attributed network data (87; 55; 94; 130). Similar to our work in the sense that community membership is related to node attributes is CESNA (130). The objective in this approach is to learn a set of propensities or affiliations for each node across all possible communities, such that two nodes with similar propensities towards communities should have more in common in terms of connectivity and attributes. In this model, each node has a vector with multiple binary attributes. The affiliation model is useful and flexible because it does not enforce a hard partitioning of nodes into communities, which is useful in social network applications. In this inference problem, the connectivity and attribute information are used

to infer a node’s affiliations to communities and then models the probability of an edge between two nodes as a function of the similarity in their community affiliation propensities.

In contrast to the affiliation model, the stochastic block model (111) (at least the more standard variants of it), seeks to determine a hard partition for each nodes across communities and models edges between a pair of nodes according to their community assignments. The partition of nodes to communities through a stochastic block model framework is accomplished through maximum likelihood optimization. An of the stochastic block model was first explored by Clauset *et al.*, (87), which adapts the classic stochastic block model to handle a single attribute with the assumption that attributes (referred to as metadata in their work) and communities are correlated. Next Hric *et al.*, developed an attributed SBM (55) from a multilayer network perspective, with one layer modeling relational information between attributes and the other modeling connectivity. The authors then seek to assign nodes to communities maximizing the likelihood of the observed data in each layer. Finally, work of Peel *et al.* made important contributions in 1) establishing a statistical test to determine whether attributes actually correlates with community structure and 2) developing an SBM with flexibility in how strongly to couple attributes and community membership in the stochastic block model inference problem (94).

The model that we seek to develop in this work is distinguished by its ability to fit a stochastic block model to networks, where each node has multiple continuous attributes. This model is most appropriate for circumstances where there is domain-specific evidence that members of a community should exhibit similarities in attributes. We highlight two examples of this, where we apply our model to a protein interaction network and a microbiome subject similarity network. Our paper is organized as follows. First, we will define our attributed SBM and an inference technique for fitting the model. Next, we test this approach on a synthetic example as well as two biological network examples. Since community detection methods are often difficult to validate due to the lack of ground truth information on the nodes, we use link prediction and collaborative filtering to quantify how well the attributed SBM represents the data.

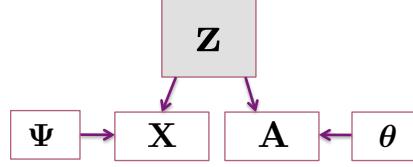


Figure 4.1: **Modeling community membership in terms of attributes and connectivity.** Node-to-community assignments specified by \mathbf{Z} are determined in terms of adjacency matrix information, \mathbf{A} and attribute matrix information, \mathbf{X} . \mathbf{A} and \mathbf{X} are assumed to be generated from a stochastic block model and a mixture of multivariate Gaussian distributions, parameterized by θ and Ψ , respectively.

4.2 An Attributed Stochastic Block Model

In this section we provide the details for our version of the attributed stochastic block model and the inference procedure used to learn the model parameters.

4.2.1 Objective

We seek to incorporate both connectivity (\mathbf{A}) and attribute information (\mathbf{X}) to infer node-to-community assignments, \mathbf{Z} . Note that for a network with N nodes, K communities and p measured attributes, \mathbf{A} , \mathbf{X} , and \mathbf{Z} have dimensions $N \times N$, $N \times p$ and $N \times K$, respectively. In particular, \mathbf{Z} is a binary indicator matrix, where entry z_{ic} is 1 if and only if node i belongs to community c . We also define \mathbf{z} to be the N -dimensional vector of node-to-community assignments. We assume that connectivity and attributes are conditionally independent, given the community membership label. The graphical model for the relationship between node-to-community labels, connectivity and attribute information is shown in 4.1.

To infer the \mathbf{Z} that best explains the data, we adopt a likelihood maximization approach. That is, we seek to find the partition of nodes to communities that best describes the observed connectivity and attribute information. Given the conditional independence assumption of \mathbf{X} and \mathbf{A} , we can express the log likelihood of the data, \mathcal{L} as the sum of connectivity and attribute log likelihoods, \mathcal{L}_A and \mathcal{L}_X , respectively, as

$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_X . \quad (4.1)$$

This likelihood reflects the joint distribution of the adjacency matrix, \mathbf{A} , the attribute matrix, \mathbf{X} , and the matrix of node-to-community indicators, \mathbf{Z} ; formally, we have

$$\mathcal{L} = p(\mathbf{A}, \mathbf{X}, \mathbf{Z}) . \quad (4.2)$$

Given that \mathbf{Z} is a latent variable that we are trying to infer, we can approach the problem using the expectation maximization (EM) algorithm (38). By doing this, we will alternate between estimating the posterior probability that a node i has community label c , or

$$p(z_{ic} = 1 | \mathbf{X}, \mathbf{A}) \quad (4.3)$$

and estimates for θ, Ψ , i.e., the model parameters specifying the adjacency and attribute matrices, respectively.

4.2.2 Attribute Likelihood

For a network with K communities, we assume that each particular community i has an associated p -dimensional mean μ_i and $p \times p$ covariance matrix, Σ_i . Note that these parameters uniquely identify a p -dimensional multivariate Gaussian distribution. To specify this model for all K communities, we define the parameter $\Psi = \{\mu_1, \mu_2, \dots, \mu_k, \Sigma_1, \Sigma_2, \dots, \Sigma_K\}$.

The log likelihood for the mixture of Gaussians on the attributes is written as,

$$P(\mathbf{X} | \Psi) = \sum_{i=1}^N \log \left\{ \sum_{c=1}^K \pi_c \mathcal{N}(\mathbf{x}_i | \mu_c, \Sigma_c) \right\} \quad (4.4)$$

Here, $\mathcal{N}(\mathbf{x}_i | \mu_c, \Sigma_c)$ is the probability density function for the multivariate Gaussian and π_c is the probability that a node is assigned to community c .

4.2.3 Adjacency Matrix Likelihood

For the adjacency matrix, \mathbf{A} and the $K \times K$ matrix of stochastic block model parameters, θ , the complete data log likelihood can be expressed as,

$$\begin{aligned} \log(P(\mathbf{A} | \mathbf{z})) &= \frac{1}{2} \sum_{i \neq j} \sum_{k < l} z_{ik} z_{il} [a_{ij} \log(\theta_{kl}) \\ &\quad + (1 - a_{ij}) \log(1 - \theta_{kl})] \end{aligned} \quad (4.5)$$

4.2.4 Inference

To use EM to maximize the likelihood of the data, we break the process into the E-step and M-Step, and perform this step sequence iteratively until the estimates converge.

E-Step. During the E-step, we use the current value of learned model parameters, θ and Ψ to compute the posterior, given in Eq. (4.3) at each step. The posterior at each step, $\gamma(z_{ic})$, of node i belonging to community c , is given by

$$\begin{aligned}\gamma(z_{ic}) &= p(z_{ic} = 1 \mid \mathbf{x}_i, \mathbf{a}_i) \\ &= \frac{p(\mathbf{x}_i \mid z_{ic} = 1)p(\mathbf{a}_i \mid z_{ic} = 1)\pi_c}{\sum_{c=1}^K p(\mathbf{x}_i \mid z_{ic} = 1)p(\mathbf{a}_i \mid z_{ic} = 1)\pi_c}.\end{aligned}\quad (4.6)$$

Here, \mathbf{x}_i and \mathbf{a}_i denote the attribute and connectivity patterns for node i , respectively.

