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

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 biological correlations, or communication between individuals in a social network, network analysis tools enable interpretation, prediction, and visualization of patterns in the data. Community detection is a well-developed subfield of network analysis, where the objective is to cluster nodes into 'communities' based on their connectivity patterns. There are many useful 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 poses challenges. In this thesis, we address three types of challenging network data and how to adapt standard community detection approaches to handle these situations. In particular, we focus on networks that are large, attributed, and multilayer. 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 and develop a test to quantify the extend to which connectivity and community structure align. Finally, we apply several case studies of these methods on biological and social networks. The implications of this work help to advance the understand of network clustering, network compression, and assist in tasks such as, link prediction, and collaborative filtering

To Thomas. Thank you for being you.

ACKNOWLEDGEMENTS

I'm glad this is the part of the thesis that people like to read because I have many thanks to share. First, thank you to my advisor, Peter. Thank you for always treating me like a scientist, and not a student. I think that a trait of a great advisor is their willingness to work collaboratively with their students, and Peter does this incredibly well. Thank you for always being positive about results (even if no positivity was warranted), for providing suggestions, and for allowing me to work on whatever I wanted to. Thank you for helping me through 'existential angst' and for supporting me in whatever career path I wanted to take. Thank you for always making sure there was a grant to pay my salary and for all of the meetings and Slack conversations. I will forever remember to ignore the gong and how one good result is already more than most of the literature. I am forever grateful for all of your support.

Next, thank you to my pseudo second advisor, Marc. Thank you for always reading my write ups and papers and always having great questions and suggestions. I always admire how successful, creative, and humble you are (oh and a great sense of humor). Thank you for all of your time and Monday meetings. The Monday meetings with Peter and Roland are some of my best memories.

Thank you to Saray and Dane who have played an important role in mentoring me as a beginning grad student and helping me to write my first paper. Dane, thank you for your very detailed editing and notation considerations. I will forever remember the suggestions you made on the first paper. Saray, I am so lucky to work with you and even more lucky to have you as a friend. I rarely have met people that I can communicate with through eye contact. Our entropy together made everything more fun, from yard time, to 'getting a mix', to flying with the random physicist to Zaragoza and getting displaced in a tiny elevator, to eating way too much at Weaver Street.

Thank you to my committee members, Jeremy, Tamara, Laura and David. Jeremy, thank you for being my first introduction to research in grad school. I will always remember p53 signaling dynamics and microRNAs. Tamara, I am so happy that I got to take your class in my second year, which inspired me so much and even made me wish I could switch to computer vision. You are

a great role model as a smart and creative researcher. Laura and David, thank you for all of your suggestions in committee meetings and for reading this thesis.

I have been lucky to interact with a lot of great people over the years in the Mucha research group. Thanks to Nishant, Wayne, Sam, Howdy, Clara, Eun, Peter D, Nic, and Sean for brightening up Chapman.

Thank you to the people who make BCB run- Tim Elston, Will Valdar, and John Cornett. I know you all work very hard for BCB and I think we have a great group of creative students. We all owe so much gratitude to John Cornett who is always friendly, positive, responsive, and on top of things. I am also lucky to have met great friends in BCB who I have done homework with, looked up to, and gotten advice from. Thank you, Bryan, Dan, Greg, and Paul.

Thanks to living in Chapel Hill, I was fortunate to make some incredible friends. To my super strong (literally strong) lady friends Jess, Mimi, and Libby: Thank you for all the nights we spent laughing and climbing. These are some of the best times. Thank you Andrew for being the most incredible nerd friend and one of the kindest people I have ever met. I can always count on you for awesome conversation.

Last but not least, I owe a huge amount of gratitude to my family. First to my parents Pat and Eric who have supported me every day of my life. They have never put any pressure on me to do anything and support all of my dreams unconditionally. Most importantly, they are really friendly and fun people. I couldn't choose better parents. Thank you for tolerating my un responded text messages, my inability to mail a letter or find a stamp, and for helping me through tough times. Next to my brother, Mike. I admire you so much for always following your dreams and doing what feels right to you. Aside from being great at everything you do, you are such a kind, wonderful person. I hope you don't find any mistakes in this thesis or ask about consistency.

Finally, thank you Thomas for supporting me in every possible way. I'm so happy that grad school lead me to you. You have enhanced my life in every way and inspire me every day to be a scientist. I am so lucky to have a great role model who works so hard, is so talented, and so kind. Thank you for always pushing me pursue things I didn't think that I could. Thank you for always telling me 'shhhhhhh' when I started to get stressed. You are my favorite Dub.

OK, this is the end. Thank you everyone that read to the end.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS & COMMON NOTATION

SBM	Stochastic Block Model
EM	Expectation Maximization
sMLSBM	Strata Multilayer Stochastic Block Model
MLSBM	Multilayer Stochastic Block Model
$\mathbf{A}$	Network adjacency Matrix
a_{ij}	The ijth entry of adjacency matrix, $\mathbf{A}$
SN	Super Node
$\mathbf{z}$	For a network with N nodes, this is the length N vector of node-to-community assignments
z_i	The community assignment of node i
$\mathbf{Z}$	The indicator matrix of node-to-community assignments
z_{ik}	A binary indicator of whether node i belongs to community k.
ALI	Acute lung injury
OTU	Operational taxonomic unit
DAG	Directed acyclic graph
MCMC	Markov Chain Monte Carlo
p_{in}	Within-community edge probability under stochastic block model
p_{out}	Between-community edge probability under stochastic block model
LP	Label propagation

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 (139), clustering (52), and prediction tasks (147; 46). Motivated by problems in fields such as, biology (79), medicine (8), neuroscience (17), and social science (58), 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 data interpretation. In this thesis, we present three new methods that enable a more thorough understanding of the structural organization patterns in networks 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, standard community detection approaches are often insufficient. 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, efficient summarization, prediction, and classification.

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

tool in computational biology, as it assists in tasks such as, clustering, biological interpretation, and prioritizing further experiments. Finally, we discuss challenges in the field of community detection and how this work addresses some of these problems.

1.1 Network Notation and Basic Summarization

In this section, we provide some basic notation and summarization techniques for representing and summarizing networks.

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 collaboration network between the six people that made the work in this thesis possible. An edge between a pair of people indicates if they have written a paper together.

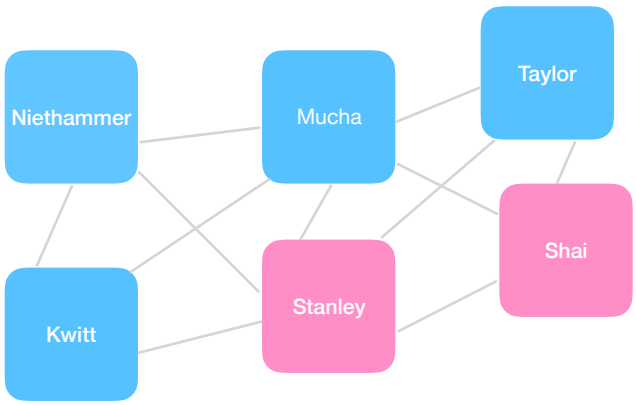

Figure 1.1: **A simple network example (coauthorship).** A co-authorship network with an edge between a pair of people if they have written a paper together.

Such a network with edges simply representing whether or not a pair of nodes interact scientifically is an example of an *undirected, unweighted* network. Among undirected networks, edges can also be weighted, which quantifies pairwise similarity between a node pair. 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,

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

In the *weighted* case of undirected networks, 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,

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

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 because they are not involved in any work in this thesis.

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, ranking the node by their importance or *centrality* in the network, clustering nodes, and predicting the existence of a link between a node pair. Toy networks, such as the one presented in Figure 1.1 or in a textbook often look deceptively clean and well-structured. In reality, most network data is large, messy, and often referred to as a hairball. This term alludes to the difficulty of immediately discerning structure or interpreting meaning from the network due to the large amount of presented information. An example of a typical hairball is shown in Figure 1.2. Here, there are many nodes and edges that from immediate inspection, it may seem like the relational patterns are too difficult to untangle and interpret.

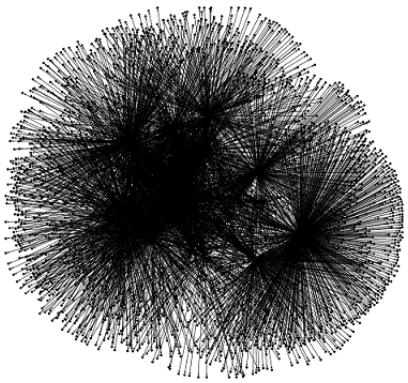

Figure 1.2: **Hairball network.** Networks are often noisy data structures and lack an immediate straight forward structural interpretation. *Image from* `https://cs.umd.edu`.

1.1.2.1 Example: A network representation of single cell data and simple summary statistics

An initially overwhelming network structure can be mediated by tools to break down, quantify, and characterize structural patterns. In this section, we will describe a few of the essential summary statistics and analyses that can be performed and will be seen throughout this thesis.

To illustrate these quantities in an applied context, we will compute them on an example network shown in Figure 1.3. This network is constructed from a single cell mass cytometry dataset, which was originally described in Ref. (149) and released publicly and processed using the Cytofkit R package (32). Each node represents a single cell and is represented with 52 features for a mass cytometry analysis. Briefly, mass cytometry is a technique to simultaneously measure multiple immunological features in a cell or tissue (19). From this data matrix, we created a network 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

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 structure. The first most basic summary statistic is known as *degree*. Given the adjacency matrix for an undirected network, $\mathbf{A}$, the degree of node i, degree(i) is computed as,

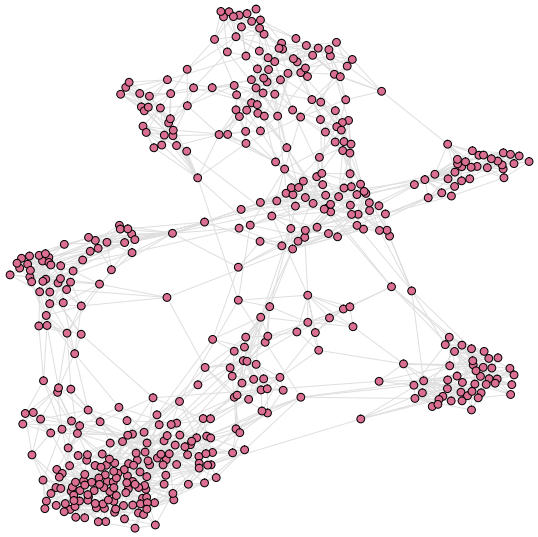

Figure 1.3: **Network of single cells.** We constructed a network from mass cytometry profiling among 500 cells in single cell dataset. Each cell has 52 measured immune features. In this network, each node is a single cell and is connected to its 5 nearest neighbors.

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

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. We visualize degree distribution in Figure 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 52 dimensional space. 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, network centrality measures.

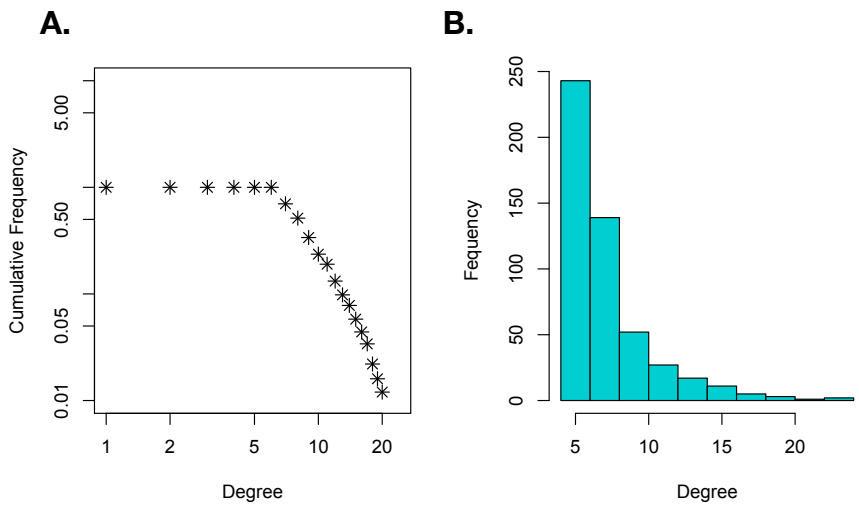

Figure 1.4: **Degree distribution for the single cell network.** We visualize the degree distribution in the single cell network presented in Figure 1.3. **A.** We compute a cumulative distribution plot for degree. **B.** Node degrees can also be visualized with a simple histogram.

1.1.2.3 Centrality

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 differently, each is intended to capture a different structural aspect of 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,

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

Betweenness centrality

Betweenness centrality quantifies node importance, based on how many shortest paths go through a

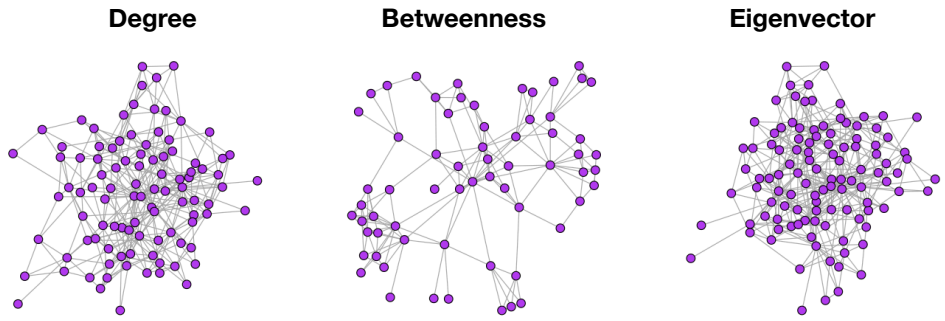

Figure 1.5: **Centralities on the single cell network.** The second order ego network for the highest centrality nodes in the single cell network according to degree, betweenness, and eigenvector in the left, center, and right plots, respectively. These plots are meant to emphasize how each of these centrality measures prioritizes different kind of stucture.

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,

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

where σ_{jt} is the total number of shortest paths between a pair of nodes, j 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 its neighboring nodes. That is, a node connected to other 'important' or high degree nodes should be ranked higher than one connected to many low degree nodes [1]. The eigenvector centrality for node i, can be computed using the spectra of the adjacency matrix, $\boldsymbol{A}$. In particular, the vector of centralities, $\mathbf{x}$ is the one satisfying the eigenvector equation,

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \tag{1.4}$$

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

[1] If you want to compliment a friend, it is nicer to say that they have high eigenvector centrality than high degree centrality.

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. Under each of the centrality measures, we selected the the highest-ranked centrality node and focused in its local ego network. This is shown for degree, betweenness, and eigenvector centralities in the left, middle, and right panels respectively. In particular from these high centrality nodes, we visualized their corresponding order 2 ego networks. An ego network for node i is simply the subgraph of all nodes within two hops of 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.

1.2 Introduction to community detection

While centrality measures allow for the prioritization of individual nodes in the network, it is also useful to look at sets of similar nodes in terms of how they are situated in the network. Each of these sets of similar nodes is known as a *community*. A community in a network is broadly defined as a set of nodes 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 define sets of nodes that maximize the within-community node similarity. 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 Figure 1.6. Communities in the network are outlined with pink dotted lines.

Alternatively, networks can have a disassortative 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. Similar to how the shape or distribution of a set

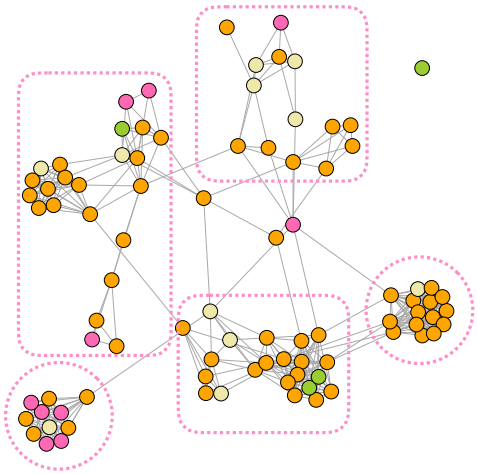

Figure 1.6: **Assortative Community Structure.** This network is an example of assortative community structure, where nodes are tightly connected to each other and more sparsely connected to the rest of the network. Each community is outlined with a pink dotted line.

of points in high dimensional space informs the ideal clustering algorithm to use, aspects of these diverse types of community structure often prescribe which algorithm to use. For a great explanation about common types of community structure in network data which patterns have been observed in the human brain, please refer to Betzel *et al.* (22).

In this section, we have only briefly introduced the history and intuition behind community detection. Since it is a well-developed domain of network science, the interested reader can refer to one of the comprehensive review articles (76; 52; 128; 95)

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 criterion or through 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, and probabilistic 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 maximization, as those are the the approaches considered throughout the novel work in this thesis.

1.3.1 Notation for Community Detection

We first define some common notation for community detection. For a network with N nodes, we use a community detection algorithm to separate these nodes into K communities. To encode the node-to-community assignments, we use the length N vector, $\mathbf{z}$, where z_i gives the community assignment for node i. For some applications, we also specify the $N \times K$ matrix, $\mathbf{Z}$, which is a binary indicator matrix, where z_{ik} indicated whether node i is assigned to community k. These two pieces of notation will be used across each of the described algorithms.

1.3.2 Quality function maximization with modularity

In quality function optimization approaches one first specifies an objective function in terms of a partition of the nodes. The most common quality function for this task is known as modularity (101). Modularity first defines a null model for community structure where edges are places between groups randomly. With this as the starting point, the partition that optimizes modularity is the one that is maximally different from this null model. In particular, this null model is a random graph model, known as the configuration model (20). To generate an N-node network from the configuration model, one first specifies a fixed degree sequence, $D = \{k_i, k_2, \ldots, k_N\}$. 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 $P(a_{ij} = 1)$.

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

Here, k_i and k_j represent the degree 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 (104). We define the modularity quality function, Q as,

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

Here, γ is a resolution parameter (123) that controls the scale of community size. Large values of γ favor more small communities while smaller values enforce fewer large communities.