M-Step. In the M-step, we can compute updates for θ and Ψ using this expectation.

Since, the attributes follow a Gaussian mixture model, it can be shown that the update for the mean vector describing community c , μ_c , can be computed as

$$\mu_c = \frac{\sum_{i=1}^N \gamma(z_{ic})\mathbf{x}_i}{\sum_{i=1}^N \gamma(z_{ic})}.\quad (4.7)$$

Similarly, the update for the covariance matrix describing a community, Σ_c , is computed as

$$\Sigma_c = \frac{\sum_{i=1}^N \gamma(z_{ic})(\mathbf{x}_i - \mu_c)(\mathbf{x}_i - \mu_c)^T}{\sum_{i=1}^N \gamma(z_{ic})}.\quad (4.8)$$

To update the parameters of θ , we follow the method in (32) and update the probability of an edge existing between community q and l , given by θ_{ql} as,

$$\theta_{ql} = \frac{\sum_{i \neq j} \gamma(z_{iq})\gamma(z_{jl})x_{ij}}{\sum_{i \neq j} \gamma(z_{iq})\gamma(z_{jl})}\quad (4.9)$$

We continue the process of iterating between the E-step and M-step until the change in the data log-likelihood, \mathcal{L} , is below a predefined tolerance threshold.

4.2.5 Initialization

Likelihood optimization approaches are often sensitive to initialization because it is easy to get stuck in a local optimum. As an initialization strategy for the nodes, we simply cluster the nodes in the network using the Louvain algorithm (20). We chose this approach because this algorithm is efficient and stable.

4.3 Synthetic Data Results

We first test the performance of our model and inference procedure on a synthetic example. Here, we generated a network for a stochastic block model with $N = 200$ nodes and $K = 4$ communities. Edges in the adjacency matrix were generated according to a stochastic block model, parameterized as follows:

$$p(A_{ij} = 1) \sim \begin{cases} \text{Bernoulli}(.10), & \text{if } z_i \neq z_j \\ \text{Bernoulli}(.25), & \text{if } z_i = z_j \end{cases} \quad (4.10)$$

Note that \mathbf{z} is a 200-dimensional vector, where z_i identifies the community label for node i .

Fig. 4.2A. shows the adjacency matrix for an example network generated according to this parametrization. The black marks in the image indicate an edge. While this network has assertive structure with members of a community having more edges on average with each other than with other communities, there are still many noisy edges going between communities, making the correct community structure more difficult to discern.

To model attributes, for a community c , we randomly generated an 8-dimensional vector, $\boldsymbol{\mu}_c$, where each entry is from a Gaussian with 0-mean and unit variance. Associated with all c , $c \in \{1, 2, 3, 4\}$ is a 8×8 diagonal covariance, $\boldsymbol{\Sigma}_c = \text{diag}(1.25)$. Moreover, using the $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$, a sample attribute vector can be generated. That is, the attribute vector for node i , \mathbf{x}_i is generated as,

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}) \quad (4.11)$$

where $\mathcal{N}(\cdot, \cdot)$ denotes a multivariate Gaussian.

Fig. 4.2B. shows a PCA plot of the attribute vectors associated with each node in an example synthetic experiment and hence, each point represents a node. Since, the true dimension of these feature vectors is 8, this plot provides a projection, based on the top 2 principal components, that allows for visualization of the relatedness between the nodes, according to the attributes. One can observe that members of community 2 are overall nicely separated in attribute space but members of communities 3 and 4 are especially hard to discern.

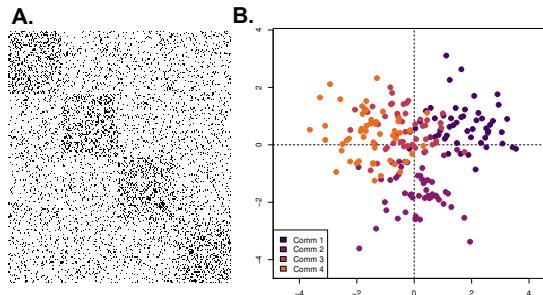


Figure 4.2: Synthetic Example. We generated a synthetic network with $N = 200$ nodes, $K = 4$ communities and an 8-dimensional multivariate Gaussian for each community. **A.** A visualization of the adjacency matrix for this network where a black dot indicates an edge. We observe that there is an assortative block structure (blocks on the diagonal), but there are also many 'noisy' edges between communities making the true community structure with only a stochastic block model a bit harder to detect. **B.** We performed PCA on the $N \times p$ dimensional attribute array and plotted each of the N nodes in two dimensions. Points are colored by their true community assignments, \mathbf{z} . Clustering the nodes according to only connectivity, only attributes, and with the attributed SBM, we quantified the partition accuracy with normalized mutual information. This gave results $\text{NMI}(\mathbf{z}, \{\mathbf{z}^{\text{connectivity}}, \mathbf{z}^{\text{attributes}}, \mathbf{z}^{\text{attribute sbm}}\}) = \{0.65, 0.68, 0.83\}$.

To assess how well the attribute SBM approach performed in successfully assigning nodes to communities, we compared the results obtained from our model to clustering results obtained either only clustering the nodes based on connectivity, and to results of clustering nodes based only on their attribute information. We quantify the correctness of the partition with normalized mutual information (NMI) (30). Letting \mathbf{z} denote the true node-to-community assignments, then $\mathbf{z}^{\text{connectivity}}$, $\mathbf{z}^{\text{attributes}}$, and $\mathbf{z}^{\text{attribute sbm}}$ denote the partition of the nodes according to only the network connectivity only, attributes only, and with the attributed SBM. To cluster the network only according to connectivity, we fit a stochastic block model with 4 blocks. To cluster nodes with only attributes, we performed k -means clustering on only the attributes. Computing the NMI between \mathbf{z} and each

of these 3 cases, or $\text{NMI}(\mathbf{z}, \{\mathbf{z}^{\text{connectivity}}, \mathbf{z}^{\text{attributes}}, \mathbf{z}^{\text{attribute sbm}}\}) = \{0.65, 0.68, 0.83\}$. These results show that by combining both sources of information, there is an improvement in the ability to correctly identify communities. To further probe this idea, we sought to empirically look closer at the so-called 'detectability limit'. Generally, detectability refers to the difficulty of correctly identifying clusters in data. Multiple works (36; 116) have explored these limits in stochastic block models observing the sharp phase transition that occurs in accuracy as soon as within-community probability (p_{in}) is sufficiently larger than the between-community probability (p_{out}).

Based on the results of the synthetic experiments in Figure 4.2 where the attributes combined with connectivity lead to a more accurate partitioning of the nodes, we hypothesized that augmenting the network connectivity with attributes could somehow move this detectability limit. In Figure 4.3, we explored how generating networks from a stochastic block model with varying ratios between p_{in} and p_{out} combined with the attributes used in Figure 2 would affect the accuracy of the node-to-community partition. To do this, we considered values of p_{in} between 0.05 and 0.3 in increments of 0.05. For each of these p_{in} values, we found the corresponding value of p_{out} such that the mean degree was 20. Fixing the mean degree allows for direct comparison of how the within-to-between community probabilities influence the detection of correct communities. For each of these p_{in} and p_{out} combinations, we generated 10 different networks using a stochastic block model. In figure 3 we plot the NMI between the true partition, \mathbf{z} and the partitions using only the connectivity with the regular SBM $\mathbf{z}^{\text{connectivity}}$ and the attributed SBM $\mathbf{z}^{\text{attribute sbm}}$. These results are plotted in blue and pink, respectively. The shaded region around the points indicates standard deviation.