In order to determine $\mathbf{z}$, the most computationally efficient approach is known as the Louvain algorithm (23). The Louvain algorithm is an agglomerative heuristic, which initially starts with each node in its own community and in the first pass 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 continued 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 (93) to image segmentation (29). It has also shown to be effective in clustering high dimensional data that has been used to create a network. In Figure 1.7, we used tSNE (88) to project the 52-dimensional single cell data into 2 dimensions. Points are colored by their cluster assignment according to k-means. We first performed k-means on the original 51 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 clusters. Moreover, in this example, the Louvain algorithm maximized modularity by partitioning the network into 10 clusters. To compare the results under the same number of clusters, we also clustered the original data into 10 clusters. From these two partitions, we observe that creating a network representation of the data before clustering assists in identifying the smaller, less prominent clusters.

1.3.3 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 the inferred community assignments. For example, given nodes i and j, one may model $P(a_{ij} = 1)$

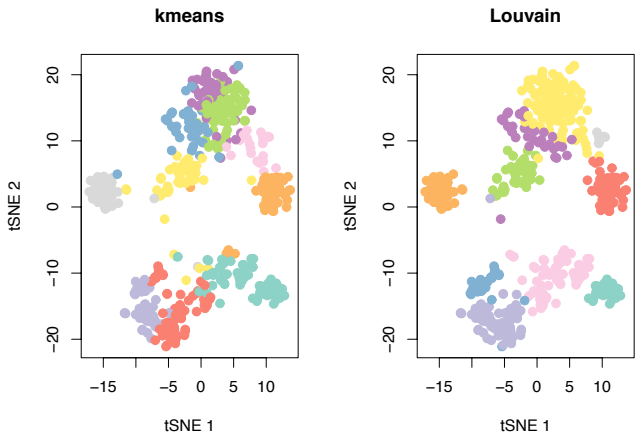

Figure 1.7: **A comparison of k-means and the Louvain algorithm on the single cell network.** A comparison of the results of clustering results on the the single cell dataset through k-means on the original 52-dimensional data (left) and by the Louvain algorithm on the nearest neighbor network (right). Each of the single cells (or nodes in the nearest neighbor network) is visualized by a 2-dimensional projection frin tSNE. Points are colored by their cluster membership under k-means on the original data (left) and Louvain community detection (right). Applying community detection to the nearest neighbor network seems to smooth out the partition and identify some smaller clusters.

as $g(z_i, z_j)$, where $g(\cdot)$ is some rule based on the node-to-community assignments. Two common probabilistic community detection models are the stochastic block model (131) and the affiliation model (152). The definition and description of these models and inference techniques are described in depth in this section. To facilitate working with probabilistic models, we first introduce some notation and background on inference techniques.

1.3.3.1 Probabilistic graphical models for statistical inference

Probabilistic community detection methods are one approach to community detection that seek to model edge existence based on the inferred 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. This is accomplished through likelihood optimization. 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.

When modeling the node-to-community assignments in a network, we often have at least two random variables and their dependency relationships to understand. First, we are interested in $\mathbf{z}$, the node-to-community assignments, and $\mathbf{A}$, the observed adjacency matrix. Probabilistic graphical

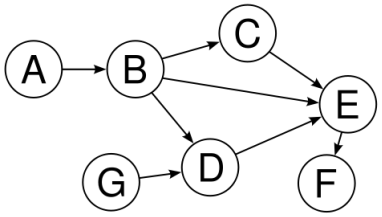

Figure 1.8: **Directed Acyclic Graph.** A directed acyclic graph (DAG) is formed based on dependency between random variable and allows for a fully factorized probability distribution. Nodes represent random variables and a directed edge from node i to node j indicates that node j depends on node i.

models (73) enable efficient specification and manipulation of large probability distributions through semantic structures.

As a brief example, 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 is computationally useful. A DAG between the set of random variables, $\{A, B, C, D, E, F\}$ is shown in Figure 1.8. Each node in the graphical model represents an random variable and a directed edge from node i to node j implies that node j depends on node i.

To translate a DAG between a set of N random variables, $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ (also in this context referred to as a Bayesian network) to its joint distribution, we rely on the chain rule for Bayesian networks (73), which specifies that a DAG factors according to its parent/child relationships with,

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

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,

$$\begin{aligned} P(A, B, C, D, E, F) = P(A)P(B \mid A)P(C \mid B) \\ \times P(D \mid B, G)P(E \mid D, B, C)P(F \mid E). \end{aligned} \tag{1.8}$$

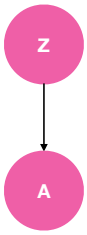

Figure 1.9: **SBM Graphical Model.** A graphical model is used to model the dependency between the node-to-community assignments, $\mathbf{z}$ and the observed network adjacency matrix, $\mathbf{A}$.

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

1.3.3.2 Stochastic Block Model

In this section, we introduce the most popular probabilistic model for community structure, known as the Stochastic Block Model (132). This model is popular and has been studied extensively, due to its simplicity and intuitive interpretation. 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. For an undirected, unweighted network with adjacency matrix $\mathbf{A}$, we seek to partition each of the N nodes into one of K communities. We denote the 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 dependency relationship between the node-to-community assignments ($\mathbf{z}$) and the network's adjacency matrix ($\mathbf{A}$). 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. To model the objective that members within and between communities connect in characteristic ways, 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 1.3.3.1, we can specify the complete data log likelihood between $\mathbf{z}$ and $\mathbf{A}$ as,

$$\log P(\mathbf{z}, \mathbf{A}) = \log(P(\mathbf{A} \mid \mathbf{z})) + \log(P(\mathbf{z})) \tag{1.9}$$

To further specify these communities, we will define additional notation. First, let $\boldsymbol{\pi}_{K\times K} = \{\pi_{ij}\}$ 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,

$$P(a_{ij} = 1) \sim \text{Bernoulli}(\pi_{z_i,z_j}). \tag{1.10}$$

Here, π_{z_i,z_j} is the connection probability between the inferred community assignments of nodes i and j.

Further, we let $Z_i = \{Z_{i1}, Z_{i2}, \ldots Z_{ik}\}$ be a collection of binary indicators where Z_{ik} is 1 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,

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

Next,

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

Optimizing the parameters of this incomplete data log likelihood requires computing the posterior $P(\mathbf{z} \mid \mathbf{A})$ but unfortunately is intractable, as shown by Daudin *et al.* (38). 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, $P(\mathbf{z} \mid \mathbf{A})$. To optimize the lower bound of $\mathcal{L}(\mathcal{A})$, we seek the $\mathcal{R}_A$ that is as close as possible to $P(\mathbf{z} \mid \mathbf{A})$. In other words, we define the lower bound of $\mathcal{L}(\mathbf{A})$ as $\mathcal{T}(\mathcal{R}_A)$, with,

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

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 (66) as,

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

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

Daudin *et al.* (38) , show that the optimal estimate for τ_{ik} denoted $\hat{\tau}_{ik}$ satisfies

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

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 $\boldsymbol{\pi}$ that maximize $\mathcal{T}(\mathcal{R}_A)$ are also shown by Daudin *et al.,* (38) to be,

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

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.* (161) also show that belief propagation (97) is very effective for this task. 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.

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.

1.3.3.3 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 and that 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 wide degree distribution). 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; 115).

First, Aicher *et al.* developed a model and associated inference technique as the initial efforts toward a weighted stochastic block model. Here, edge weights can be modeled by any exponential family probability 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 an estimate for the number of expected communities, K. However, 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 (115) 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 Markov Chain Monte Carlo (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 even classic unweighted data is often invalid. That is, for some networks, the fitted model may not actually be a good fit, in that samples from the learned model are substantially different from the network. Work by Karrer *et al.* (69), introduced a simple extension to the classic stochastic block model, known as the degree corrected stochastic block model. This model is informed by degree distribution as a proxy for the network structure. In networks where there is a wide degree distribution (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 slightly modify the learned $K \times K$ matrix, $\boldsymbol{\pi}$ slightly. Here, π_{ij} described the number of edges between nodes i and j. Further, these edges counts are modeled as poisson random variables. The likelihood of the observed network under this poisson assumption takes into each node's degree.

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 (11) or be members of overlapping communities (80). Airoldi *et al.*, pioneered the development of the mixed membership stochastic block model (11), 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 the $K \times K$ matrix, $\boldsymbol{\theta}$ representing the between community connection probabilities.

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

Following the development of the mixed membership stochastic block model, Latocuhe *et al.* (80) addressed an important limitation of (11). Since the probability of an edge between a pair of nodes p and q depends on a single draw of $\mathbf{z}_{p\to q}$ and $\mathbf{z}_{q\to p}$, 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 a higher order resolution of structure by considering each node in the context of its neighbors.

1.3.3.4 Affiliation model and inference

We have previously discussed extensions of the stochastic block model that account for the assumption that nodes can belong to multiple communities. Another interesting idea is the idea of *pluralistic homophily*, where the probability that two individuals are connected is related to the affiliations of the nodes (50). In other words, the more groups a pair of nodes share, the more likely they are to have a connection. For example, if two two people were graduate students, studying computational biology at the same university, they are more likely to be connected than a pair of graduate students studying different subjects at the same university. A state-of-the-art method called BIGCLAM was presented for this task by Yang *et al.* in 2013 (153). The objective here is to model the connection probability between a pair of nodes based on the similarity in their learned affiliations towards communities. To do this, individual nodes are connected with communities with some number of links, with more links from a node to a community indicating that the node has a higher 'affilation' to that group. For a network with N nodes and c communities, the affiliation between nodes and communities is encoded by a matrix, $\mathbf{F}$, where F_{uc} is the learned count of links (again encoding the affiliation), between node i and community c. Similarly, let F_u and F_v be the community affiliations for nodes u and v. Then the probability that an edge exists between nodes u and v, or $P(A_{uv} = 1)$ is modeled as,

$$P(A_{uv} = 1) = 1 - \exp(-F_u F_v^T). \tag{1.17}$$

The node to community affiliations can be used as a proxy for the total amount of interaction between a pair of nodes u and v with a Poisson distribution. This modeling paradigm will for the straightforward modeling of the probability that an edge exists between the node pair. To do this, the total amount of interaction between nodes u and v is modeled as,

$$X_{uv} = \sum_c X_{uv}^c, X_{uv}^c \sim \text{Poisson}(F_{uc} \cdot F_{vc}). \tag{1.18}$$

The convenience of this model lies in the fact that we also know how to handle the sum of Poisson random variables is distributed ($X_{uv} \sim \text{Poisson}(\sum_c F_{uc} \cdot F_{vc})$), and corresponds to equation shown in 1.17. From here, it is straightforward to model the probability of u and v sharing connections based on the Poisson probability mass function as,

$$P(X_{uv} > 0) = 1 - P(X_{uv} = 0) = 1 - \exp(-\sum_c F_{uc} \cdot F_{vc}) \tag{1.19}$$

From here the task is then to learn the $\mathbf{F}$ that maximizes the log-likelihood (ll) of the observed network, $\mathbf{A}$, $ll(\mathbf{F} \mid \mathbf{A})$. This can be expressed (in terms of the set of edges, E) as,

$$ll(\mathbf{F}) = \sum_{(u,v)\in E} \log(1 - \exp(-F_u F_v^T)) - \sum_{(u,v)\notin E} F_u F_v^T. \tag{1.20}$$

This optimization problem can be easily solved with a block coordinate gradient ascent algorithm, which updates the F_u for each u, while keeping all other v fixed. Ultimately, after $\mathbf{F}$ has converged, there needs to be a rule to decide which communities u is a member of. To do this, some threshold is chosen, δ such that now u belongs to community c if $F_{uc} > \delta$. BIGCLAM was shown to outperform competing methods, such as the mixed membership stochastic block model on large social networks with ground truth communities.

1.3.4 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 (119) to learn a latent space representation of nodes in some d-dimensional Euclidean space (i.e. an emedding). Once the network is embedded in a lower dimensional space, simple clustering techniques, such as k-means (62) 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 (107; 57). 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 adapted the Word2Vec approach to this context. Word2Vec is a powerful tool from natural language

understanding that allow for the specification of a node embedding that enables accurate prediction of a word's context, given the word (94). 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 predict a set of nodes likely to be seen on a random walk with the node of interest. The embeddings should preserve these learned rules. 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 (59). 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 (74). 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 (87). For example, a community under structural equivalence might group could group nodes with similar degree or centrality. 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 in 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 random walk 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 hubs, high degree nodes, or low degree nodes).

1.3.5 Higher order network analysis

One of the most recent advances in community detection is clustering nodes based on 'higher order' features, or at the level of small subgraphs or *motifs*. The structural organization of the network is then interrogated by identifying clusters of network motifs. The flexibility and appeal of this framework is that different kinds of organizational patterns of the network can be identified, depending on the motif used. Seminal work using this approach was proposed by Benson *et al.* (**?**) . Given a motif, M,

higher order clustering frameworks seek to identify a cluster of nodes S that minimize the following ratio,

$$\phi_M(S) = \text{cut}_M(S, \bar{S}) / \min(\text{vol}_M(S), \text{vol}_M(\bar{S})), \tag{1.21}$$

where $\bar{S}$ denoted the set of nodes not in S, $\text{cut}_M(S, \bar{S})$ is the number of instances of motif M with a least one node in S and one node in $\bar{S}$. Finally, $\text{vol}_M(S)$ is the number of nodes in instances of M that belong to S. The optimization framework to identify near-optimal clusters is an extension to standard spectral clustering methods and it outlined as follows.

1. For a motif of interest, M, define a motif adjacency matrix, W_M where the ijth entry is the the number of instances of M that contain nodes i and j

2. Compute the spectral ordering (i.e. order the eigenvalues in descending order), δ, of the nodes from the normalized motif Laplacian of W_M. Note this motif Laplacian is the standard Laplacian matrix for W_M (92).

3. Identify the set of δ with the smallest motif conductance, or, $S := \arg\min_r \phi_M(S_r)$, where $S_r = \{\delta_1, \ldots, \delta_r\}$.

The novelty in this approach stems from the fact that it can be applied to a variety of network types that not all methods can handle. In particular, Benson *et al.*, highlight how this approach can be used to deal with directed, undirected, weighted, unweighted, and signed networks (21), as well as the flexibility to uncover diverse types of structural organization.

1.4 Community detection in computational biology

Analyzing networks with community detection has shown to be fruitful, particularly in biology and neuroscience applications. In this section, we will describe examples of analyses where the identification of communities provided insight and understanding for a biological 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 biology and medicine (39; 12; 85). In this section, we outline particular examples of how the application of community detection to diverse types of biological data lead to improved scientific 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 (19) was used to quantify various features of the immune system, such as, cell type abundances and signaling activity. A correlation network between the measured set of immune features in the training cohort was constructed to develop hypotheses about their interactions. In addition to the construction of the network, an elastic net regression model (164) was trained to identify immune features associated with increased gestational age. When communities were identified in the correlation network of the 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 a particular time in 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.

The use of community detection in this analysis helped to understand which immune features and combinations of immune features are predictive of increased gestational age. The implications of such an observation is an increased ability to predict when a pregnancy is diverting from its normal, healthy progression.

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 (130). 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 (81). A network in this context is typically constructed based on some notion of co-occurence 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-occurence 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 (56). After identifying a community structure for each network, the similarity of these partitions was quantified with the Rand index (140), 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 in each community were interrogated. Some interesting co-occurence 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 (145). Nodes have the potential to be *connecters*, 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-occurence patterns for further investigation.

1.4.3 Community detection for analysis of flow cytometry data

Flow cytometry allows for the 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 (117). 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.* (159). In this approach, the authors seek to segment a population of cells into distinct subpopulations, through the construction of pairwise single cell similarity network. After constructing this network, communities detection can be applied to cluster the cells into sub populations. To recap, in this network, the nodes are comprised of the individual cells within a sample, and edges between nodes indicate the similarity between a pair of cells, based on the quantification of their emitted light. Another useful feature of SamSPECRAL is that it also does some preprocessing to reduce the size of the constructed network.

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, the unregistered nodes are attempted to be registered or agglomerated with the set of registered nodes. For example, for one of the registered nodes, which can be denoted 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 quantified through flow cytometry) between a pair of a communities. Once this weighted graph is created, a spectral community detection method (150) 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. Some examples of challenging shapes are, overlapping clusters, non-elliptical shaped clusters, or low-density clusters. Ultimately the retrieved population of cells avoided manual segmentation and provided a solution to a computationally challenging task.

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, (79) 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 (70).

After identifying communities within each HVR network, the authors used two summary statistics to formulate their biological hypothesis. First, the variation of information (126) 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* (100) 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, (79) 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 heterogenity

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 (84). 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 for high dimensional profiling of single cells. 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 (90). 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 ($T < L$), 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 (138).

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 at 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 n ew way to integrate genomic and individual patient information to determine which types of antibiotics might be most effective.

1.5 Thesis Contribution

In the previous section, we presented several case studies for how community detection enables and simplifies biological understanding. To recap, a network representation data and a community detection analysis can help to uncover structural organizational patterns and important co-occurence relationships in applications, such as, immunology and microbiome analysis. Further, community detection enables clustering, even if the task seems computationally challenging, or has complicated geometry in high dimensions. While previous work in community detection is well-developed for standard networks modeling a single type of relational definition between nodes, we find it necessary to adapt these approaches to more diverse types of network data.

1.5.1 Thesis Statement

In this thesis, we address three challenging types of network data, where the identification of communities is challenging. Among these classes of networks, we have developed four new methods that adapt standard community detection to these situations. **1) We focus on how to identify communities in multilayer networks through an extension of the stochastic block model.** Our method learns a collection of models to describe the entire multilayer network. **2) We provide a method to pre-process a large network into a smaller network of *super nodes* that can be used as input to a community detection algorithm.** This pre-processing step decreases the run time of community detection algorithms and decreases the variability across multiple runs of the community detection algorithm. **3) We extend the stochastic block model to handle attributed**

network data, which allows the inference of node-to-community assignments to handle both connectivity and continuous attribute information. Our learned model for the connectivity and attributes can be used to perform link prediction and collaborative filtering. **4) We develop a test to generate an empirical p-value to quantify the extent to which attributes and connectivity align.** Our approach is based on label propagation and we use synthetic data and a single cell mass cytometry dataset to validate the empirical p-value generated by our test.

1.5.2 Summary of the novelty of this work

To succinctly summarize the contributions of this thesis, we outline each of the 3 developed methods, their top 3 pieces of related work, and why our approach makes a novel contribution in table 1.1.