4.3.1 Detectability analysis

We see that while both inference approaches undergo a phase transition at a similar ratio of $p_{in}/p_{out} = 3$, we notice that the curve for the attribute SBM model is slightly shifted to the left suggesting that the extra attribute information does positively impact the ability to correctly identify communities. Future work could focus on understanding detectability questions in relation to the parameters for the underlying multivariate Gaussian distributions parametrizing each community. For example, how does the difference in means between a pair of communities shift the detectability curve?

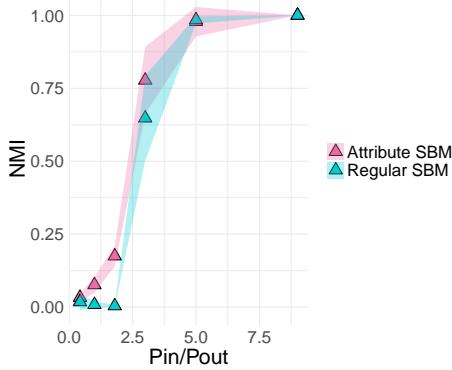


Figure 4.3: Detectability Analysis in Synthetic Example. To understand how attribute information can be combined with connectivity to assign nodes to communities accurately, we generated synthetic networks for within-probabilities of p_{in} between 0.05 and 0.3 with corresponding p_{out} or between-community probabilities such that the mean degree of the network was 20. For each of these synthetic networks, we used the attributes from the analysis in figure 2 to fit the attributed SBM. Here, we plot the correctness of the node-to-community assignment with normalized mutual information using the partition obtained from regular SBM (blue) and the partition under the attributed SBM model fit (pink). For each combination of p_{in} and p_{out} , we generated 10 networks and hence the bands around the points denote standard deviation. Incorporating attributes with the attributes stochastic block model shifts the detectability limit slightly to the left.

4.4 Using the fitted attributed SBM for link prediction and collaborative filtering

One of the benefits of a generative network model is that it can be applied to prediction tasks. Most notably, in the absence of one source of information about a node (connectivity or attributes), the model can be used to predict the complementary information source (attributes or connectivity, respectively). By fitting an attributed SBM, obtained successful performance in two fundamental network prediction tasks, link prediction and collaborative filtering.

4.4.1 Link Prediction Methods

In the link prediction problem, when given two node stubs, the objective is to determine whether a link exists between them. Since we are modeling connectivity with a stochastic block model, we can predict links using the learned parameters. In particular, we highlight how this task can be performed using just the attribute information of the node stubs of interest. In the experiments to follow, we compare to 3 commonly-used link prediction methods. In all of these methods, a score is computed

for all pairs of edge candidates and ultimately the top x set of edges with highest weights are kept (where x is some user-defined parameter). Let m and n be a pair of nodes and $\Gamma(m)$ denote the set of neighbors for a node m . Then, under the following 3 common link prediction methods (125), we can calculate the score of the potential link as, $\text{Score}(m, n)$.

$$\textbf{Jaccard: } \text{Score}(m, n) = \frac{\Gamma(m) \cap \Gamma(n)}{\Gamma(m) \cup \Gamma(n)}$$

$$\textbf{Adamic Adar: } \text{Score}(m, n) = \sum_{c \in \Gamma(m) \cap \Gamma(n)} \frac{1}{\log |\Gamma(c)|}$$

$$\textbf{Preferential Attachment: } \text{Score}(m, n) = |\Gamma(m)| \times |\Gamma(n)|$$

For the link prediction tasks shown in figures 5 and 9, we performed a link prediction task by sampling pairs of nodes and utilizing the complementary source of attribute information. We sampled 10 different sets of 50 pairs of nodes. In each sample, 25 of the node pairs were those having an edge in the original network and 25 were pairs with no edge. For each of the 50 edges in each sample, we sought to predict whether an edge existed between the corresponding node pair in a leave one out manner. To do this, for each edge we fit the attributed SBM to the network with the pair of nodes (stubs) associated with the edge removed. We then use the nearest neighbor in attribute space of each stub as the input to each of the 3 baseline community detection methods (Jaccard, Adamic Adar, and Preferential Attatchment). To use our attributed SBM in this link prediction task, we also consider the most commonly observed community among the 3 nearest neighbors for the stubs of the edge of interest. Again, using the nearest neighbors, which we denote by n and m of the stubs, then we define the link prediction score for the edge as θ_{z_n, z_m} , or the probability that an edge exists between nodes n and m according to the fitted model. After generating 10 samples of 50 edge pairs, this results in 500 total edge scores. Since we know the ground truth of whether or not these edges actually exist from the original network, we can construct an ROC curve for each method. From these curves we can plot area under the curve (AUC) to quantify the quality of the link prediction result. Using the attributed SBM is a way to incorporate community information into the link prediction problem which has previously shown to be effective (113).

4.4.2 Collaborative Filtering Methods

Conversely, the collaborative filtering problem seeks to predict a node's attributes based on its similarity to its neighbors. For some node of interest, we can use our fitted attributed SBM model

to predict a node's attributes, given only the information about its connectivity. Formally, for node i , we seek to predict \mathbf{x}_i . In the following experiments, we compare our results to two common collaborative filtering approaches (110). Let $\mathcal{N}^k(m)$ be the set of k -nearest neighbors in the network for node m . Let $\hat{\mathbf{x}}_i$ be the predicted attribute vector for node i and s_{ij} be a similarity measure between nodes i and j .

$$\textbf{Neighborhood Avg: } \hat{\mathbf{x}}_i = \frac{1}{|\mathcal{N}^k(i)|} \sum_{j \in \mathcal{N}^k(i)} \mathbf{x}_j$$

$$\textbf{Weighted Neighborhood Avg: } \hat{\mathbf{x}}_i = \frac{1}{\sum_{j \in \mathcal{N}^k(i)} s_{ij}} \sum_{j \in \mathcal{N}^k(i)} s_{ij} \mathbf{x}_j$$

In collaborative filtering experiments, the objective is to predict the vector of attributes for each node. In our experiments, we used leave-one-out validation to predict the attribute vector for each node. That is, for each node in the network, we created a single node test set. The training set, was then the rest of the network with the node to predict removed. For this single test set node, we identified neighbors it connects to in only connectivity space within the training set. For standard collaborative filtering approaches (Neighborhood average and weighted neighborhood average), the predicted attribute for the test set node is then the specified averaging of the neighbors. To use our model for this task, we first fit the attributed SBM model to the training set. Similar to the standard link prediction approaches, we identify the nearest neighbors for our test node in connectivity space within the training set. We then predict the community membership of our test node to be the most-frequently observed community among its neighbors. Using this community assignment, c , we then predict the attribute vector for our test node to be μ_c , or the mean vector that was learned to describe community c . The results of collaborative filtering experiments for the microbiome and protein network examples are shown in figures 6 and 10. For a node i and its associated vector of attributes, \mathbf{x}_i we quantify the accuracy of the predicted attribute vector, $\hat{\mathbf{x}}_i$ with a relative error measure, \mathcal{E} , such that

$$\mathcal{E} = \frac{\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2}{\|\mathbf{x}_i\|_2}. \quad (4.12)$$

Similar to the success of integrating community information for link prediction, an collaborative filtering tasks have previously shown success from the integration of network community structure (39).

4.5 Applications in Biological Networks

We evaluate the potential to combine similarity or relational information between a set of entities for application in biological data. Whether that be networks of proteins, genes, or bacterial species with extra experimental data. Our application of this model to biological problems provides a framework to predict attribute or connectivity information about a new observation. Note that we do not intend to suggest any new biological insights, but rather that we can combine two sources of information for prediction tasks and alternative definitions of what constitutes a community in the data. Applying the attributed stochastic block model to integrate connectivity and attribute data provides a way to find a partition that takes into account two different sources of information, or a method to predict one source of information (connectivity, attributes) in the absence of the other (attributes, connectivity).

4.5.1 Microbiome Subject Similarity Results

4.5.1.1 Motivation

In the analysis of biological data, it is often useful to cluster subjects based on a set of their measured biological features and to then determine what makes each of the subgroups different. One type of biological data gaining much attention in recent years is metagenomic sequencing data, used to profile the composition of a microbiome. We refer to this as the 'metagenomic profile' and each feature is a count for each bacterial species, also known as operational taxonomic unit (OTU). Lahti et al. conducted a study among subjects across a variety of ethnicities, body mass (BMI) classifications, and age groups to understand differences in the intestinal microbiota (66). Using metagenomic sequencing, the counts for 130 OTUs were provided for each subject. We created an experiment to test our model by seeing if we could overlay a similarity network between subjects with the individual OTU count vectors for each subject.

4.5.1.2 Pre-Processing

The data were downloaded from <http://datadryad.org/resource/doi:10.5061/dryad.pk75d>. We extracted a subset of the subjects from Eastern Europe, Southern Europe, Scandinavia, and the United States. Using only these subjects, a between-subject similarity network was con-

structed between the 121 individuals who had a BMI measurement. This resulted in a network of 121 nodes, where each edge is the pearson correlation between their microbial compositions. We then removed all edges in the network with weight (correlation) < 0.7 . Note that our attributed SBM does not allow for edge weights, so as input of this network into the model, we simply ignored the edge weights.

4.5.1.3 Constructing Node Attributes

Since each node had a 130-dimensional vector of attributes (counts), we used this information to create a lower-dimensional attribute vector for each node by performing PCA and then representing each node with the first 5 principal components. Each dimension of this new attribute vector was then centered and scaled and has an approximately gaussian distribution.

We first visualized the differences in partitions obtained according to the classic and attributed stochastic block models in figure 4A-B, respectively. In both networks, nodes are colored by their community assignment. Using the classic stochastic block model and the model selection criterion described in (32), 7 blocks were identified. With the attributed stochastic block model, 6 blocks were identified. While we do not have ground truth labels on the nodes, it is visually apparent that adding the attributes to the inference problem helps to 'clean up' the partition. For example, in figure 4A there is mixing between the dark and lighter purple communities in the upper left of the network. In figure 4B, this mixing was reduced by assigning all of the nodes in the general region to the lighter purple community.

4.5.1.4 Link Prediction Experiments

We performed link prediction on the microbiome subject similarity network as described in section 4.1. The associated ROC curves are plotted in figure 5. All four methods have satisfactory performance with the attributed stochastic block model giving the best results. The AUC values for the attributed sbm, Jaccard, Adamic Adar, and preferential attachment are 0.71, 0.69, 0.69, and 0.62, respectively.

4.5.1.5 Collaborative Filtering Experiments

We performed the collaborative filtering experiments on the microbiome subject similarity network in the manner described in section 4.2 to predict the 5-dimensional attribute vector for each node.

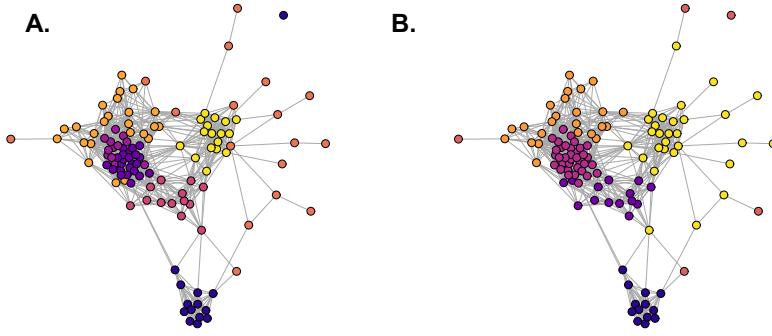


Figure 4.4: Microbiome subject similarity network: A visualization of the 121 node microbiome subject similarity network with nodes colored by the partition using the classic (**A.**) and attributed (**B.**) stochastic block model. **A.** Fitting the classic stochastic block model to the network, 7 communities were identified. **B.** Fitting the attributed stochastic block model to the network with the attributes being the first 5 principle components of each subject's OTU count vector (metagenomic profile), 6 communities were identified. Incorporating attributes in inferring this partition removed some of the noise in the partition on the network, specifically in the mixed purple community in the left of **A.**

The box plots indicate the distribution of relative errors over the 121 nodes for the attribute SBM (blue), neighbor average (pink) and weighted neighbor average (orange). While the attributed SBM plotted has a similar distribution of relative errors with the standard collaborative filtering methods, the mean is slightly lower, at 0.21, compared to 0.26 and 0.27 in the neighbor average and weighted neighbor average, respectively.

4.5.2 Protein Interaction Network Results

We also apply our attributed SBM approach to the protein interaction network presented in (22). This network represents interactions between proteins, predicted from the literature. Associated with each node (protein), is a classification of one of 6 experimental modifications observed from the exposure of cancer cells to a chemotherapeutic drug. While communities in this network should reflect functional relatedness among proteins (i.e. similar biological functions, in general), we also expect that members of a community should share similarities in the observed modification type. Also associated with each of the 6 modification types is whether that particular type of modification became either more or less prominent after treatment with the drug. Since we have two types of labels associated with these nodes, we also sought to explore how these two labeling schemes (6 class vs. 2 class) aligned with the communities returned by the algorithm.

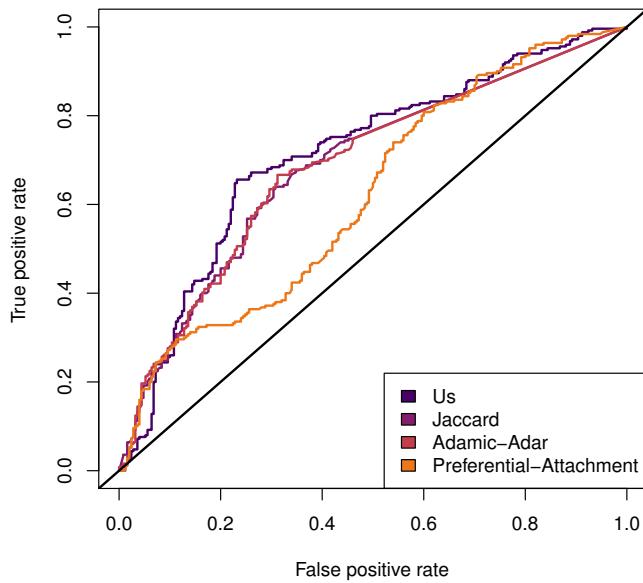


Figure 4.5: Link Prediction on the microbiome subject similarity network: The results for link prediction on the microbiome subject similarity network for the attributed SBM, Jaccard, Adamic–Adar and preferential attachment methods. The corresponding AUC values for these methods, respectively are, 0.71, 0.69, 0.69, and 0.62.