Method	Brief Description	3 Similar Approaches	Novelty
sMLSBM (Chapter 2)	Stochastic block model for multilayer networks	MLSBM through aggregation methods (144), Restricted MLSBM (109), MLSBM (61)	Instead of fitting a consensus SBM to all layers, we learn a set of models that represents different clusters of layers.
SuperNode (Chapter 3)	Pre-processing a network into a smaller network of super nodes before applying community detection	Identify communities on core of network and propagate labels outward (116), Create super nodes with prior information about nodes expected to be together (156), Reduce the size of the network by systematically removing nodes and edges (55)	We recast the entire network as a network of super nodes and input this smaller version into community detection algorithms. We do not require prior knowledge or side information
Attribute SBM (Chapter 4)	Incorporate both network connectivity and continuous attributes into account when assigning nodes to communities	SBM inference with a single attribute (103), iLouvain Modifying modularity to incorporate attributes (34), CESNA: Affiliation model with attributes (155)	We adapt the SBM to handle multiple continuous attributes
Attribute Alignment Test (Chapter 5)	A test to quantify the extent to which attributes and connectivity align.	neoSBM + BESTest (attribute augmented SBM) (111). To our knowledge, there are not any related tests other than (111)	We use label propagation to generate an empirical p-value that quantifies how the attributes relate to connectivity. We do not require assuming that a stochastic block model is an appropriate representation of the network.

Table 1.1: **Summarizing the novelty of our 3 developed methods**. For each of the 3 methods we developed, we provide a brief description of what it does, the top 3 most similar approaches, and why our approach is novel.

1.5.3 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.

1.5.4 Software

The four 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

4. **AttributeAlign: For testing alignment between attributes and connectivity**. https://github.com/stanleyn/AttributeAlign

CHAPTER 2

Strata Multilayer Stochastic Block Model

This work is done in collaboration with Saray Shai, Dane Taylor and Peter Mucha.

Multilayer networks are a useful data structure for simultaneously capturing multiple types of relationships between a set of nodes. In such networks, each relational definition gives rise to a layer. While each layer provides its own set of information, community structure across layers can be collectively utilized to discover and quantify underlying relational patterns between nodes. To concisely extract information from a multilayer network, we propose to identify and combine sets of layers with meaningful similarities in community structure. In this paper, we describe the "strata multilayer stochastic block model" (sMLSBM), a probabilistic model for multilayer community structure. The central extension of the model is that there exist groups of layers, called "strata", which are defined such that all layers in a given stratum have community structure described by a common stochastic block model (SBM). That is, layers in a stratum exhibit similar node-to-community assignments and SBM probability parameters. Fitting the sMLSBM to a multilayer network provides a joint clustering that yields node-to-community and layer-to-stratum assignments, which cooperatively aid one another during inference. We describe an algorithm for separating layers into their appropriate strata and an inference technique for estimating the SBM parameters for each stratum. We demonstrate our method using synthetic networks and a multilayer network inferred from data collected in the Human Microbiome Project.

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-occurence 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 (163), or the desire to study multiple microbial co-occurence networks in different sites of the body (142). Multilayer networks provide a framework to do this, in that each relational definition leads to a new layer in the network (72; 24; 42). 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 (35), in understanding underlying relationships and community structure across social networks (58), and in the analysis of temporal networks (96). Furthermore, recent advances in the mathematical foundations for multilayer networks have made analysis of these types of data more feasible. In particular, (42) 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 (96). 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 (41) 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 (89) or co-clustering (47) 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 (47), 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 (143; 141); multiscale properties such as community structure (108; 106; 65), the spectra of network-related matrices (28) and by defining latent roles (27). 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 (120; 51). Although there are many definitions for what constitutes a "community" (125), 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 (102) and fitting a generative probabilistic model (67). Because each of these approaches present computational challenges for efficiently detecting communities, numerous heuristics exist for developing practical algorithms (121; 51; 82; 33; 105).

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 (106), 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 (65), 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) (131), 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. (125; 9)). 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 (67; 60). 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, $\boldsymbol{\pi}$ and $\mathbf{Z}$. Parameter $\boldsymbol{\pi}$ 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 $\boldsymbol{z}$, 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 (96) and studying dynamics (40) 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 (61; 109; 15; 144; 114). In this context, the general assump-

tion 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. (61; 109; 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. (61) 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 $L \to \infty$. 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 $\boldsymbol{\pi}^l$ 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 (38) to the multilayer setting.

Ref. (61) was followed up by Ref. (109), 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^l_{mn}, for each pair, (m, n), each entry in $\boldsymbol{\pi}$ 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 $\text{logit}(\pi^l_{mn}) = \pi_{mn} + \beta_l$. 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. (61) and (109) 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. (144) and (114), which provide techniques to determine whether a single layer network is the result of an aggregation procedure in a multilayer network. In Ref. (144), 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. (114) 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 (41), 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

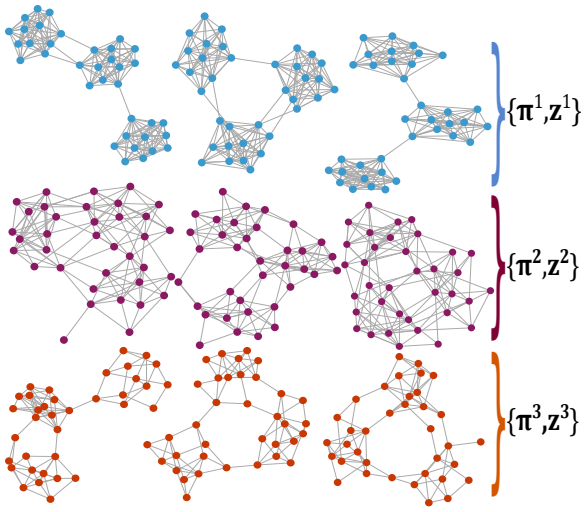

Figure 2.1: **Objective of strata multilayer stochastic block model (sMLSBM)**. Each of the $L = 9$ networks here represents a layer in a multilayer network. Every network layer has $N = 36$ nodes that are consistent across all layers. There are $S = 3$ strata as indicated by the three rows and the colors 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 strata 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.

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.

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 $s \in \{1, 2, \cdots, S\}$, are parameterized by $\boldsymbol{\pi}^s$ and $\mathbf{Z}^s$ (or equivalently, vector $\boldsymbol{z}^s$, which has entries $z_i^s = \text{argmax}_m\{Z_{im}^s\}$). Note that the parameters $\boldsymbol{\pi}^s$

and $\mathbf{Z}^s$ 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 $s, t \in \{1, \ldots, S\}$, $s \neq t$. 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, $\mathbf{\Pi} = \{\boldsymbol{\pi}^1, \boldsymbol{\pi}^2, \ldots, \boldsymbol{\pi}^S\}$ and $\mathcal{Z} = \{\mathbf{Z}^1, \mathbf{Z}^2, \ldots \mathbf{Z}^S\}$. 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 (41). 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^s_{ij} gives the probability of an edge between nodes i and j in stratum s. That is, $\theta^s_{ij} = \pi^s_{z^s_i z^s_j}$, where z^s_i 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 $\boldsymbol{y}$, 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,

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

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, $p(A^l_{ij} = 1) \sim \text{Bernoulli}(\pi^s_{z_i z_j})$.

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 $l \in \{1, \ldots, L\}$, which yields inferred SBM parameters $\hat{\boldsymbol{\pi}}^l$ and node-to-community memberships $\hat{\mathbf{Z}}^l$. Then we cluster the layers based on the similarities of $\hat{\boldsymbol{\pi}}^l$ and $\hat{\mathbf{Z}}^l$. To infer $\hat{\boldsymbol{\pi}}^l$ and $\hat{\mathbf{Z}}^l$, we use the the inference method described in Ref. (38). 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 $\hat{\mathbf{\Pi}} = \{\hat{\boldsymbol{\pi}}^1, \hat{\boldsymbol{\pi}}^2, \ldots, \hat{\boldsymbol{\pi}}^L\}$ and $\hat{\mathcal{Z}} = \{\hat{\mathbf{Z}}^1, \hat{\mathbf{Z}}^2, \ldots \hat{\mathbf{Z}}^L\}$ (that is, at this stage of the procedure, each layer is temporarily treated as its own stratum). Note also that each $\hat{\boldsymbol{Z}}^l$ can be equivalently represented by vector $\hat{\boldsymbol{z}}^l$. Using the estimates $\hat{\boldsymbol{\pi}}^l$ and $\hat{\mathbf{Z}}^l$ for a given layer, l, we can construct the corresponding adjacency probability matrix, $\hat{\boldsymbol{\theta}}^l$, which is defined entry-wise by $\hat{\boldsymbol{\theta}}^l_{ij} = \hat{\pi}^l_{\hat{z}_i, \hat{z}_j}$. Doing this for each layer results in a collection of adjacency probability matrices, $\hat{\mathbf{\Theta}} = \{\hat{\boldsymbol{\theta}}^1, \hat{\boldsymbol{\theta}}^2, \cdots, \hat{\boldsymbol{\theta}}^L\}$.

Now, we seek an initial partition of layers into strata based on $\hat{\mathbf{\Theta}}$. The goal is to identify S sets $\mathcal{L}^s$ so that the matrices $\{\hat{\boldsymbol{\theta}}^l\}$ with $l \in \mathcal{L}^s$ are close to one another, but they are distant from the remaining matrices, $\{\hat{\boldsymbol{\theta}}^l\}$ with $l \in \mathcal{L} \setminus \mathcal{L}^s$. This is accomplished by treating each $\hat{\boldsymbol{\theta}}^l$ as a feature

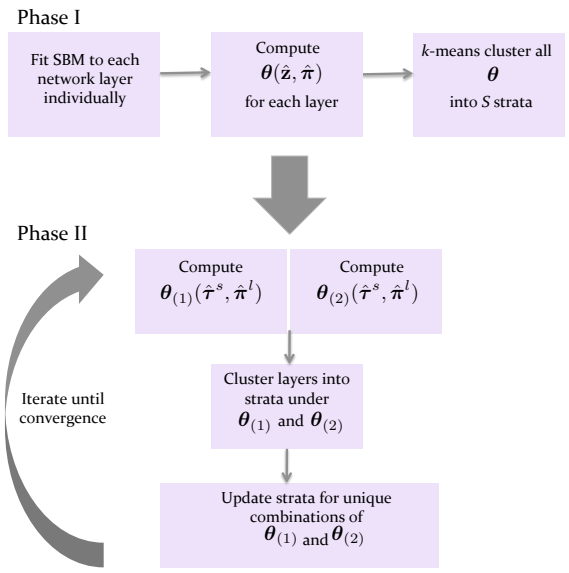

Figure 2.2: **Schematic illustration of our algorithm**: Our algorithm for fitting an sMLSBM is broken up into two phases: an initialization phase to cluster layers into strata, and an iterative phase that allows learning of node-to-community and layer-to-strata assignments.

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 (137). 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 $\boldsymbol{\pi}^s$ and the $N \times K^s$ matrix $\mathbf{Z}^\mathbf{s}$ 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.

We now proceed to maximize the likelihood in each stratum, by extending the framework of Ref. (38) to a multilayer context. Note that this is similar to Ref. (61), 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,

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

where

$$p(\mathcal{A}^s \mid \mathbf{Z}^s) = \prod_{l \in \mathcal{L}^s} \prod_{i<j} \prod_{mn} {\pi^s_{mn}}^{A^l_{ij}} (1 - \pi^s_{mn})^{(1-A^l_{ij})}. \tag{2.3}$$

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

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

$$p(\mathbf{Z}^s) = \prod_i \prod_m {\alpha^s_m}^{(Z^s_{im})}. \tag{2.4}$$

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

$$\begin{aligned} \log P(\mathcal{A}^s, \mathbf{Z}^s) &= \log(P(\mathbf{Z}^s)) + \log(P(\mathcal{A}^s \mid \mathbf{Z}^s)) \\ &= \sum_i \sum_m Z^s_{im} \log(\alpha^s_m) \\ &+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} A^l_{ij} \log(\pi^s_{mn}) \\ &+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} (1 - A^l_{ij}) \log(1 - \pi^s_{mn}). \end{aligned} \tag{2.5}$$

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 (45). Doing so requires the ability to compute $P(\mathbf{Z}^s \mid \mathcal{A}^s)$; however, Ref. (38) 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 (61; 15; 38). In general, a variational approximation seeks to optimize a lower bound on the log-likelihood. To do this, we first approximate the conditional distribution, $P(\mathbf{Z}^s \mid \mathcal{A}^s) \approx R_{\mathcal{A}^s}$, where

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

Here, matrix $\boldsymbol{\tau}^s$ contains entries τ^s_{im} that approximate the probability that node i belongs to community m in stratum s. Further, function $h(\cdot)$ represents the multinomial distribution, with parameters,

$\{\boldsymbol{\tau}^s_{im}\}$ for $m \in \{1, \ldots, K^s\}$. Using this, we define the variational approximation as

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

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, $P(\mathbf{Z}^s \mid \mathcal{A}^s)$, 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

$$\begin{aligned}
\mathcal{J}(R_{\mathcal{A}^s}) = &\sum_i \sum_m \tau^s_{im} \log(\alpha^s_m) \\
&+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} \tau^s_{im} \tau^s_{jn} [A^l_{ij} \log(\pi^s_{mn})] \\
&+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} \tau^s_{im} \tau^s_{jn} [(1 - A^l_{ij}) \log(1 - \pi^s_{mn})] \\
&- \sum_i \sum_m \tau^s_{im} \log(\tau^s_{im}).
\end{aligned} \tag{2.8}$$

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:

$$\hat{\alpha}^s_m = \sum_i \hat{\tau}^s_{im} / N \,, \tag{2.9}$$

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

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

To find the best estimates for $\hat{\boldsymbol{\tau}}^s$ and $\hat{\boldsymbol{\pi}}^s$, we alternate between updating $\hat{\boldsymbol{\tau}}^s$ and $\hat{\boldsymbol{\pi}}^s$ until convergence. When convergence has occurred, we refer to the resulting estimates as the consensus $\overline{\boldsymbol{\tau}^s}$ and $\overline{\boldsymbol{\pi}^s}$ for stratum s. Similarly, $\overline{\boldsymbol{Z}^s}$ represents the consensus indicator matrix of node-to-community assignments computed from $\overline{\boldsymbol{\tau}^s}$. 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 $\overline{\boldsymbol{\tau}^s}$ and $\overline{\boldsymbol{\pi}^s}$ 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 $\overline{\boldsymbol{\tau}^s}$, we compute the maximum-likelihood SBM parameters for a particular layer l, which we denote with a tilde and hence, $\tilde{\boldsymbol{\pi}}^l$ and $\tilde{\boldsymbol{\tau}}^l$. Similarly, for fixed $\overline{\boldsymbol{\pi}^s}$, we compute the node-to-community assignments $\tilde{\boldsymbol{\tau}}^l$. 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 $\boldsymbol{\theta}(\boldsymbol{\tau}, \boldsymbol{\pi})$ represent the adjacency probability matrix specified by $\boldsymbol{\tau}$ and $\boldsymbol{\pi}$, we define

$$\boldsymbol{\theta}^l_{(1)} = \boldsymbol{\theta}^l(\overline{\boldsymbol{\tau}^s}, \tilde{\boldsymbol{\pi}}^l), \tag{2.12}$$

$$\boldsymbol{\theta}^l_{(2)} = \boldsymbol{\theta}^l(\tilde{\boldsymbol{\tau}}^l, \overline{\boldsymbol{\pi}^s}) \tag{2.13}$$

Note that the first definition uses the strata-consensus estimate for $\boldsymbol{\tau}^s$ and a layer-specific estimate for $\boldsymbol{\pi}^s$, whereas the latter uses a layer-specific estimate for $\boldsymbol{\tau}^s$ and the strata-consensus estimate for $\boldsymbol{\pi}^s$.

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, $\boldsymbol{\theta}^l_{(1)}$ and $\boldsymbol{\theta}^l_{(2)}$ 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 $\boldsymbol{\theta}^l_{(1)}$ and $\boldsymbol{\theta}^l_{(2)}$, 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 $\boldsymbol{\theta}^l_{(1)}$ and $\boldsymbol{\theta}^l_{(2)}$ 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 $\boldsymbol{\theta}^l_{(1)}$ and stratum

2 according to $\boldsymbol{\theta}^l_{(2)}$, 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 $\boldsymbol{\theta}^l_{(1)}$ and $\boldsymbol{\theta}^l_{(2)}$ (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 $\boldsymbol{y}$) as input, the algorithm yields (ideally) improved layer-to-strata assignments (as well as consensus estimates for the SBM parameters of the strata, $\overline{\boldsymbol{\tau}^s}$ and $\overline{\boldsymbol{\pi}^s}$). 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

In this section, 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 $\boldsymbol{\pi}^s$ for each stratum s in terms of two parameters, p^s_{in} and p^s_{out}, which give the within-community edge probabilities and between-community edge probabilities, respectively. That is, we define $\boldsymbol{\pi}^s_{mn} = p^s_{in}$ when $m = n$ and $\boldsymbol{\pi}^s_{mn} = p^s_{out}$ when $m \neq n$. It follows that

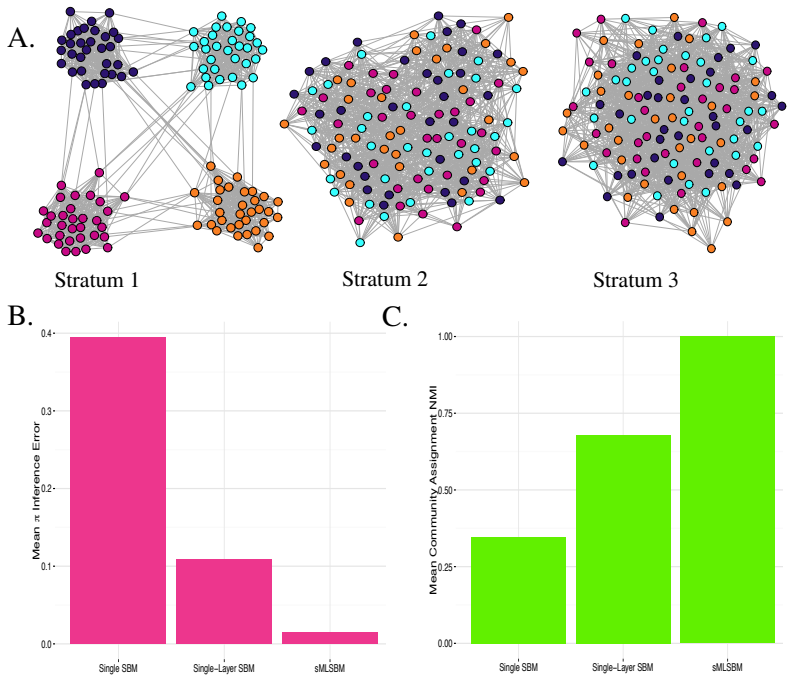

Figure 2.3: **Synthetic experiment comparing sMLSBM to other SBMs. A**. We specified a model with $S = 3$ strata and $L = 10$ layers per stratum. A representative layer from each stratum is plotted. Note that nodes in all networks are colored according to their community membership in stratum 1. Each network has $N = 128$ nodes, $K = 4$ communities and mean degree, $c = 20$. The p^s_{in} parameters for $s = 1, 2$ and 3 are 0.6, 0.4 and 0.25, respectively. Corresponding values of p^s_{out} 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 $\overline{\boldsymbol{\pi}^{s_l}}$ for the true SBM of each layer l, which is denoted $\boldsymbol{\pi}^l$. Here s_l denotes the inferred strata for layer l. On the vertical axis we plot the mean $\ell 2$ norm error $||\text{vec}(\boldsymbol{\pi}^l) - \text{vec}(\overline{\boldsymbol{\pi}^{s_l}})||_2$. **C**. For each of the three models, we computed the normalized mutual information (NMI) between the true node-to-community assignments $\mathbf{z}^l$ and the inferred values $\overline{\mathbf{z}^{s_l}}$.

the expected mean degree is given by $c = N(p^s_{in} + (K-1)p^s_{out})/K$. In our experiment, we select the following SBM parameters: $(p^1_{in}, p^1_{out}) = (0.6, 0.0083)$; $(p^2_{in}, p^2_{out}) = (0.4, 0.075)$; and $(p^3_{in}, p^3_{out}) = (0.125, 0.167)$. 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.