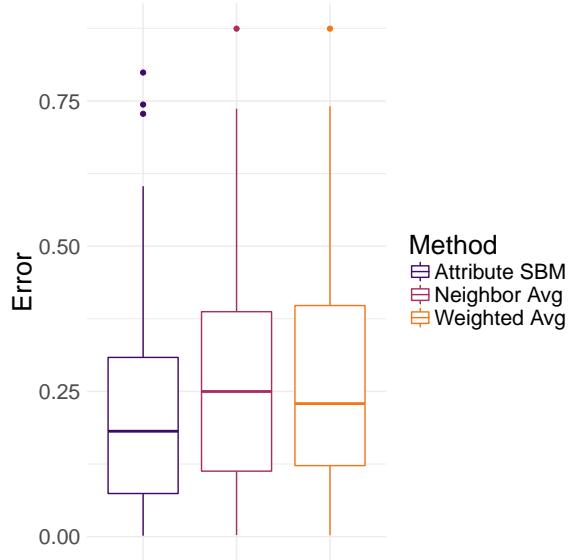


Figure 4.6: Collaborative Filtering Accuracy in Microbiome Subject Similarity Network: For each of the 121 nodes, we fit a model to the remaining 120 node network and given the node's closest neighbors (based on network connectivity) sought to predict its 5-dimensional attribute vector. The reported error is the relative error \mathcal{E} between the difference between the true attribute vector (\mathbf{x}_i) and its predicted attribute vector ($\hat{\mathbf{x}}_i$). The mean error in \mathbf{x}_i is 0.21, as opposed to the neighbor average and weighted neighbor averages, having errors of 0.26 and 0.27, respectively.

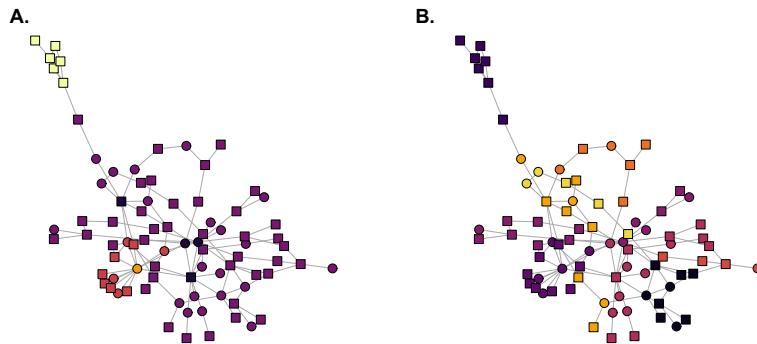


Figure 4.7: Protein interaction network. We visualize the 82 node protein interaction network under the classic stochastic block model **A.** and the attributed stochastic block model **B.** In both networks, nodes are colored by their community assignment and the node shape indicates whether the modification status increased (square) or decreased. **A.** Nodes colored according to the community partition under the stochastic block model. Nodes are assigned to one of five communities. **B.** Nodes are colored to the community partition under one of nine communities.

4.5.2.1 Data Pre-Processing

We downloaded the unweighted protein interaction network data and the modification information from the supplement of (22). We removed 13 nodes that were not connected to the giant component of the network and considered only the 82 node giant connected component.

4.5.2.2 Constructing Node Attributes

As previously described, each node is classified with 1 of 6 possible modification type. For each node, we created an attribute vector that captured the modification types of its neighbors. To do this, we considered the 4th order neighborhood of each node. That is, for each node, we collected its neighbors who were four hops or less away in the network. Then to define the value for attribute c of node i , or x_{ic} , we counted the number of fourth order neighbors of node i with label c . After defining these attributes across all nodes, for each of the 6 classes, we centered and scaled each attribute across all of the nodes to have mean 0 and unit variance.

Figure 7A-B show the results of fitting a classic SBM and attributed SBM, respectively. The 6 possible modifications exist for 3 biological processes that can either increase or decrease. The node shape reflects whether the experimental modification for a node increased (square) or decreased (circle) after treatment with the chemotherapeutic agent. Again by fitting an SBM with the model selection criterion in (32), 5 communities were identified. With our attributed SBM, 9 communities were identified. Note that using the attributed SBM created more communities in that it split up the purple core community under the classic SBM into more small communities. The implications of this new partition are explored with an entropy calculation based on the biological classifications of the protein in Figure 4.8.

Using the partition of the nodes under the classic and attributed stochastic block models, we sought to use the two different classifications of the nodes (6 class modification type and 2 class increase/decrease) to compute entropy of labels within communities. The expectation is that by incorporating attribute information that is related to the functional protein information into the community detection problem, we should see a decrease in the entropy over the classification labels in communities. In figure 8A-B, we plot the entropy for the 2 class and 6 class node classifications, respectively. We define E_c , the entropy for community c as,

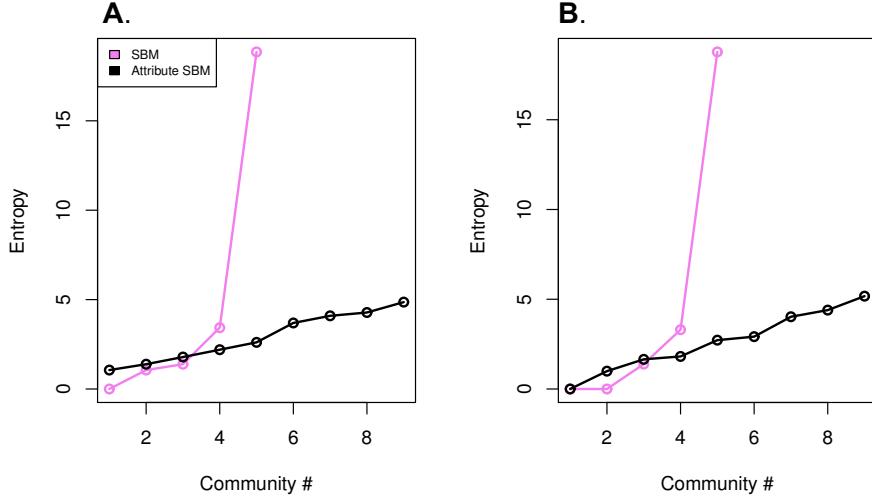


Figure 4.8: Community entropies in the protein interaction network. We studied the entropy of the 2 class and 6 class classifications of the nodes in **A.** and **B.**, respectively under the classic SBM (black) and attributed SBM (purple) partitions. For **A.** – **B.** the horizontal axis denotes the community index for the particular partition. Nodes belonged to 1 of 5 communities under the classic SBM and belong to 1 of 9 communities with the attributed SBM. Incorporating attributes under both classifications succeeds in breaking up a high entropy community (5) from the classic SBM partition to lower entropy communities in the attributed SBM partition.

$$E_c = - \sum_k p_k \log(p_k). \quad (4.13)$$

Here, k is indexing the unique classifications found in community c and p_k is the probability that a node in community c belonged to classification k in community c . In these plots the black and purple curves correspond to the fits of the classic and attribute SBM fits, respectively. Using both types of node classifications to compute these entropy quantities, we see that the attribute SBM succeeds in breaking up one high entropy community (5) from the classic SBM partition into lower entropy communities.

4.5.2.3 Link prediction experiments

We performed link prediction on the protein interaction network using the procedure described in section 4.1. Given that this protein network is sparse, none of the link prediction methods performed

particularly well. The AUC values for the attributed SBM, jaccard, adamic adar and preferential attachment were 0.61, 0.58, 0.58, and 0.54, respectively. The associated ROC curves are shown in figure 9.

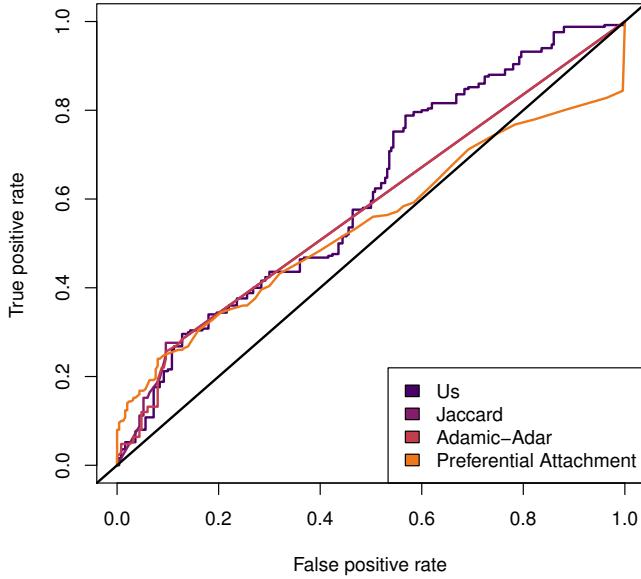


Figure 4.9: **Link Prediction in the protein interaction network.** Performing link prediction using the attributed SBM, Jaccard, Adamic Adar, and preferential attachment. The corresponding AUC curves for these methods were 0.61, 0.58, 0.58, and 0.51, respectively.

4.5.2.4 Collaborative filtering experiments

Collaborative filtering was performed using the method described in section 4.2. Note that unlike the microbiome sample similarity network, the edges in this network are unweighted and hence the neighbor average and weighted neighbor average methods produce the same result. We note that performing collaborative filtering with the attributed stochastic block model results in a lower mean error of 0.21 compared to that of 0.48 when using the neighbor average. Similar to figure 5, the box plots in Figure 4.10 represent the distribution of errors across each of the 82 nodes.

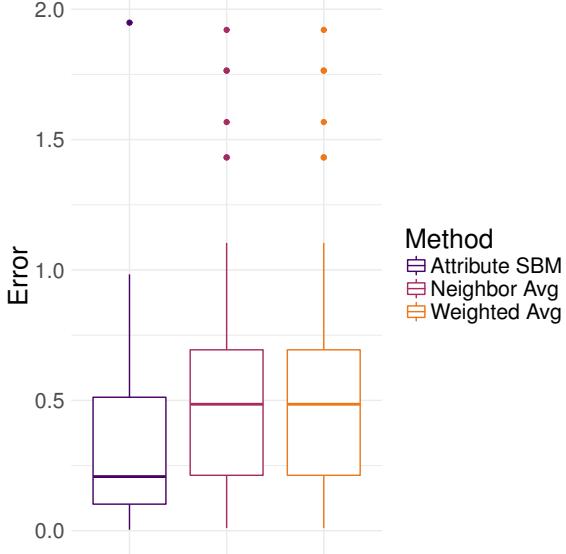


Figure 4.10: **Collaborative filtering in the protein interaction network.** For each of the 82 nodes, we fit a model to the remaining 81 node network and given the node’s closest neighbors (based on network connectivity) sought to predict its 6-dimensional attribute vector. The reported error is the relative error \mathcal{E} between the difference between the true attribute vector (\mathbf{x}_i) and its predicted attribute vector ($\hat{\mathbf{x}}_i$). The mean error in \mathbf{x}_i using the attributed SBM is 0.21, as opposed to the neighbor average error where it is 0.48.

4.6 Conclusion and Future Work

In this paper, we defined an attributed stochastic block model, where a node’s community assignment determines its connectivity and its attribute vector. Our model builds on previous work with attributed stochastic block models because it can handle multiple continuous attributes. The continuous attributes are modeled by a Gaussian mixture model, where the assumption is that the attributes for members for each community are parameterized by a unique multivariate Gaussian. Since community detection results are often difficult to validate due to the lack of unlabeled network data, or definition of what a correct partition means, we quantified the ability of the fitted attributed stochastic block model to represent a particular network by performing link prediction and collaborative filtering tasks. Applying link prediction and collaborative filtering to two biological networks, we observed that the attributed SBM is useful for these applications.

Future work could extend the model to handle a combination of multiple discrete and continuous attributes. Further, while the inference or understanding of fitting a stochastic block model to

weighted networks is not well understood, figuring out how to integrate edge weights and attributes in determining community structure could be useful. Finally, we briefly discussed a detectability problem in figure 3. Interesting future work could aim to characterize the properties of the attributes and connectivity and how they relate to effective identification of community structure.

Networks used across fields are becoming increasingly complex, often with multiple sources of information to integrate in order to make a conclusion for the data. Our approach to an attributed SBM advances the understanding of how to jointly consider attribute and connectivity information in a probabilistic framework.

CHAPTER 5

Community detection for understanding burn inhalation injury

CHAPTER 6

Conclusion and future work

BIBLIOGRAPHY

- <https://snap.stanford.edu/data>.
- <http://www-personal.umich.edu/~mejn/netdata/>.
- <https://github.com/vtraag/louvain-igraph>.
- <https://graph-tool.skewed.de/>.
- Abbe, E., Bandeira, A. S., and Hall, G. (2016). Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487.
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.
- Aghaeepour, N., Finak, G., Hoos, H., Mosmann, T. R., Brinkman, R., Gottardo, R., Scheuermann, R. H., Consortium, F., Consortium, D., et al. (2013). Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, 10(3):228.
- Aghaeepour, N., Ganio, E. A., Mcilwain, D., Tsai, A. S., Tingle, M., Van Gassen, S., Gaudilliere, D. K., Baca, Q., McNeil, L., Okada, R., et al. (2017). An immune clock of human pregnancy. *Science immunology*, 2(15):eaan2946.
- Aicher, C., Jacobs, A. Z., and Clauset, A. (2014). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.
- Aicher, C., Jacobs, A. Z., and Clauset, A. (2015a). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.
- Aicher, C., Jacobs, A. Z., and Clauset, A. (2015b). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.
- Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014.
- Attias, H. (2000). A variational bayesian framework for graphical models. In *Advances in neural information processing systems*, pages 209–215.
- Baldassano, S. N. and Bassett, D. S. (2016). Topological distortion and reorganized modular structure of gut microbial co-occurrence networks in inflammatory bowel disease. *Scientific reports*, 6:26087.
- Barbillon, P., Donnet, S., Lazega, E., and Bar-Hen, A. (2015). Stochastic block models for multiplex networks: an application to networks of researchers. *arXiv preprint arXiv:1501.06444*.
- Barry, A. E., Leliwa-Sytek, A., Tavul, L., Imrie, H., Migot-Nabias, F., Brown, S. M., McVean, G. A., and Day, K. P. (2007). Population genomics of the immune evasion (var) genes of plasmodium falciparum. *PLoS pathogens*, 3(3):e34.
- Benaych-Georges, F. and Nadakuditi, R. R. (2011). The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics*, 227(1):494–521.