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, $\overline{\boldsymbol{\pi}^{y_l}}$, and community

assignments, $\overline{\mathbf{Z}^{s_l}}$. First, for each layer and each model, we quantified the error (ℓ^2 norm) between vec($\overline{\boldsymbol{\pi}^{y_l}}$) and its true value, vec($\boldsymbol{\pi}^l$). Note that 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 $||\text{vec}(\boldsymbol{\pi}^l) - \text{vec}(\overline{\boldsymbol{\pi}^{s_l}})||_2$. 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) (36) between the true node-to-community assignments, $\mathbf{z}^l$, and the inferred values, $\overline{\mathbf{z}^{y_l}}$, under each model. In other words, for each layer, we compute, NMI($\mathbf{z}^l, \overline{\mathbf{z}^{y_l}}$). 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^s_{in} and p^s_{out}. Since we have already specified the expected mean degree, these parameters must satisfy the constraint $c = N(p^s_{in} + p^s_{out})/2$ for both strata. In all simulations, we fixed the SBM parameters of the first strata as $(p^1_{in}, p^1_{out}) = (.1836, .1055)$. It is also convenient to define the quantity, $N(p^1_{in} - p^1_{out}) = 10$, which relates to the detectability of communities (43). For example, the ability to detect community structure in a given layer and/or

strata is, in general, expected to improve with increasing $N(p^s_{in} - p^s_{out})$. For the second strata, we allow $N(p^2_{in} - p^2_{out})$ 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 $N(p^2_{in} - p^2_{out}) = 10$, which represents the point where the two strata are indistinguishable since $(p^1_{in}, p^1_{out}) = (p^2_{in}, p^2_{out})$. In Fig. 4(A), we show the NMI between the true layer-to-strata assignments and those inferred by sMLSBM, or NMI$(\mathbf{y}, \hat{\mathbf{y}})$. As a baseline, we compare sMSLBM 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.

2.8 Human Microbiome Project Example

As an application of sMLSBM, we consider correlation networks constructed from data from the Human Microbiome Project (142). For various sites on the body, the human microbiome project has

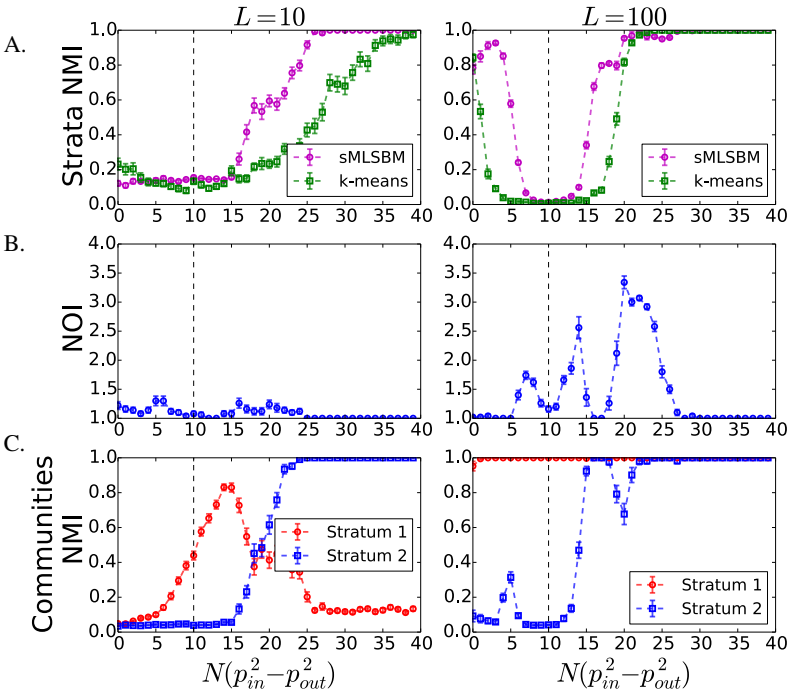

Figure 2.4: **Synthetic experiment with two strata.** We conducted numerical experiments with multilayer networks with $N = 128$ nodes, mean degree $c = 16$, $S = 2$ strata and $K^1 = K^2 = 4$ communities. The networks contained either $L = 10$ (left column) or $L = 100$ layers (right column), which 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).

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 (49). 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 (53) 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. (53), 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 (53)). 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 (41). 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 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. (48) offers a state-of-the-art clustering of body sites based on biological expertise. Here, the authors

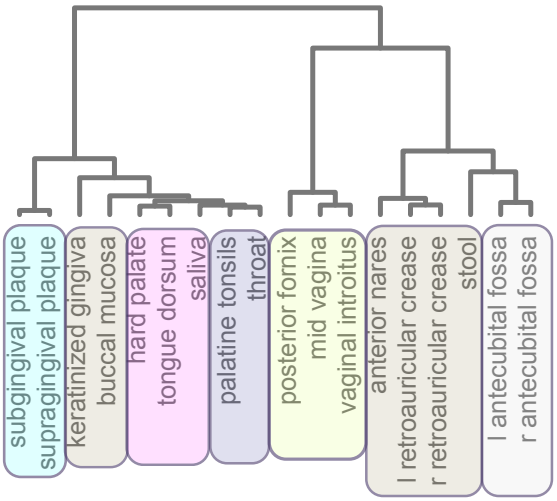

Figure 2.5: **Comparison of sMLSBM on the OTU interaction networks (53) for each of the body sites to a reducibility hierarchy (41).** As described in the text, we consider a multiplex network with $L = 18$ layers and $N = 213$ nodes, which we group here into $S = 6$ strata, while the dendrogram was 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.

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 (48) 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, $\overline{\boldsymbol{\pi}}^s$ and $\overline{\mathbf{Z}}^s$. 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 (41). We stress that although reducibility is a closely related pursuit, it is fundamentally different from our co-clustering pursuit of simultaneously identifying 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,

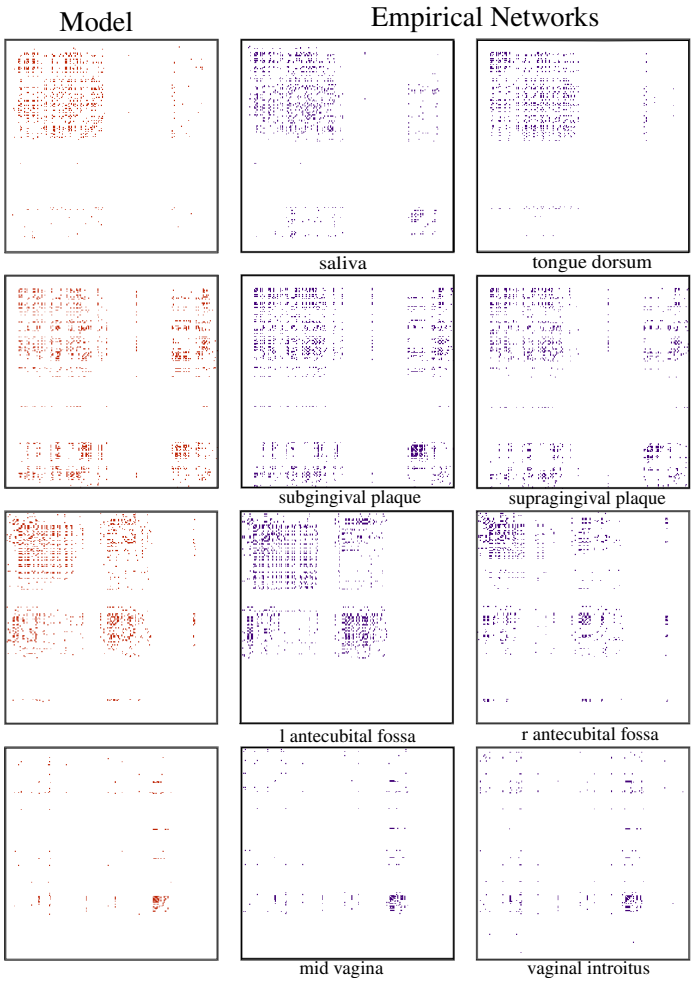

Figure 2.6: **Visualization of Strata in SparCC Networks.** We visualize the adjacency matrices for SparCC networks that encode microbiome interactions at body sites. In each panel, a colored dot at position (i, j) indicates the existence of an edge (i, j) in the corresponding network layer. The four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

there are a variety of ways to aggregate layers in a strata. See, for example, Ref. (136), 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. (41) (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 (41) 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 (10) and (148), respectively. It would also be useful to extend to degree-corrected and overlapping (i.e., mixed-membership) communities (69), 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 (158). 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. In work lead by Dane Taylor and collaborators Saray Shai, and Peter Mucha, we investigate these questions in *Enhanced detectability of community structure in multilayer networks through layer*

aggregation (136). 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 (77; 124; 64; 44; 98; 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 (122), hierarchical structure (112; 127), and in temporal networks (54), 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 $N \to \infty$, there is a solution to the detectability limit (44; 98), characterized by the solution curve (Δ^*, ρ) to

$$N\Delta = \sqrt{4N\rho}, \tag{2.14}$$

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 (44) and random matrix theory (98).

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 $\tilde{L} = \{1, \ldots, L\}$ 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 (18; 99) is a useful approach for studying partitioning and phase transitions in detectability (i.e. node-to-community assignment accuracy) (98; 112; 127). 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 $N \to \infty$ limiting distribution is given by a spectral density $P(\lambda)$. The theory we develop in this work is based on the modularity matrix, $\bar{B_{ij}} = \bar{A}_{ij} - \rho L$ (104).

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

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

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

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

The largest eigenvalue of $\bar{\mathbf{B}}$ in the $N \to \infty$ limit is the isolated eigenvalue,

$$\lambda_1 = NL\Delta/2 + 2[\rho(1-\rho) - \Delta^2/4]/\Delta. \tag{2.17}$$

The eigenvector $\mathbf{v}$ corresponding to λ_1 gives the spectral bipartition. Here, the inferred community label of node i is determined by the sign of v_i and provided that the largest eigenvalue corresponds to this isolated eigenvalue, λ_1, the eigenvector entries $\{v_1\}$ are correlated with the node-to-community labels, $\mathbf{z}$. As shown in (136), we can derive a detectability equation that accounts for the number of layers,

$$NL\Delta = \sqrt{4NL\rho(1-\rho)}. \tag{2.18}$$

We now study Δ^* for the thresholded networks, which correspond to single-layer SBMs in which the community labels, $\mathbf{z}$ are identical to those of the multilayer SBM, but are new effective block edge probabilities

$$\hat{\Pi}_{nm}^{(\tilde{L})} = 1 - F(\tilde{L} - 1; L, \Pi_{nm}), \tag{2.19}$$

where $F(a; L, p)$ is the cumulative distribution function for the binomial distribution $f(a; L, p)$. The effective probabilities for the AND and OR networks are $\hat{\Pi}_{nm}^{(L)} = (\Pi_{nm})^L$ and $\hat{\Pi}_{nm}^{(1)} = 1 -$

$(1-\Pi_{nm})^L$, respectively. For the two-community SBM, the effective probabilities are $\hat{p}_{\text{in,out}}^{(\tilde{L})} = 1 - F(\tilde{L}-1; L, p_{\text{in,out}})$, $\hat{\Delta}^{(\tilde{L})} = \hat{p_{in}}^{(\tilde{L})} - \hat{p_{in}}^{(\tilde{L})}$ and $\hat{\rho}^{(\tilde{L})} = \hat{p_{in}}^{(\tilde{L})} - \hat{p_{in}}^{(\tilde{L})}/2$.

2.10.4 Results

In Figures 2.7 (a) and (b), we show Δ^* versus the mean edge probability, ρ for the different aggregation methods: (i) a single layer (red dot-dashed curves), which is identical in panels (a) and (b); (ii) the summation network (blue dashed curves), for which the curve in (b) corresponds to the curve in panel (a) rescaled by a factor of 1/2; and (iii) thresholded networks (solid curves), which shift left-to-right with increasing $\tilde{L}$. This is evident by comparing Δ^* for the AND ($\tilde{L} = L$, gold circles) and OR $\tilde{L} = L$, cyan squares) networks. We find that when ρ is large that the AND (OR) network has a relatively large (small) detectability limit. In other words, aggregating layers using the AND (OR) operation is beneficial for dense (sparse) networks. Note that the results shown in Figure 2.7 are just a subset of results from Taylor *et al.,* (136).

2.10.5 Conclusion

In this work, we studied limitations on community detection for multilayer networks with layers drawn from a common SBM. As an illustrative model, we analyzed the effect of layer aggregation on the detectability limit Δ^* for two equal-sized communities. When layers are aggregated by summation, we analytically showed that Δ^* vanishes as $O(L^{-1/2})$. When layers are aggregated by thresholding this summation, Δ^* depends on the choice of threshold, $\tilde{L}$. For $\tilde{L} = \lceil \rho L \rceil$, we analytically found Δ^* to also vanish as $O(L^{-1/2})$. We note that our analysis also describes layer aggregation by taking the mean, $L^{-1}\sum_l \mathbf{A}^{(l)}$, since the multiplication of a matrix by a constant simply scales the eigenvalues by that constant. Thus, our results are in agreement with previous work that proved the consistency of spectral clustering via the mean adjacency matrix (61).

Finally, it is commonplace to threshold pairwise-interaction data to construct network representations that are sparse and unweighted and can be studied at a lower computational cost. Our research provides insight into this common-yet not well understood-practice. It would be interesting to extend this work to allow the SBMs of layers to be correlated (5) (that is, rather than identical) or organized

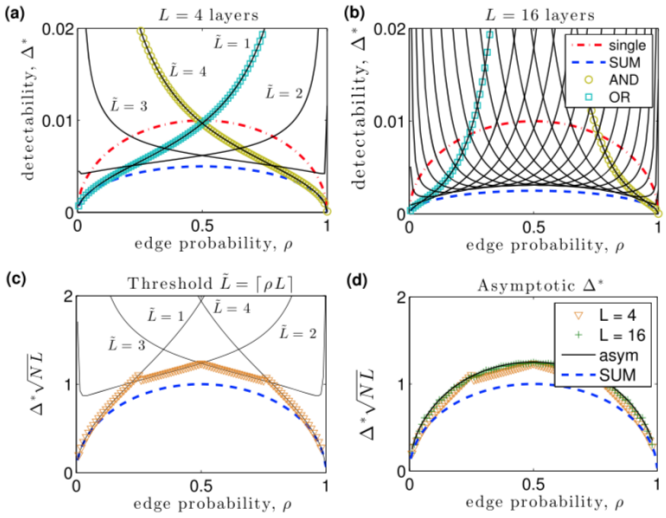

Figure 2.7: **Effects of layer aggregation on detectability.** Layer aggregation enhances the detectability of community structure. (a),(b). We plot the detectability limit Δ^* versus mean edge probability ρ for a single network layer (red dot-dashed curves), the aggregate network obtained by summation (blue dashed curves), and aggregate networks obtained by thresholding this summation at $\tilde{L} \in \{1, 2, 3, 4\}$ (solid curves). Gold circles and cyan squares highlight $\tilde{L} = L$ and $\tilde{L} = 1$, which we refer to as AND and OR networks, respectively. Results are shown for $N = 10^4$ nodes with (a) $L = 4$ and (b) $L = 16$ layers. (c) For $L = 4$, we show Δ^* versus ρ for the optimal threshold $\tilde{L} = \lceil \rho L \rceil$ (orange triangles), which lies on the solution curves for $\tilde{L} \in \{1, \ldots, L\}$ (solid curves). (d) We show Δ^* for $\tilde{L} = \lceil \rho L \rceil$ with $L \in \{4, 16\}$. These piecewise-continuous solutions collapse onto the asymptotic solution δ^*_{asym} (black curve) as L increases. In panels (c), (d), we additionally plot δ^* for the summation network (blue dashed curves).

into ‘strata’ (135) (i.e., layers within a single stratum are similar, but they differ across strata). We are currently extending our analysis to hierarchical stochastic block models.

CHAPTER 3

Network compression for community detection with super nodes

This work is done in collaboration with Roland Kwitt, Marc Niethammer, and Peter Mucha.