- Bendall, S. C., Nolan, G. P., Roederer, M., and Chattopadhyay, P. K. (2012). A deep profiler's guide to cytometry. *Trends in immunology*, 33(7):323–332.
- Bender, E. A. and Canfield, E. R. (1978). The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307.
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.
- Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C. I., Gómez-Gardeñes, J., Romance, M., Sendina-Nadal, I., Wang, Z., and Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122.
- Bonacci, T., Audebert, S., Camoin, L., Baudelet, E., Bidaut, G., Garcia, M., Witzel, I.-I., Perkins, N. D., Borg, J.-P., Iovanna, J.-L., et al. (2014). Identification of new mechanisms of cellular response to chemotherapy by tracking changes in post-translational modifications by ubiquitin and ubiquitin-like proteins. *Journal of proteome research*, 13(5):2478–2494.
- Brandes, U., Lerner, J., and Nagel, U. (2011). Network ensemble clustering using latent roles. *Advances in Data Analysis and Classification*, 5(2):81–94.
- Brandes, U., Lerner, J., Nagel, U., and Nick, B. (2009). Structural trends in network ensembles. In *Complex networks*, pages 83–97. Springer.
- Browet, A., Absil, P.-A., and Van Dooren, P. (2011). Community detection for hierarchical image segmentation. In *IWCIA*, volume 11, pages 358–371. Springer.
- Chen, H., Lau, M. C., Wong, M. T., Newell, E. W., Poidinger, M., and Chen, J. (2016). Cytofkit: a bioconductor package for an integrated mass cytometry data analysis pipeline. *PLoS computational biology*, 12(9):e1005112.
- Clauset, A., Moore, C., and Newman, M. E. (2007). Structural inference of hierarchies in networks. In *Statistical network analysis: models, issues, and new directions*, pages 1–13. Springer.
- Combe, D., Largeron, C., Géry, M., and Egyed-Zsigmond, E. (2015). I-louvain: An attributed graph clustering method. In *Advances in Intelligent Data Analysis XIV*, pages 181–192. Springer.
- Costanzo, M., Baryshnikova, A., Bellay, J., Kim, Y., Spear, E. D., Sevier, C. S., Ding, H., Koh, J. L., Toufighi, K., Mostafavi, S., et al. (2010). The genetic landscape of a cell. *science*, 327(5964):425–431.
- Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005a). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008.
- Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005b). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008.
- Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and computing*, 18(2):173–183.
- De Domenico, M., Lancichinetti, A., Arenas, A., and Rosvall, M. (2015a). Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, 5(1):011027.

- De Domenico, M., Nicosia, V., Arenas, A., and Latora, V. (2015b). Structural reducibility of multilayer networks. *Nature communications*, 6.
- De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M. A., Gómez, S., and Arenas, A. (2013). Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022.
- Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011a). Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701.
- Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011b). Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.
- Deng, W., Patil, R., Najjar, L., Shi, Y., and Chen, Z. (2014). Incorporating community detection and clustering techniques into collaborative filtering model. *Procedia Computer Science*, 31:66–74.
- Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM.
- Ding, T. and Schloss, P. D. (2014). Dynamics and associations of microbial community types across the human body. *Nature*, 509(7500):357–360.
- Faust, K., Sathirapongsasuti, J. F., Izard, J., Segata, N., Gevers, D., Raes, J., and Huttenhower, C. (2012). Microbial co-occurrence relationships in the human microbiome. *PLoS computational biology*, 8(7):e1002606.
- Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.
- Fortunato, S. and Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, 659:1–44.
- Friedman, J. and Alm, E. J. (2012). Inferring correlation networks from genomic survey data. *PLoS computational biology*, 8(9):e1002687.
- Ghasemian, A., Zhang, P., Clauset, A., Moore, C., and Peel, L. (2016). Detectability thresholds and optimal algorithms for community structure in dynamic networks. *Physical Review X*, 6(3):031005.
- Gilbert, A. C. and Levchenko, K. (2004). Compressing network graphs. In *Proceedings of the LinkKDD workshop at the 10th ACM Conference on KDD*, volume 124.
- Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.
- Gleich, D. F. (2015). Pagerank beyond the web. *SIAM Review*, 57(3):321–363.
- Greene, D. and Cunningham, P. (2013). Producing a unified graph representation from multiple social network views. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 118–121. ACM.

- Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.
- Guimerà, R. and Sales-Pardo, M. (2009). Missing and spurious interactions and the reconstruction of complex networks. *Proceedings of the National Academy of Sciences*, 106(52):22073–22078.
- Han, Q., Xu, K., and Airoldi, E. (2015). Consistent estimation of dynamic and multi-layer block models. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1511–1520.
- Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.
- Hric, D., Peixoto, T. P., and Fortunato, S. (2016). Network structure, metadata, and the prediction of missing nodes and annotations. *Physical Review X*, 6(3):031038.
- Hu, D., Ronhovde, P., and Nussinov, Z. (2012). Phase transitions in random potts systems and the community detection problem: spin-glass type and dynamic perspectives. *Philosophical Magazine*, 92(4):406–445.
- Iacovacci, J., Wu, Z., and Bianconi, G. (2015). Mesoscopic structures reveal the network between the layers of multiplex datasets. *arXiv preprint arXiv:1505.03824*.
- Jaakkola, T. (2001). 10 tutorial on variational approximation methods. *Advanced mean field methods: theory and practice*, page 129.
- Jacobs, A. Z. and Clauset, A. (2014). A unified view of generative models for networks: models, methods, opportunities, and challenges. *arXiv preprint arXiv:1411.4070*.
- Jung, K., Heo, W., and Chen, W. (2012). Irie: Scalable and robust influence maximization in social networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 918–923. IEEE.
- Karrer, B. and , M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107.
- Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107.
- Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM.
- Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, 2(3):203–271.
- Kossinets, G. and Watts, D. J. (2009). Origins of homophily in an evolving social network. *American journal of sociology*, 115(2):405–450.
- Lahti, L., Salojärvi, J., Salonen, A., Scheffer, M., and De Vos, W. M. (2014). Tipping elements in the human intestinal ecosystem. *Nature communications*, 5.

- Lancichinetti, A. and Fortunato, S. (2009). Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117.
- Lancichinetti, A. and Fortunato, S. (2011). Limits of modularity maximization in community detection. *Physical review E*, 84(6):066122.
- Larremore, D. B., Clauset, A., and Buckee, C. O. (2013). A network approach to analyzing highly recombinant malaria parasite genes. *PLoS computational biology*, 9(10):e1003268.
- Latouche, P., Birmelé, E., and Ambroise, C. (2011). Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, pages 309–336.
- Layeghifard, M., Hwang, D. M., and Guttman, D. S. (2017). Disentangling interactions in the microbiome: a network perspective. *Trends in microbiology*, 25(3):217–228.
- Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123.
- Levine, J. H., Simonds, E. F., Bendall, S. C., Davis, K. L., El-ad, D. A., Tadmor, M. D., Litvin, O., Fienberg, H. G., Jager, A., Zunder, E. R., et al. (2015). Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197.
- Lisewski, A. M., Quiros, J. P., Ng, C. L., Adikesavan, A. K., Miura, K., Putluri, N., Eastman, R. T., Scanfeld, D., Regenbogen, S. J., Altenhofen, L., et al. (2014). Supergenomic network compression and the discovery of exp1 as a glutathione transferase inhibited by artesunate. *Cell*, 158(4):916–928.
- Lorrain, F. and White, H. C. (1971). Structural equivalence of individuals in social networks. *The Journal of mathematical sociology*, 1(1):49–80.
- Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45.
- Marusyk, A., Almendro, V., and Polyak, K. (2012). Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*, 12(5):323.
- Meunier, D., Lambiotte, R., Fornito, A., Ersche, K. D., and Bullmore, E. T. (2009). Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878.
- Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc.
- Nadakuditi, R. R. and Newman, M. E. (2012). Graph spectra and the of community structure in networks. *Physical review letters*, 108(18):188701.