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 (`https://snap.stanford.edu/data/`) 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 Nodes* (134).

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 (29). 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

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.

boundaries, maintaining or improving segmentation accuracy (6). The SLIC super pixel method (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,

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

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

We let $\mathbf{z}^{\text{Full}}$ and $\mathbf{z}^{\text{SN}}$ denote the node-to-community assignments for the full network and super node network representations, respectively. To compute the similarity between $\mathbf{z}^{\text{Full}}$ and $\mathbf{z}^{\text{SN}}$, we use Normalized Mutual Information (NMI) (37). That is, for partitions $\mathbf{z}^{\text{Full}}$ and $\mathbf{z}^{\text{SN}}$ 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 $\mathbf{z}^{\text{Full}}$ and community j in $\mathbf{z}^{\text{SN}}$, the NMI between the two partitions is

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

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' (154). 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,(86; 156; 55; 116) with Yang *at al.,* (156) and Peng *et al.* (116), 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.* (156), and Lisewski *et al.* (86); 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.*(86), 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.* (156), 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.* (55), and Peng *et al.* (116). Gilbert *et al.*, introduce the 'KeepAll' method, (55) 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.,* (116) 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 (23) and fitting SBMs (113) 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.

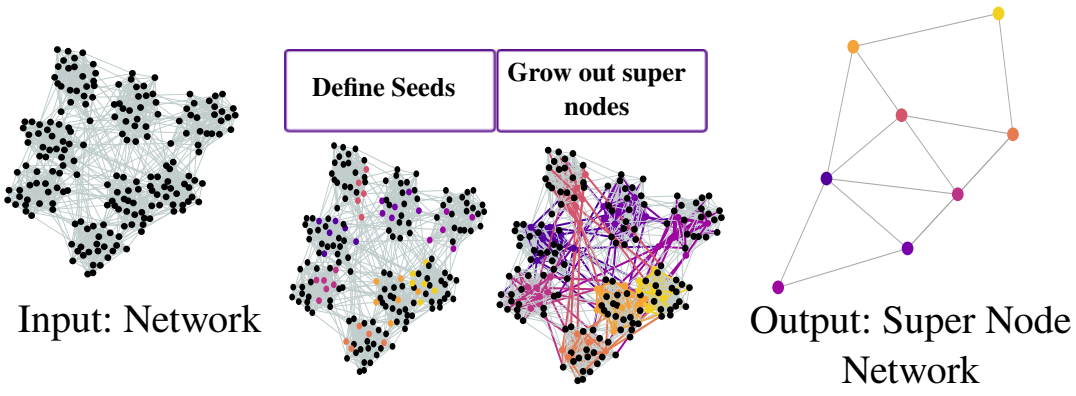

Figure 3.2: **Defining super nodes**. To define the super node representation of a network, we select S seeds and agglomerate local regions around them to create super nodes. This then leads to a new network with weighted edges between the S super nodes upon which community detection can be more efficiently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

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.

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. 113. 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 (68; 71). 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.

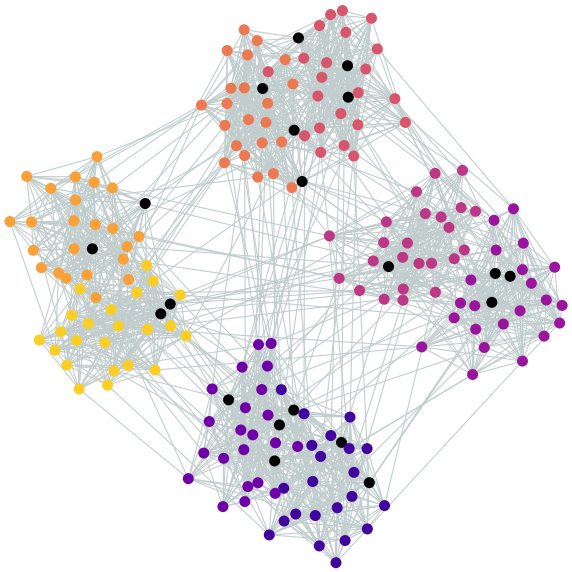

Figure 3.3: **Choosing seeds in a synthetic network**. The identification of 20 seeds with the CoreHD algorithm in a network generated from a stochastic block model with 8 communities. Seeds (black nodes) are well distributed across communities.

At slightly higher computational cost, we employ the CoreHD algorithm, which nearly optimally identifies nodes in network decycling and dismantling (160). 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, 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 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.

.

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.

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 (23) and the stochastic block model (SBM) inference(tia) described in Ref. 113. 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 tat 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 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

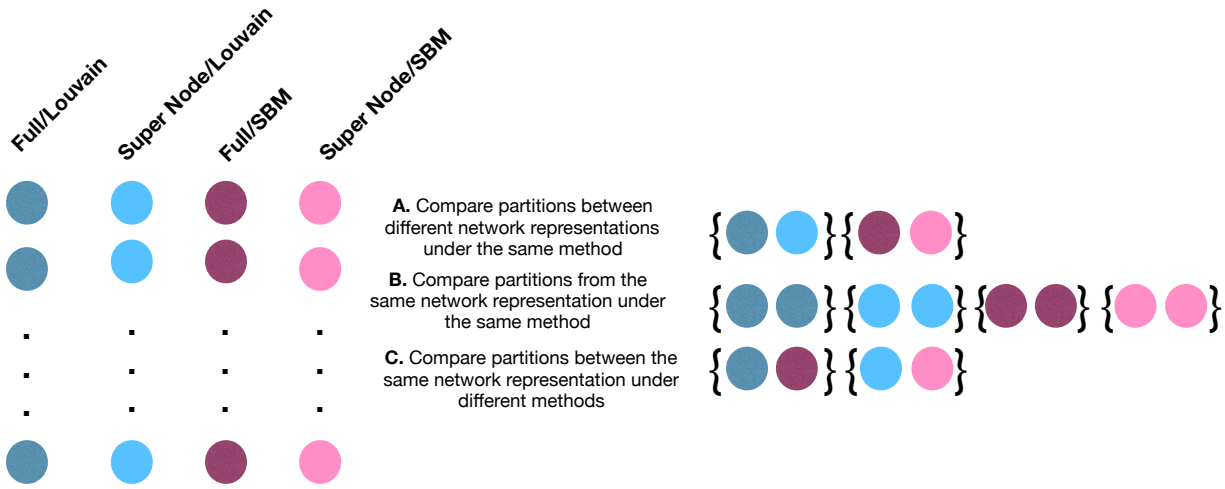

Figure 3.4: **Schematic of possible partition comparisons.** We outline the types of possible comparisons between partitions generated according to various combinations of network representation and community detection method. According to these comparison rules, we compute normalized mutual information (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.

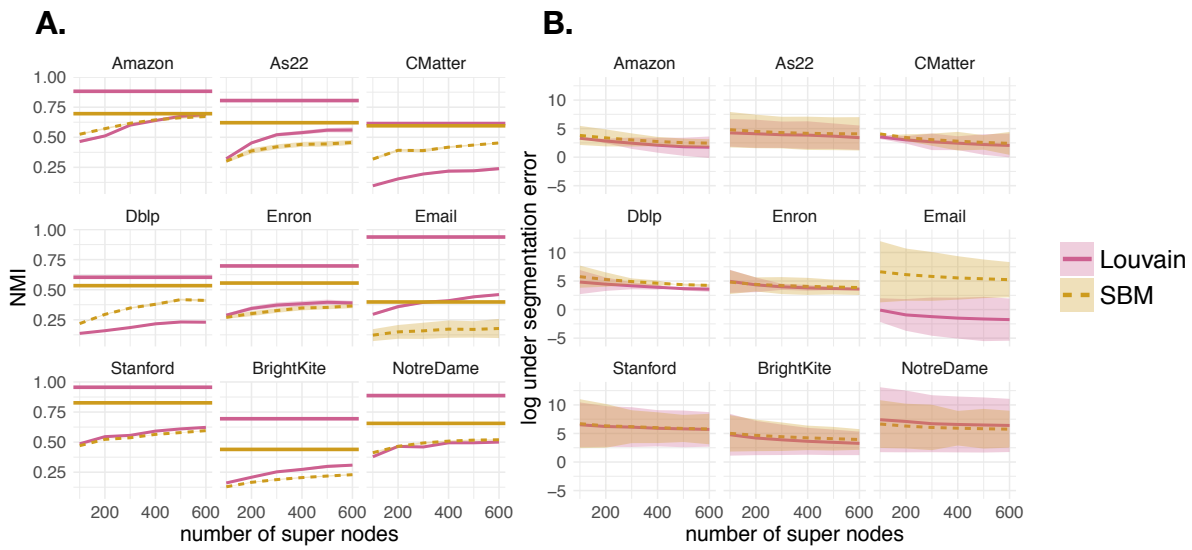

Figure 3.5: **Super Node Quality.** We computed normalized mutual information (**A.**) and under segmentation error (**B.**) for networks represented by between 100 and 600 super nodes. Line type and color indicate the community detection algorithm applied (Louvain algorithm or SBM fitting). Each curve 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. $\text{NMI}(\mathbf{z}^{Full}, \mathbf{z}^{SN})$]. 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.

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. $\text{NMI}(\mathbf{z}^{Full}, \mathbf{z}^{SN})$) 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,(113) becomes significantly slower on the full networks with more than 200,000 edges.

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 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.

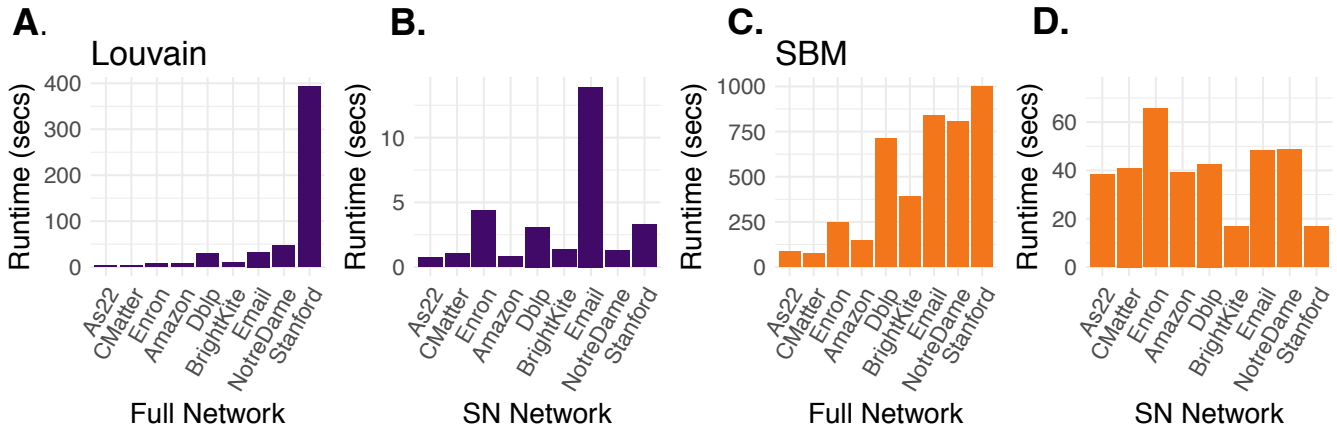

Figure 3.6: **Runtimes**. We compare community detection runtimes (in seconds) with the Louvain algorithm and by fitting an SBM on the full networks and super node representations for the 9 data sets. **(A.)** Louvain on the full network. **(B.)** Louvain on the super nodes. **(C.)** SBM on the full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

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 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

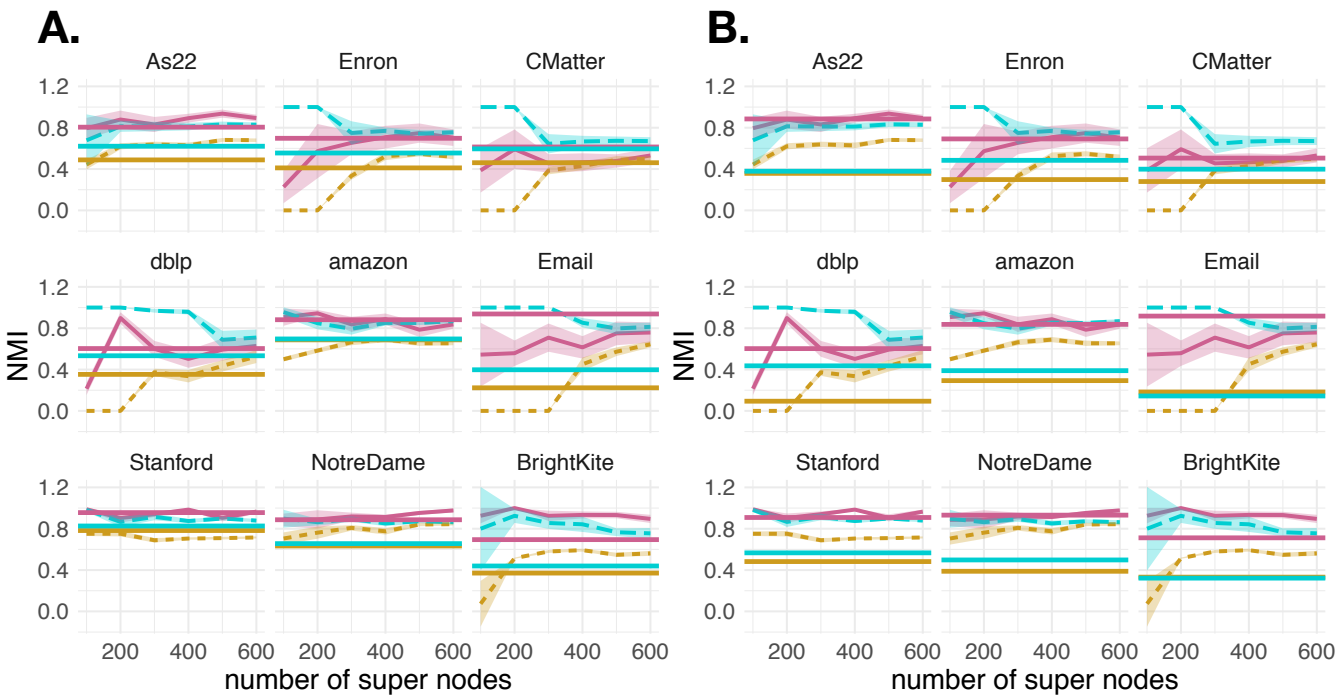

Figure 3.7: **Quantifying partition variability.** For each of the 9 networks, we obtained 10 different partitions by the Louvain algorithm and 10 different SBM fits under the default (**A.**) and matched settings (**B.**). To assess the similarity between partitions within and between a community detection 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.

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

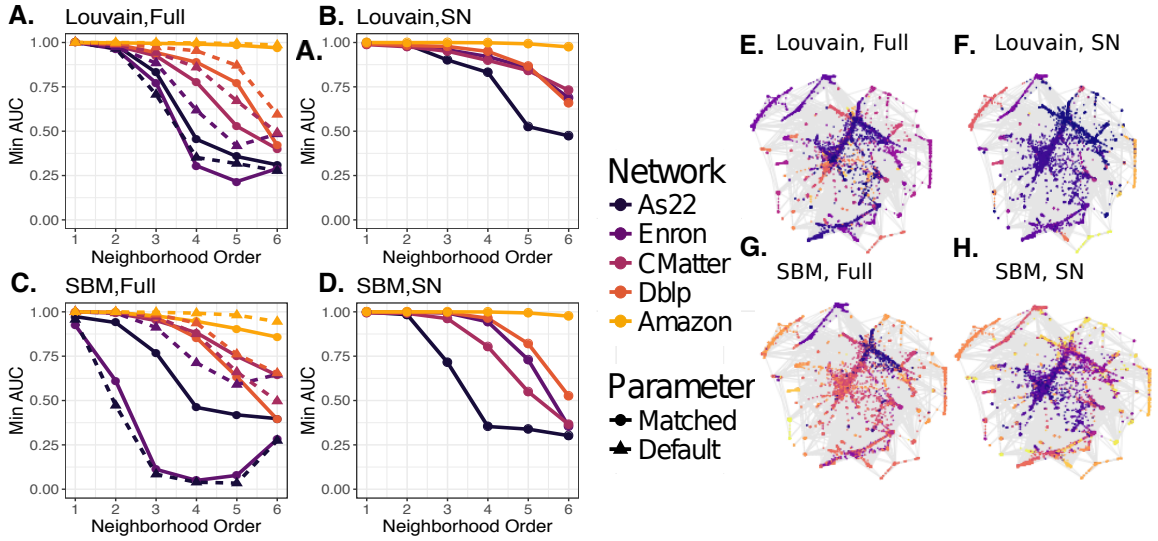

Figure 3.8: **Agreement of community assignments with local connectivity**. We study how consistent partitions are within local neighborhood regions of the network by examining how well a node's neighbors (for various order neighborhoods) can be used to predict its community assignment, under some community partition $\mathbf{z}$. 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.

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 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

Stochastic Block Models with Multiple Continuous Attributes

This work is done in collaboration with Thomas Bonacci, Roland Kwitt, Marc Niethammer, and Peter Mucha.

Stochastic block models (SBMs) are probabilistic models for community structure in networks, where nodes within a community are assumed to be connected to nodes within and between communities in a uniform, characteristic way. Typically, only the adjacency matrix is used to perform SBM parameter inference. In this paper, we consider circumstances in which nodes have an associated vector of continuous attributes that are also used to assign nodes to communities. While this assumption is not realistic for every application, our model assumes that the attributes associated with the nodes in a network's community can be described by a multivariate Gaussian model. In this augmented, attributed SBM, the objective is to simultaneously learn the SBM connectivity probabilities with the multivariate Gaussian parameters describing each community. While there are recent examples in the literature that combine connectivity and attribute information to inform community detection, our model is the first augmented stochastic block model to handle multiple continuous attributes. This provides the flexibility in biological data to, for example, augment connectivity information with continuous measurements from multiple experimental modalities. Because the lack of labeled network data often makes community detection results difficult to validate, we highlight the usefulness of our model for two network prediction tasks: link prediction and collaborative filtering. As a result of fitting this attributed stochastic block model, one can predict the attribute vector or connectivity patterns for a new node in the event of the complementary source of information (connectivity or attributes, respectively). We also highlight two biological examples where the attributed stochastic block model provides satisfactory performance in the link prediction and collaborative filtering tasks.