- Nadakuditi, R. R. and Newman, M. E. (2013). Spectra of random graphs with arbitrary expected degrees. *Physical Review E*, 87(1):012803.
- Newman, M. E. (2002). Assortative mixing in networks. *Physical review letters*, 89(20):208701.
- Newman, M. E. (2006a). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.
- Newman, M. E. (2006b). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.
- Newman, M. E. and Clauset, A. (2016). Structure and inference in annotated networks. *Nature Communications*, 7:11863.
- Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.
- Newman, M. E. J. (2006c). Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104.
- Ni, J., Tong, H., Fan, W., and Zhang, X. (2015). Flexible and robust multi-network clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 835–844. ACM.
- Noh, J. D. and Rieger, H. (2004). Random walks on complex networks. *Physical review letters*, 92(11):118701.
- Onnela, J.-P., Fenn, D. J., Reid, S., Porter, M. A., Mucha, P. J., Fricker, M. D., and Jones, N. S. (2012). Taxonomies of networks from community structure. *Physical Review E*, 86(3):036104.
- Paul, S. and Chen, Y. (2015). Community detection in multi-relational data with restricted multi-layer stochastic blockmodel. *arXiv preprint arXiv:1506.02699*.
- Peel, L., Larremore, D. B., and Clauset, A. (2017). The ground truth about metadata and community detection in networks. *Science Advances*, 3(5):e1602548.
- Peixoto, T. P. (2013). Eigenvalue spectra of modular networks. *Physical review letters*, 111(9):098701.
- Peixoto, T. P. (2014). Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. *Physical Review E*, 89(1):012804.
- Peixoto, T. P. (2015). Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Phys. Rev. E*, 92:042807.
- Peixoto, T. P. (2018). Nonparametric weighted stochastic block models. *Physical Review E*, 97(1):012306.
- Peng, C., Kolda, T. G., and Pinar, A. (2014). Accelerating community detection by using k-core subgraphs. *arXiv preprint arXiv:1403.2226*.
- Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

- Porter, M. A., Onnela, J.-P., and Mucha, P. J. (2009a). Communities in networks. *Notices of the AMS*, 56(9):1082–1097.
- Porter, M. A., Onnela, J.-P., and Mucha, P. J. (2009b). Communities in networks. *Notices of the AMS*, 56(9):1082–1097.
- Radicchi, F. (2013). Detectability of communities in heterogeneous networks. *Physical Review E*, 88(1):010801.
- Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1):016110.
- Reichardt, J. and Leone, M. (2008). (un) detectable cluster structure in sparse networks. *Physical review letters*, 101(7):078701.
- Rombach, M. P., Porter, M. A., Fowler, J. H., and Mucha, P. J. (2014). Core-periphery structure in networks. *SIAM Journal on Applied mathematics*, 74(1):167–190.
- Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.
- Sarkar, S., Henderson, J. A., and Robinson, P. A. (2013). Spectral characterization of hierarchical network modularity and limits of modularity detection. *PloS one*, 8(1):e54383.
- Shai, S., Stanley, N., Granell, C., Taylor, D., and Mucha, P. J. (2017). Case studies in network community detection. *arXiv preprint arXiv:1705.02305*.
- Shi, Y., Larson, M., and Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):3.
- Snijders, T. A. and Nowicki, K. (1997a). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100.
- Snijders, T. A. and Nowicki, K. (1997b). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100.
- Soundarajan, S. and Hopcroft, J. (2012). Using community information to improve the precision of link prediction methods. In *Proceedings of the 21st International Conference on World Wide Web*, pages 607–608. ACM.
- Stanley, N., Kwitt, R., Niethammer, M., and Mucha, P. J. (2017). Compressing networks with super nodes. *arXiv preprint arXiv:1706.04110*.
- Stanley, N., Shai, S., Taylor, D., and Mucha, P. J. (2016). Clustering network layers with the strata multilayer stochastic block model. *IEEE transactions on network science and engineering*, 3(2):95–105.
- Taylor, D., Shai, S., Stanley, N., and Mucha, P. J. (2015). Enhanced detectability of community structure in multilayer networks through layer aggregation. *arXiv preprint arXiv:1511.05271*.
- Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

- Tong, H., Faloutsos, C., and Pan, J.-Y. (2008). Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, 14(3):327–346.
- Traud, A. L., Kelsic, E. D., Mucha, P. J., and Porter, M. A. (2011). Comparing community structure to characteristics in online collegiate social networks. *SIAM review*, 53(3):526–543.
- Tsuda, K. and Kudo, T. (2006). Clustering graphs by weighted substructure mining. In *Proceedings of the 23rd international conference on Machine learning*, pages 953–960. ACM.
- Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C. M., Knight, R., and Gordon, J. I. (2007). The human microbiome project. *Nature*, 449(7164):804–810.
- Ugander, J., Backstrom, L., and Kleinberg, J. (2013). Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1307–1318. International World Wide Web Conferences Steering Committee.
- Valles-Catala, T., Massucci, F. A., Guimera, R., and Sales-Pardo, M. (2014). Stochastic block models reveal the multilayer structure of complex networks. *arXiv preprint arXiv:1411.1098*.
- van den Heuvel, M. P. and Sporns, O. (2013). Network hubs in the human brain. *Trends in cognitive sciences*, 17(12):683–696.
- Wang, P., Xu, B., Wu, Y., and Zhou, X. (2014). Link prediction in social networks: the state-of-the-art. *arXiv preprint arXiv:1411.5118*.
- Wang, Y. J. and Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19.
- Xiang, T. and Gong, S. (2008). Spectral clustering with eigenvector selection. *Pattern Recognition*, 41(3):1012–1029.
- Yang, J. and Leskovec, J. (2012). Community-affiliation graph model for overlapping network community detection. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1170–1175. IEEE.
- Yang, J. and Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213.
- Yang, J., McAuley, J., and Leskovec, J. (2013). Community detection in networks with node attributes. In *Data mining (ICDM), 2013 ieee 13th international conference on*, pages 1151–1156. IEEE.
- Yang, L., Jin, D., He, D., Fu, H., Cao, X., and Fogelman-Soulie, F. (2017). Improving the efficiency and effectiveness of community detection via prior-induced equivalent super-network. *Scientific Reports*, 7(1):634.
- Zapién-Campos, R., Olmedo-Álvarez, G., and Santillán, M. (2015). Antagonistic interactions are sufficient to explain self-assemblage of bacterial communities in a homogeneous environment: a computational modeling approach. *Frontiers in Microbiology*, 6:489.
- Zare, H., Shooshtari, P., Gupta, A., and Brinkman, R. R. (2010). Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC bioinformatics*, 11(1):403.

- Zdeborová, L., Zhang, P., and Zhou, H.-J. (2016). Fast and simple decycling and dismantling of networks. *Scientific Reports*, 6.
- Zhang, P., Krzakala, F., Reichardt, J., and Zdeborová, L. (2012). Comparative study for inference of hidden classes in stochastic block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(12):P12021.
- Zitnik, M. and Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198.