4.1 Introduction

As we have previously observed in this thesis, numerous approaches exist to identify communities in a network. In these methods, typically only the adjacency matrix encoding connectivity patterns is taken into account. In various 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, in a protein interaction network, each protein could contain multiple experimental measurements or classifications. Significant attention has been given to the interplay between connectivity-based (or structural) community organization of the network and the attribute information of nodes within communities. Importantly, it is often unclear whether it is valid to assume that a *structural* community should necessarily correlate with an attribute-based *functional* community (63; 111; 154). While such studies suggest that extreme caution should be taken in assuming a correlation between structural and functional communities, we limit our focus in the present 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 nodes 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. In doing so, our objectives are two-fold: first, we develop a probabilistic approach to jointly model connectivity and attributes; second, we wish to ensure that our model can handle multiple, continuous attributes.

4.1.1 Related work in attributed networks

Recently, there have been numerous efforts to incorporate attribute information into the community detection problem (155; 103; 34; 63; 111). In describing our contribution, we distinguish between methods that descriptively obtain communities through optimization of a quality function and those that generatively capture communities through probabilistic models. 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 (34). This method approaches the problem as an extension to the Louvain algorithm, which is the state-of-the-art scalable modularity quality function community detection method (23). The

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 the modularity quality function. The I-louvain method modifies the standard modularity quality function to what they label 'inertia-based modularity', incorporating a Euclidean distance between nodes based on their attributes, and demonstrating with multiple examples how incorporating connectivity and attributes allows for a partition of nodes to communities that aligns better with ground truth than that obtained using connectivity or attributes in isolation.

Alternatively, there a variety of probabilistic approaches to handling attributed network data (103; 63; 111; 155). Similar to our work in the sense that community membership is related to node attributes is CESNA (155). 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 (**?**) (at least the more standard variants of it), seeks to determine a hard partition of 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. A variant of the stochastic block model explored by Clauset *et al.*, (103) adapts the classic stochastic block model to handle a single attribute with the assumption that attributes (referred to as 'metadata') and communities are correlated. Hric *et al*. (63) developed an attributed SBM from a multilayer network perspective, with one layer modeling relational information between attributes and the other modeling connectivity, then assigning nodes to communities maximizing the likelihood of the observed data in each layer. Finally, work by Peel *et al.* made important contributions in 1) establishing a statistical test to determine whether attributes actually correlate 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 (111).

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 the attributes. We highlight two such examples in section 5, where we apply our model to a protein interaction network and a microbiome subject similarity network. Before discussing these examples, we first define our attributed SBM and an inference technique for fitting the model. We test this approach on a synthetic example. Since community detection methods are often difficult to validate due to the lack of ground truth information on the nodes, we describe the tasks of link prediction and collaborative filtering to quantify how well the attributed SBM represents the data. We then consider these tasks on two biological network examples.

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 Figure 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

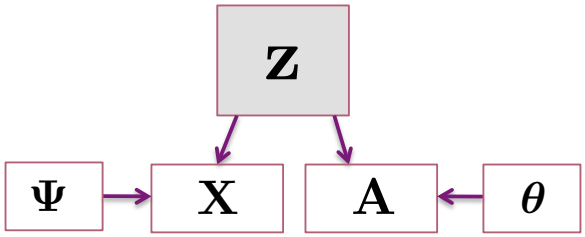

Figure 4.1: **Modeling community membership in terms of attributes and connectivity**. Node-to-community assignments specified by $\mathbf{Z}$ are determined in terms of adjacency matrix information, $\mathbf{A}$ and attribute matrix information, $\mathbf{X}$. $\mathbf{A}$ and $\mathbf{X}$ are assumed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

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

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

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

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

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 (45). By doing this, we will alternate between estimating the posterior probability that a node i has community label c, or

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

and estimates for $\boldsymbol{\theta}$, $\boldsymbol{\Psi}$, 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 $\boldsymbol{\mu}_i$ and $p \times p$ covariance matrix, $\boldsymbol{\Sigma}_i$. Note that these parameters uniquely identify a p-dimensional multivariate Gaussian distribution. To specify this model for all K communities, we define the parameter $\boldsymbol{\Psi} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots \boldsymbol{\Sigma}_K\}$.

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

$$P(\mathbf{X} \mid \mathbf{\Psi}) = \sum_{i=1}^{N} \log\{\sum_{c=1}^{K} \pi_c \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \mathbf{\Sigma}_c)\} \tag{4.4}$$

Here, $\mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \mathbf{\Sigma}_c)$ 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, $\boldsymbol{\theta}$, the complete data log likelihood can be expressed as

$$\begin{aligned} \log(P(\mathbf{A} \mid \mathbf{z})) = \frac{1}{2} \sum_{i \neq j} \sum_{k,l} z_{ik} z_{jl} [a_{ij} \log(\theta_{kl}) \\ + (1 - a_{ij}) \log(1 - \theta_{kl})] \ . \end{aligned} \tag{4.5}$$

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, $\boldsymbol{\theta}$ and $\mathbf{\Psi}$ 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

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

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 $\boldsymbol{\theta}$ and $\mathbf{\Psi}$ 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, $\boldsymbol{\mu}_c$, can be computed as

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

Similarly, the update for the covariance matrix describing a community, $\boldsymbol{\Sigma}_c$, is computed as

$$\boldsymbol{\Sigma}_c = \frac{\sum_{i=1}^{N} \gamma(z_{ic})(\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T}{\sum_{i=1}^{N} \gamma(z_{ic})} \quad . \tag{4.8}$$

To update the parameters of $\boldsymbol{\theta}$, we follow the method in (38) and update the probability of an edge existing between community q and l, given by $\boldsymbol{\theta}_{ql}$ as,

$$\boldsymbol{\theta}_{ql} = \frac{\sum_{i \neq j} \gamma(z_{iq})\gamma(z_{jl})x_{ij}}{\sum_{i \neq j} \gamma(z_{iq})\gamma(z_{jl})} \tag{4.9}$$

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 (23). 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. We generated networks with a stochastic block model with $N = 200$ nodes and $K = 4$ communities, parameterized as follows:

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

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

Fig. 4.2(A) 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 assortative 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 each $c \in \{1, 2, 3, 4\}$ is an 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 $\mathbf{x}_i$ for node i is generated as

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

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

Fig. 4.2(B) shows a PCA plot of the attribute vectors associated with each node in an example synthetic experiment, with each point representing a node. Since the true dimension of these feature vectors is 8, this plot provides a projection onto the first 2 principal components, allowing a visualization of the relatedness between node attributes. One can observe that members of community 2 are overall nicely separated from other communities in the projected attribute space but members of communities 3 and 4 are especially hard to discern here.

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 clustering based only on connectivity and to clustering based only on the attribute information. We quantify the correctness of the obtained partitions with normalized mutual information (NMI) (36). 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 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, we obtain 0.65, 0.68, and 0.83, respectively. These results show that by combining both sources of information, there

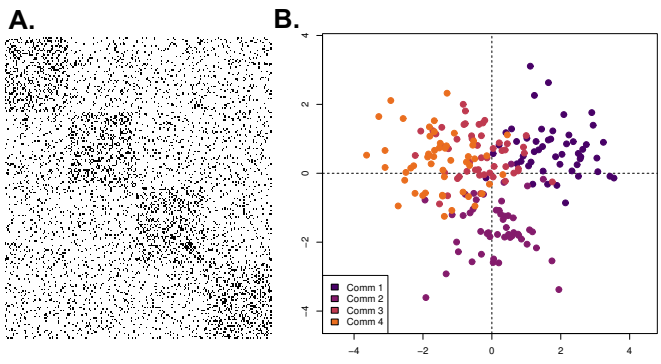

Figure 4.2: **Synthetic Example.** We generated a synthetic network with $N = 200$ nodes, $K = 4$ communities and an 8-dimensional multivariate Gaussian for each community. **A.** A visualization of the adjacency matrix for this network where a black dot indicates an edge. We observe that there is an assortative block structure (blocks on the diagonal), but there are also many edges between communities making the true community structure using only connectivity harder to detect. **B.** We performed PCA on the $N \times p$ 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, yielding $\text{NMI}(\mathbf{z}, \{\mathbf{z}^{\text{connectivity}}, \mathbf{z}^{\text{attributes}}, \mathbf{z}^{\text{attribute sbm}}\}) = \{0.65, 0.68, 0.83\}$.

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; in particular, sharp phase transitions are observed in fitting stochastic block models, with accurate capture of the correct communities only if the within-community probability, p_{in}, is sufficiently larger than the between-community probability, p_{out} (43; 136).

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 may 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 4.2 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 4.3 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.

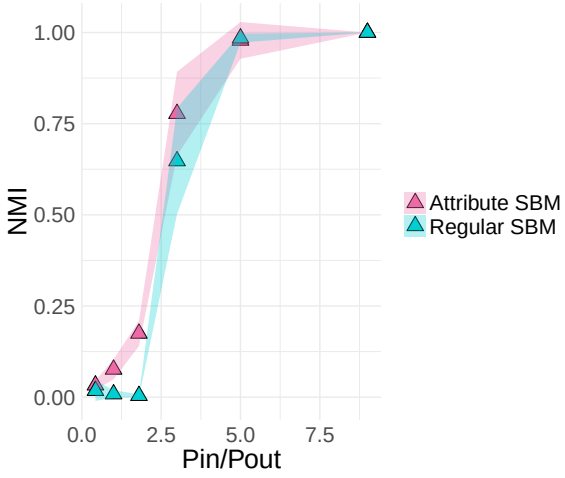

Figure 4.3: **Detectability Analysis in Synthetic Example.** To understand how attribute information can be combined with connectivity to assign nodes to communities accurately, we generated synthetic networks for within-probabilities of p_{in} between 0.05 and 0.3 with corresponding p_{out} or between-community 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 improves results, particularly near and below the detectability limit, and appears to smooth out the sharp phase transition.

We see that while both inference approaches undergo a strong increase in accuracy at a similar ratio of $p_{in}/p_{out} = 3$, we notice that the curve for the attribute SBM results are slightly shifted to the left due to the use of the extra attribute information positively impacting the ability to correctly identify communities. Moreover, we note that the attribute SBM results appear to smooth out the sharp phase transition that is visible in the results from the SBM without attributes. Future work could focus on better understanding the impact on such detectability questions in terms of the parameters for the underlying multivariate Gaussian distributions parametrizing each community.

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). We demonstrate here that fitting an attributed SBM may provide a means to successfully perform two fundamental network prediction tasks: link prediction and collaborative filtering.

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 edge-candidate dyads and ultimately the top x set of prospective 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 (147), we can calculate the score of the potential link as Score(m, n).

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

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

Preferential Attachment: Score$(m, n) = |\Gamma(m)| \times |\Gamma(n)|$

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 (129). 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.

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

Weighted Neighborhood Avg:

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

We show results for these two tasks in two different biological network examples in section 5. In particular, the experiments were designed in the following ways.

4.4.1 Link Prediction Experiments

For the link prediction tasks shown in Figures 4.5 and 4.9, 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 the input to each of the 3 baseline community detection methods (Jaccard, Adamic Adar, and Preferential Attachment). 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 (133).

4.4.2 Collaborative Filtering Experiments

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 4.6 and 4.10. 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 a relative error measure, $\mathcal{E}$, such that

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

Similar to the success of integrating community information for link prediction, collaborative filtering tasks have previously shown success from the integration of network community structure (46).

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. For example, one might consider 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 here, 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

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 (75). 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.

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 constructed 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 we simply ignored the edge weights as input to the model.

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 we observed an approximately Gaussian distribution.

We first visualized the differences in partitions obtained according to the classic and attributed stochastic block models in Figure 4.4A-B, respectively. In both networks, nodes are colored by their

community assignment. Using the classic stochastic block model and the model selection criteron described in (38), 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 4.4A there is mixing between the dark and lighter purple communities in the upper left of the network. In Figure 4.4B, this mixing was reduced by assigning all of the nodes in the general region to the lighter purple community.

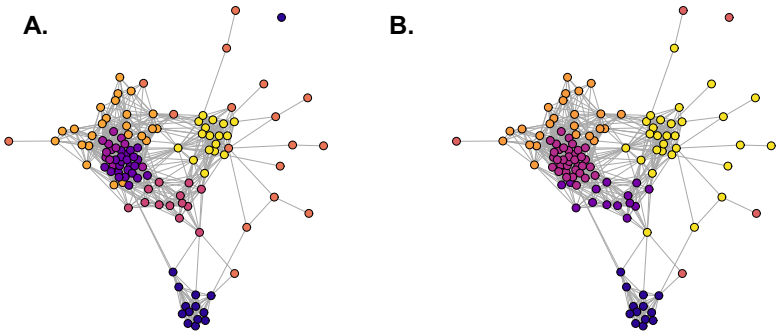

Figure 4.4: **Microbiome subject similarity network:** A visualization of the 121 node microbiome subject similarity network with nodes colored by the partition using the classic (**A.**) and attributed (**B.**) stochastic block model. **A.** Fitting the classic stochastic block model to the network, 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.**

Microbiome Link Prediction We performed link prediction on the microbiome subject similarity network. The associated ROC curves are plotted in Figure 4.5. 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.

Microbiome Collaborative Filtering We performed the collaborative filtering experiments on the microbiome subject similarity network to predict the 5-dimensional attribute vector for each node. The box plots in Figure 4.6 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

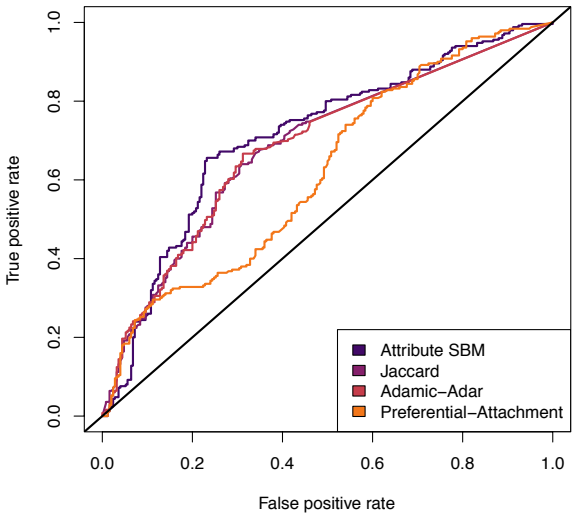

Figure 4.5: **Link Prediction on the microbiome subject similarity network:** The results for link prediction on the microbiome subject similarity network for the attributed SBM, Jaccard, Adamic-Adar and preferential attachment methods. The corresponding AUC values for these methods, respectively are,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

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 (25). This network represents interactions between proteins, predicted from the literature. Associated with each 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 (e.g. 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.

Data Pre-Processing: We downloaded the unweighted protein interaction network data and the modification information from the supplement of (25). We removed 13 nodes that were not

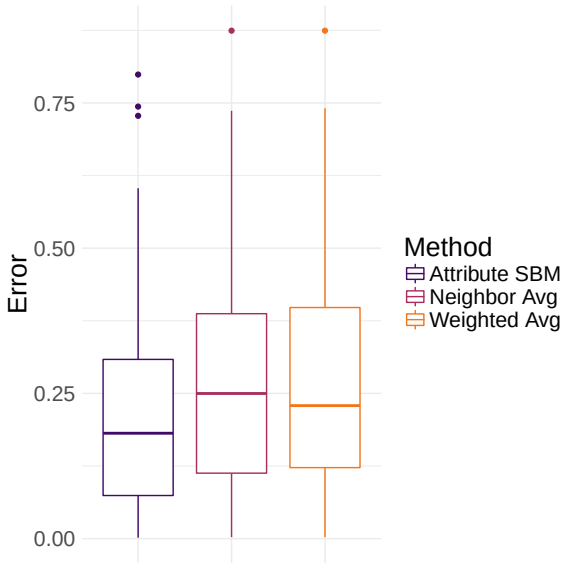

Figure 4.6: **Collaborative Filtering Accuracy in Microbiome Subject Similarity Network**: For each of the 121 nodes, we fit a model to the remaining 120 node network and given the node's closest neighbors (based on network connectivity) sought to predict its 5-dimensional attribute vector. The reported 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.

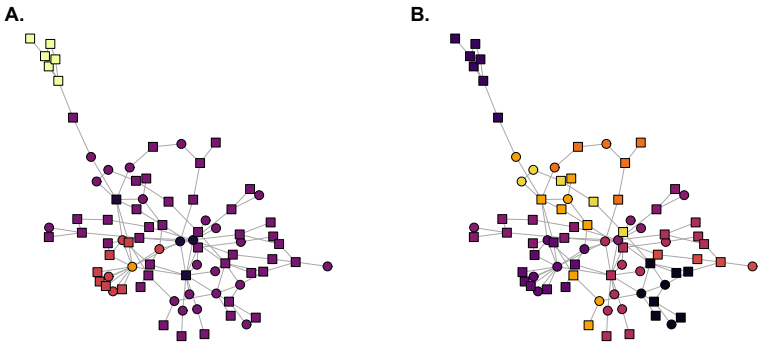

Figure 4.7: **Protein interaction network.** We visualize the 82 node protein interaction network under the classic stochastic block model **A.** and the attributed stochastic block model **B.** In both networks, nodes are colored by their community assignment and the node shape indicates whether the 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.

connected to the largest component of the network and considered only the 82 node largest connected component.

Constructing Node Attributes: Each node is classified with 1 of 6 possible modification types. 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 4th 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 4.7A-B show the results of fitting a classic SBM and attributed SBM, respectively, with nodes colored by community assignment. The 6 possible modifications exist based on 3 biological processes that can either increase or decrease after exposure to the drug. 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 (38), 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.

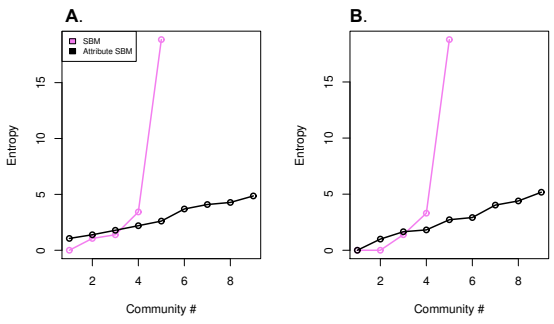

Figure 4.8: **Community entropies in the protein interaction network.** We studied the entropy of the 2 class and 6 class classifications of the nodes in **A.** and **B.**, respectively under the classic SBM (black) and attributed SBM (purple) partitions. For $\mathbf{A.} - \mathbf{B.}$ the horizontal 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.

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 4.8A-B, we plot the entropy for the 2 class and 6 class node classifications, respectively. We define $\mathbf{E}_c$, the entropy for community c as

$$\mathbf{E}_c = -\sum_k p_k \log(p_k). \tag{4.13}$$

Here, k indexes 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.

Link Prediction in the Protein interaction network We performed link prediction on the protein interaction network. 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 4.9.

Collaborative filtering in the protein interaction network Collaborative filtering were performed. 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 4.5, the box plots in Figure 4.10 represent the distribution of errors across each of the 82 nodes.

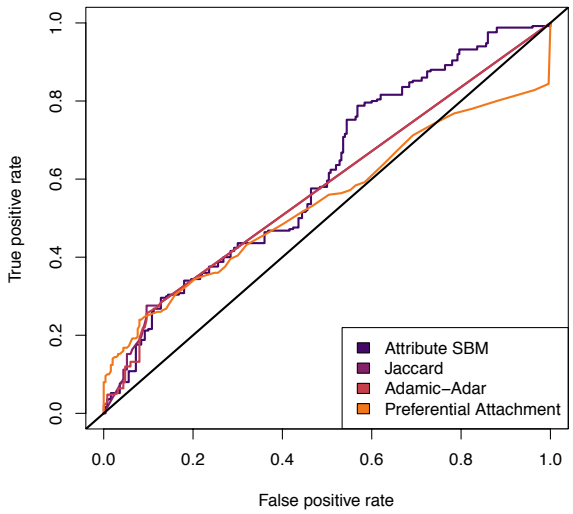

Figure 4.9: **Link Prediction in the protein interaction network**. Performing link prediction using the attributed SBM, Jaccard, Adamic Adar, and preferential attachment. The corresponding AUC curves for these methods were 0.61, 0.58, 0.58, and 0.51, respectively.

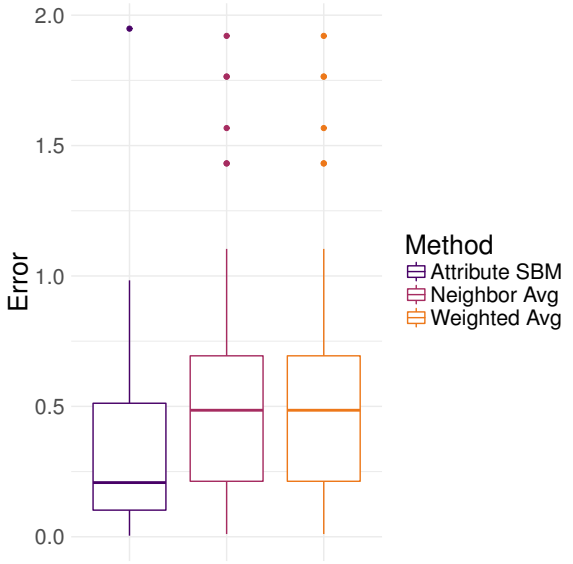

Figure 4.10: **Collaborative filtering in the protein interaction network**. For each of the 82 nodes, we fit a model to the remaining 81 node network and given the node's closest neighbors (based on network connectivity) sought to predict its 6-dimensional attribute vector. The reported error is the 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

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, with the assumption 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 absence of a known ground truth, 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 observed detectability properties in Figure 4.3, noting that it would be interesting to characterize how the properties of the attributes and connectivity 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

Testing the Alignment of Node Attributes with Network Structure

Attributed network data is becoming increasingly common across fields, as we are often equipped with information about nodes in addition to their pairwise connectivity. This extra information can manifest as a single value or classification for the nodes, or as multidimensional vector of features. Recently developed methods that seek to extend community detection approaches to attributed networks have explored how to most effectively combine connectivity and attribute information for quality identification of communities. These methods often rely on some assumption of the dependency relationships between attributes and connectivity. In this work, we seek to develop a statistical test to assess whether node attributes align with network connectivity. The objective is to quantitatively evaluate whether nodes with similar connectivity patterns also have similar attributes. To address this problem, we will use a node sampling and label propagation approach. We apply our method to several synthetic examples that explore how network structure and attribute characteristics effect the empirical p-value computed by our method. Finally, we apply the test to a network generated from a single cell mass cytometry dataset and show that our test can identify markers associated with particular inferred cellular phenotypes.

5.1 Introduction

Community detection in networks is a common pursuit that seeks to partition the network's nodes into sets of structurally coherent groups, where members of a group of *community* have strong similarity in connectivity patterns (95; 52; 128). While the identification of communities based solely on the network's adjacency matrix is straight forward, the implications of having extra information about the network nodes and how to integrate that into the community detection problem is not

well-understood. We refer to a *structural community* as a community identified according to only the adjacency matrix, while an *attribute community* can be thought of as a community identified using the attribute information. Recently, there have been numerous approaches extending common community detection techniques to attributed networks (63; 111; 34; 155; 103; 118). While each of these methods provide extensions to different community detection approaches, they also differ in their assumption about the dependence relationships between the attributes and connectivity. On one hand, it seems reasonable to assume that members of a structural community should be highly similar in attribute space seems like a valid assumption. However, work by Clauset *et al.* (103) and Peel *et al.,* (111) have provided examples of when this assumption could be invalid. In this work, we seek to develop a test that returns a statistic providing insight into how closely the node attributes correlate with connectivity patterns. Our test is based on label propagation and ultimately returns an empirical p-value that can be interpreted as the significance of the relationship between network connectivity and node attributes. We validate that the empirical p-value is meaningful with several synthetic examples and on a network representation of a single-cell mass cytometry dataset.

This paper is organized as follows: First, we will describe the latest advances in attributed community detection. Next, we will describe our method and validate the quality of the computed empirical p-value on synthetic examples and on a single cell mass cytometry dataset.

5.1.1 Attributed Network Community Detection Methods

The integration of attribute and connectivity information has been explored thus far with stochastic block models and modularity-based methods. In this section, we will describe this related work. The discussion here will be similar to that in Section 4.1, but will align more closely with our proposed problem of quantifying the overlap between connectivity and attributes.

5.1.1.1 Probabilistic approaches

The assumption of the stochastic block model is that nodes within a community are connected to nodes within and between communities in a characteristic way. Moreover, the objective in the fitting and parameter inference of a stochastic block model in a network with K communities is to learn the node-to-community assignments and the within and between community connection probabilities that maximize the model likelihood. The stochastic block model has been extensively studied in

the literature and has at least three attempts to be extended to attributed networks (63), (111), (103). First, Clauset *et al.*, modified the traditional stochastic block model likelihood to incorporate a piece of metadata (103). Related to this work, Peel *et al.*, proposed the neoSBM (111), which developed a permutation-based test to assess the relatedness of attributes and connectivity and incorporated the information only to the extent to which they were aligned. Next, Hric *et al.*, constructed a joint stochastic block model for both the attributes and metadata through a nonparametric, bayesian framework (63). They assessed the alignment of the attributes with the connectivity based on their application in link prediction tasks.

The affiliation model assumes that nodes can be affiliated to multiple communities to varying extents (153). Moreover, the edges between a pair of nodes is based on their similarity in community affiliations. A useful method for integrating multidimensional vectors of binary attributes was introduced by Yang *et al.* in a method called CESNA (155), which modifies the affiliation model likelihood to incorporate this information. This is achieved by allowing the attributes and connectivity information to be modeled as conditionally independent, giving the node-to-community affiliations and feature importance weights for the attributes.

5.1.1.2 Quality function maximization

The next class of methods with extensions to attributed network is the class of quality function maximization techniques. When community detection is formulated with a quality function, the objective is to specify a null model for a network with no community structure and find the partition of nodes to communities that maximizes the difference from this null model. A standard quality function for communities is known as modularity (102). The state-of-the-art optimization heuristic for maximizing modularity is the Louvain algorithm (23). Work by Combe *et al.,* adapted the modularity to take into account multidimensional attribute vectors and optimized this quantity in a Louvain-style manner with 'i-louvain' (34).

Finally, recent work by Perozzi *et al.,* defines an extension to modularity known as *community normality* (118). This measure prioritizes partitions where members of a community are very similar to each other in attribute space (and obviously in connectivity patterns). Further, members of a community are also expected to be different from nodes on the community boundary or in a different community.

While the methods profiled in this section indicate great progress in the integration of attributes in community detection, our objective in this paper are as follows. First, we seek to define a statistic (in the form of an empirical p-value) that reflects the alignment between networks and connectivity. Second, seek to explore the properties of this empirical p-value and how it responds to different types of networks and attributes. Finally, we show that our empirical p-value is useful in confirming immunological markers that can separate cells with different phenotypes in a single cell dataset. This suggests that the empirical p-value returned by our method can also be used as measure of feature discriminative power.

5.2 Methods

This method is built on label propagation (LP), where given a set of partially labeled nodes in the network, the objective is to take a partially labeled network and use this information to predict the labels of the unlabeled nodes (151). In this work, we first *label* the nodes from their attribute information and then take several samples of labeled nodes and use the performance on the prediction of the unlabeled nodes as a proxy for how closely the attributes align with the network connectivity. In particular, we use a label propagation approach that returns a probability distribution for each node over each of the attribute defined node classes. We then quantify the uncertainty in the prediction with a simple entropy measure. In doing this, we assume that if the attributes are aligned with the network connectivity patterns, the entropy should be low. Alternatively, if attributes and connectivity are disparate, then predicting the unlabeled nodes will be difficult and entropy should be higher.

As an overview of this process, we first label the nodes according to their attribute information. This can be achieved by classifying the nodes according to a single, discrete value, or through simple clustering of the nodes, based on their attributes. After obtaining a label of the nodes, we begin our label propagation and permutation process. For a large number of trials, S, we take a sample of the nodes and their corresponding labels according to the attribute information and denote these nodes as *labeled*. We then try to predict the labels of the remaining nodes, or the *unlabeled* set, by propagating the labels outward. The label propagation method we use returns a probability distribution for each of the unlabeled nodes, which allows us to compute an entropy measure. Alongside this process, in each of these trials, we also permute the labels of the nodes in our sample set to generate a null

distribution of entropy values for the unlabeled nodes. Finally, the overlap between the null and empirical entropy distributions are used to compute a p-value. This process is outlined in Figure 5.1. We will now provided a detailed description of each step in this process.

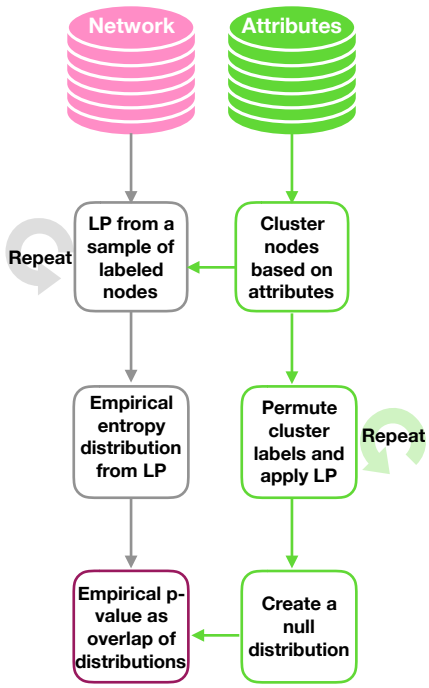

Figure 5.1: **Overview of the method**. Our test first labels the nodes according to attribute information, $\tilde{\mathbf{z}}$. Then in a collection of T trials, a sample of l nodes is treated as labeled, according to $\tilde{\mathbf{z}}$. In each trial, a label propagation task is performed to predict the probability distribution over communities for the unlabeled $N - l$ nodes. The entropy of the node-to-community assignment probabilities is used as an estimate of how well the attributes align with connectivity. Also in each trial, $\tilde{\mathbf{z}}$ is permuted and subjected to the label propagation task to compute a 'null' entropy value. After repeating this process in T trials, the empirical p-value is calculated based on the overlap between the null entropy distribution and the empirical entropy distribution.

5.2.1 Notation

For convenience, we define some notation that assists in setting up this problem. For a network with N nodes, we let $\mathbf{z}$ be the N-length vector of node-to-community assignments, based on only the network connectivity information given in adjacency matrix, $\mathbf{A}$. This implies that the i-th entry, z_i gives the community assignment for node i. Alternatively, when nodes are labeled according to the attribute information, we denote their community assignments with $\tilde{\mathbf{z}}$. Finally, our permutation test involves taking a subset of nodes and their labels in $\tilde{\mathbf{z}}$ to treat as the *labeled* nodes and propagate the

labels out to the unlabeled nodes. We denote this distinction by $\tilde{\mathbf{z}}^L$ and $\tilde{\mathbf{z}}^U$, denoting the community labels for the labeled and unlabeled subsets, respectively. Finally, we assume that each node has p associated attributes, which are stored in the $N \times p$ matrix, $\mathbf{X}$. That is, the ith row of $\mathbf{X}$, X_i gives the values of the p attributes for node i.

5.2.2 Classifying Nodes

The first step is to classify nodes according to attributes, denoted by $\tilde{\mathbf{z}}$. We assume some prior knowledge for the K, specifying many communities are in the data, hence each $\tilde{z}_i$ takes on 1 of K values. In the case where nodes are classified discretely, according to a single source of information, this labeling occurs without any effort. In the case where each node has multiple attributes, we have found that a simple clustering method, such as k-means works well.

5.2.3 Sampling Nodes and Creating Entropy Distributions

In the sampling step, for a large number of trials, S, we randomly select l nodes, $\{L\}$ and their corresponding labels, $\tilde{\mathbf{z}}^L$. From here, we seek to predict the labels for the the remaining $N - l$ nodes that comprise the unlabeled set, $\{U\}$, with labels $\tilde{\mathbf{z}}^L$.

After splitting all N nodes into their labeled an unlabeled sets, we use the label propagation approach described by Zhu *et al.,* (162) to generate a probability distribution for each of the nodes in $\{U\}$. Ultimately, we seek to define the $N \times K$ matrix, $\mathbf{Y}$, where Y_{ic} is the probability that node i belongs to class c. We can split this matrix into two matrices, $\mathbf{Y}^L$ and $\mathbf{Y}^U$ with the containing the subset of rows corresponding to nodes in $\{L\}$ and $\{U\}$, respectively. Therefore, the label propagation task is to effectively estimate $\mathbf{Y}^U$. We use $\tilde{\mathbf{z}}^L$ to initialize $\mathbf{Y}^L$ so that for node i, with $\tilde{z_i}^L = c$, $Y_{ic}^L = 1$ and all other elements in the row are 0.

To compute $\mathbf{Y}^U$, we first compute the row-normalized adjacency matrix, $\bar{\mathbf{A}}$. That is, $\bar{A_{ij}} = A_{ij} / \sum_t A_{it}$. We then rearrange $\bar{A_{ij}}$ so that the first l rows and columns correspond to the labeled nodes, and the next $N - l$ rows and columns correspond to the unlabeled nodes. To do this, $\mathbf{A}$ is split into 4 submatrices after the lth row and lth column.That is, we write $\bar{\mathbf{A}}$ as,

$$\bar{\mathbf{A}} = \begin{bmatrix} \bar{A}_{ll} & \bar{A}_{lu} \\ \bar{A}_{ul} & \bar{A}_{uu} \end{bmatrix}.$$

From here, we iteratively update $\mathbf{Y}^U$ according to the update rule provided by Zhu *et al.,* (**?**) as,

$$\mathbf{Y}^U \leftarrow \bar{A}_{uu}\mathbf{Y}^U + \bar{A}_{ul}\mathbf{Y}_L. \tag{5.1}$$

In practice, $\mathbf{Y}^U$ is continuously updated until convergence.

Computing $\mathbf{Y}^U$ for one pair of $\mathbf{L}$ and $\mathbf{U}$ comprises the true label propagation task of one trial. To generate our null distribution, we first permute the entries of $\tilde{\mathbf{z}}^L$, and denote this permuted version as $\tilde{\mathbf{z}}^L_{\text{perm}}$. Just as we showed in the true label propagation task, we use $\tilde{\mathbf{z}}^L_{\text{perm}}$ to define a corresponding permuted version of $\mathbf{Y}^L_{\text{perm}}$ with $Y^L_{\text{perm}_{ic}}$ set to be 1 if node i belongs to community c, under the permuted labels, given by $\tilde{\mathbf{z}}^L_{\text{perm}}$. The analogous update relationship shown in equation 5.1 gives $\mathbf{Y}^L_{\text{perm}}$.

After computing $\mathbf{Y}^U$ and $\mathbf{Y}^U_{\text{perm}}$, the next step is to compute their corresponding entropies, E and E_{perm}. We compute entropy $H(x)$ as,

$$H(x) = -\sum_{ic} p_{ic}\log(p_{ic}). \tag{5.2}$$

Moreover, $H(\mathbf{Y}^U)$ and $H(\mathbf{Y}^U_{\text{perm}})$ give E and E_{perm}, respectively. We let $\mathcal{E} = \{E_1, E_2, \ldots E_T\}$ and $\mathcal{E}_{\text{perm}} = \{E_1, E_2, \ldots E_T\}$ be the collection of entropies over the T trials.

5.2.4 Computing the empirical p-value

After having performed T trials, we compute the empirical p-value for the test and is interpreted as the overlap between $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$. In the case where attributes, $\mathbf{X}$ and connectivity $\mathbf{A}$ are well-aligned with connectivity, $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$ should not overlap because the entropy for the label propagation task should be very low. Alternatively, as $\mathbf{X}$ and $\mathbf{A}$ become less aligned, the entropy of the prediction from the label propagation task should be higher and hence $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$ will overlap. Then the empirical p-value, p is calculated as,

$$p = P(\mathcal{E}_{\text{perm}} < \max(\mathcal{E})). \tag{5.3}$$

That is, our interpretation of the empirical p-value is the proportion of $\mathcal{E}_{\text{perm}}$ that are less than the maximum value of $\mathcal{E}$.

5.3 Results

We present results on synthetic networks and on a network representation of a single cell mass cytometry dataset. In this section, we seek to confirm that the empirical p-value leads to an accurate and interpretable conclusion. The results on synthetic data re useful because we have an understanding of when the p-value should be significant, due to our knowledge of how the data were generated. Similarly, in the single cell mass cytometry dataset, we use particular marker features to validate our computed empirical p-values.

5.3.1 Synthetic Examples

First, we sought to verify that our empirical p-value was capturing desirable behavior. First, we expected the p-value to decrease in significance as the label propagation distribution increases in overlap with the empirical null distribution. Second, we sought to have a p-value that decreased in significance as the entropy of the partition increased. In Figure 5.2, we considered a network generated from a stochastic block model with $N = 200$ nodes, $K = 4$ communities, within-community edge probability (p_{in}), $p_{in} = 0.6$, and between-community edge probability, (p_{out}), $p_{out} = 0.02$. That is for a pair of nodes, i and j, the probability of an edge existing between them is modeled as $P(A_{ij} = 1) = p_{in}$ if $z_i = z_j$ and $P(A_{ij} = 1) = p_{out}$ if $z_i \neq z_j$.

Associated with each node is a 3-dimensional Gaussian attribute vector, drawn from 1 of K multivariate Gaussian distributions. Under this formulation, each community has its own associated multivariate Gaussian distribution and attribute vectors and a node in community k is parameterized by mean $\boldsymbol{\mu}_k = [\mu_1, \mu_2, \mu_3]$ and covariance matrix $\boldsymbol{\Sigma}_k$.

To generate each $\boldsymbol{\mu}_k = [\mu_1, \mu_2, \mu_3]$, we draw each μ_d from a standard Normal distribution with mean 0 and unit variance. For a community k, $\boldsymbol{\Sigma}_k$ is also the identify covariance matrix.

When performing our label propagation task, we used a sample of $S = 1000$ nodes and repeated the experiment $T = 1000$ times.

First, we visualized the distribution of the attributes, using a 2-dimensional projection with tSNE (88). In Figure 5.2A., each point represents a node and is colored by its community assignment, $\mathbf{z}$. We can see that there are clearly clusters of nodes from the same community, but there is also a some mixing. To test how the empirical p-value behaved as the label propagation distribution converged to

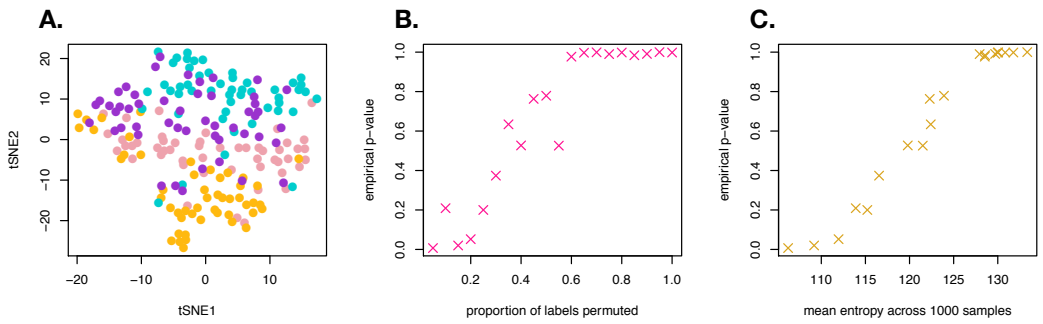

Figure 5.2: **Properties of the empirical p-value.** To understand the properties of our empirical p-value, we generated a synthetic network, $\mathbf{A}$ from an SBM with $N = 200$ nodes, $K = 4$. The vector of continuous attributes for a node i, (X_i) was drawn from a multivariate Gaussian distribution parameterized by its community assignment ($\mathbf{z}$) or $\{\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}\}$. In these experiments, we permuted varying fractions of $\tilde{\mathbf{z}}$ and observed the effects on entropy and empirical p-value. **A**. We used tSNE to visualize the two dimensional projection of the 200 nodes. For the most part, members of the same community cluster together. **B.** We plotted the empirical p-value as a function of the proportion of labels permuted and observed decreased statistical significance (increased empirical p-value) with an increasing proportion of permuted labels. **C.** We plotted the empirical p-value as a function of the mean entropy ($\mathcal{E}$) across $T = 1000$ trials used to generate the entropy distributions for each experiment. Increased entropy corresponding to a larger proportion of $\tilde{\mathbf{z}}$ permuted leads to a decreased p-value.

the null distribution, we experimentally perturbed various proportions of the node labels based on attributes ($\tilde{\mathbf{z}}$) that were input to the label propagation task. This test was implemented to verify that with a higher proportion of permuted entries in $\tilde{\mathbf{z}}$, the empirical entropy distributions, $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$ would have more extensive overlap. As expected, in Figure 5.2B., we observed that by permuting a larger proportion of the labels, $\tilde{\mathbf{z}}$, there was an associated increase in the empirical p-value (decreased significance). Finally, in Figure 5.2C. we examined the relationship between the average entropy across each element of $\mathcal{E}$ obtained over the 1000 simulations, and the empirical p-value. As expected, these quantities are highly related, with a higher entropy leading to a more significant empirical p-value.

5.3.1.1 Comparison to BESTest

We used the synthetic data from the experiment described in Figure 5.2 to compare our results to those obtained using BESTest. BESTest is the method developed by Peel *et al.*, to inform their attributed neoSBM model of how informative attributes are on community structure (111). BESTest works first by labeling the nodes according to $\tilde{\mathbf{z}}$, based on the attribute information. Under this

partition of the nodes, the SBM parameters are optimized, where the maximum likelihood estimate for the connection probability between a pair of communities r and s is given by $\hat{\omega}_{rs}$. This maximum likelihood estimate $\hat{\omega}_{rs}$ is computed as $\hat{\omega}_{rs} = m_{rs}/n_r n_s$. Here, m_{rs} is the number of edges between communities r and s, while n_r and n_s are the number of nodes in communities r and s, respectively. The entropy, $\mathcal{H}$ of this partition across the communities is computed as,

$$\mathcal{H}(\tilde{\mathbf{z}}) = -\frac{1}{2}[\sum_{rs} \log \hat{\omega}_{rs} + (n_r n_s - m_{rs}) \log(1 - \hat{\omega}_{rs})] + O(N^{-1}). \tag{5.4}$$

The empirical p-value is computed with BESTest through a permutation test which computes $\mathcal{H}(\tilde{\mathbf{z}}_{\text{perm}})$ many times and reports the fraction of $\mathcal{H}(\tilde{\mathbf{z}}_{\text{perm}}) < \mathcal{H}(\tilde{\mathbf{z}})$. This method is appropriate and useful for the attributed SBM. We distinguish our method because it is testing alignment of the connectivity in genera and does not make an assumption about the stochastic block model being an appropriate model for the data.

Varying proportions of the attribute labels allowed us to examine different levels of entropy on the partition of a held-out set. We used $T = 1000$ permutations to compute the BESest empirical p-value. In Figure 5.3A., we looked at the relationship between the p-value under the LP test (horizontal axis) and from BESTest (vertical axis). We see that there is broader range of p-values with LP, in comparison to BESTest, which seems to jump quickly from not-significant to significant. In Figure 5.3B., we plotted the BestTest empirical p-value as a function of the BESTest entropy. We observed a pattern very similar to Figure 5.3A., where a wide range of entropies across experiments are not reflected in a continuously changing p-value. Next, we sought to understand how the BESTest entropy compared with the p-values computed with LP. In Figure 5.3C., we observe a wider range of p-values across the continuum of observed entropies. Finally, in Figure 5.3D. , we examined the relationship between LP and BESTest entropies and found them to well correlated ($r = 0.95$). This suggests that both tests are capturing the same level of uncertainty, but the interpretation of the empirical p-value differs.

5.3.1.2 Strength of community structure

Now we explored how structural communities, according to connectivity, influence the entropy and corresponding empirical p-values. We refer to a strong community structure as one that has many

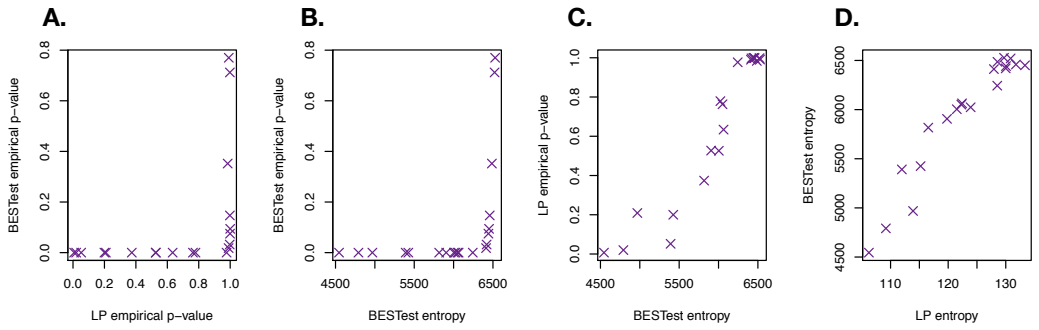

Figure 5.3: **Comparison with BESTest.** We sought to understand the relationship between our empirical p-value and that computed according to BESTest. To study this, we used the same experiment described in Figure 5.2, where we varied the proportion of permuted labels from $\tilde{\mathbf{z}}$. We denote our empirical p-value by 'LP empirical p-value. **A.** We plotted the BESTest empirical p-value against our LP empirical p-value. **B.** We plotted the BESTest empirical p-value as a function of the BESTest entropy. BESTest gives a significant empirical p-value for a much wider range of entropy levels than our test. **C.** The experiments produced a wide range of entropies under BESTest, which are captured by corresponding differences in our empirical p-value. **D.** We compared the BESTest approach to computing entropy to our LP method and observed a high correlation between these entropy measures ($r = 0.95$).

within-community connections and few between-community connections. To approximate this, we considered the p_{in} to p_{out} ratio for a stochastic block model. As previously described, p_{in} is defined as the probability of observing an edge between a pair of nodes in the same community, while p_{out} is the probability of observing an edge between a pair of nodes in different communities. We expected that the entropy and empirical p-value would increase with an increasing p_{in}/p_{out} ratio. That is, as the community structure becomes less prominent with an increased number of connections between communities, the label propagation task should become more difficult. To study this with synthetic data, we varied the p_{in}/p_{out} ratio, by considering a four community stochastic block model with values of p_{in} between 0.05 and 0.45 and choosing a corresponding p_{out}, such that the mean degree was equal 30. For each pair of p_{in} and p_{out}, we generated 10 synthetic stochastic block models. Accompanying each synthetic network was a fixed 3-dimensional attribute matrix, $\mathbf{X}$, where the attribute vectors for the members of community k were drawn from a 3-dimensional multivariate Gaussian, paramertized by $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$. For each synthetic network, we computed the entropy under our label propagation method and the corresponding p-value.

In Figure 5.4A., we plot the mean LP entropy over the $T = 1000$ samples used to construct the empirical entropy distribution, $\mathcal{E}$, across the 10 networks for each set of p_{in} and p_{out}. The shaded

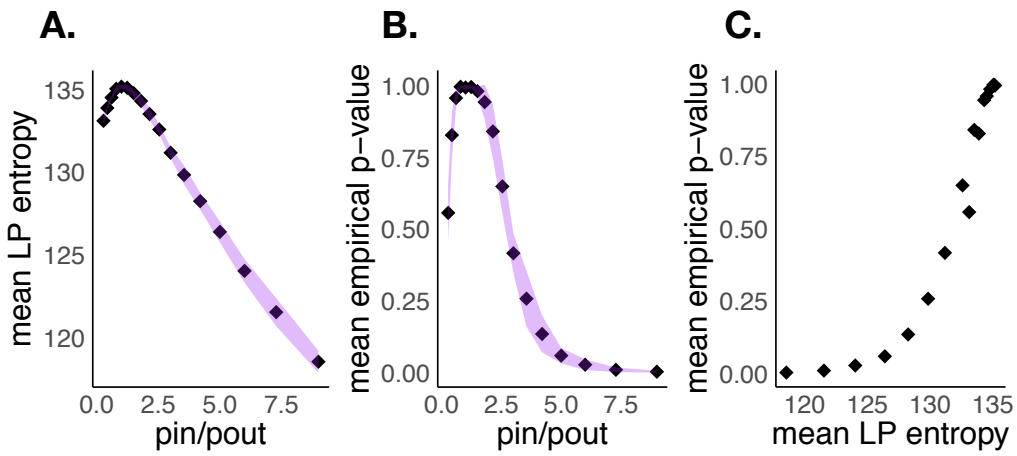

Figure 5.4: **Analysis of the strength of structural communities.** To understand the effect of network structure on our test, we generated synthetic networks from stochastic block models with various p_{in} (within-community) and p_{out} (between-community) parameters. Networks were generated with p_{in} varying between 0.05 and 0.45 and we chose a corresponding p_{out} such that the mean degree was 30. We used p_{in}/p_{out} as a proxy for the strength of community, with a higher value of this ratio indicating a stronger community structure with more within-community edges and fewer between community edges. For each p_{in}, p_{out} combination, we generated 10 synthetic network realizations. **A.** We plotted the relationship between our LP entropy and p_{in}/p_{out}. The shaded area denotes standard deviation of the mean entropy over the 10 networks for each p_{in}, p_{out} combination. **B.** Similar to (**A.**), we plotted the mean empirical p-value over the $T = 1000$ trials used to generate the entropy distributions, $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$. For large p_{in}/p_{out}, the empirical p-value became more significant. The shaded area denotes standard deviation of empirical p-value over the 10 networks for each p_{in}, p_{out} combination. **C.** Finally, we plotted the relationship between the mean entropy over the T=1000 trials, $\mathcal{E}$ and the empirical p-value. These values are strongly correlated with $r = 0.91$.

region denotes the standard deviation of the LP entropy. As the ratio between p_{in} and p_{out} increases, the empirical LP entropy decreases. We see a similar effect in Figure 5.4B., where we plot the empirical p-value as a function of the p_{in}/p_{out} ratio. In this plot, the shaded region denotes the standard deviation of the empirical p-value. Here, a significant p-value (at $\alpha = 0.05$) was reached (implying attributes and connectivity are aligned) when $p_{in}/p_{out} \approx 5$. Finally in Figure 5.4C., we examined the relationship between the mean entropy and the associated mean empirical p-value across the 10 networks generated under each parameter pair. These values were strongly correlated ($r = 0.91$).

5.3.2 Mass Cytometry Network Example

We also applied our test to a mass cytometry dataset. Mass cytometry (19) is an immunological profiling technique that gives simultaneous quantification of various immune features, such as, cell type abundances and signaling information. We used a freely available mass cytometry dataset, originally described in Ref. (149), but pre-processed in an R tool called CytofKit (32). The dataset profiles 51 immune features across single cells on human T helper (T(H)) cells from peripheral blood and tonsils, which have shown to be heterogeneous within a sample. To untangle the heterogeneity and infer cellular phenotypes, dimension reduction and clustering are applied to single cell data. In this pursuit, the objective is to cluster the single cells into predicted phenotypes, based on the measured features. A powerful way to segment the single cells into their respective phenotypes is by constructing a similarity network between the cells and clustering with community detection. This method for studying single cell data is called PhenoGraph and is described in Ref. (84). We studied the data in an analogous way by constructing a 5-nearest neighbor network and applying community detection. To do this, each cell is connected to its 5 nearest neighbors, based on euclidean distance on the pairwise Euclidean distance for the 51 features. In this particular dataset, we considered a subset of 1000 single cells and their 51 measured features. After constructing the network, we predicted phenotypes by identifying communities ($\mathbf{z}$) with the Louvain algorithm (23). Applying the Louvain algorithm identified 10 communities. As shown in (32), one further analysis after clustering the single cells is to identify marker features with discriminative power between the inferred phenotypes.

The first test we performed on the single cell mass cytometry network was to examine how each marker feature related to the community partition, $\mathbf{z}$ identified with the Louvain algorithm. To assess this first without using our method, we used each of the 51 features in isolation to generate a partition of the network, $\tilde{\mathbf{z}}$. Since this generate a single continuous feature for each node, we computed $\tilde{z}$ in each case with k-means across the 1000 cells, with 10 clusters. We chose 10 clusters so that and $\tilde{\mathbf{z}}$ would be on the same scale. Before applying our LP test to this network, we used normalized mutual information (NMI) (36) to quantify the similarity between $\mathbf{z}$ and $\tilde{\mathbf{z}}$. A high NMI (i.e. close to 1), indicates that the attribute used to create $\tilde{\mathbf{z}}$ creates a similar partition to the $\mathbf{z}$, obtained from the Louvain algorithm. Conversely, an NMI near 0 idicates that when nodes (cells) are clustered based on the particular feature, their partition is very similar to that obtained using connectivity information.

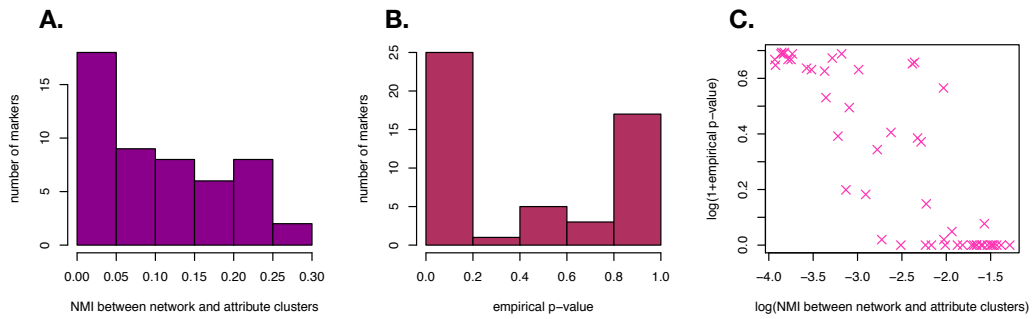

Figure 5.5: **Alignment of markers with communities.** We considered each of the possible 51 features in the single cell data and their potential to be used as markers of particular inferred cellular phenotypes. We identified 10 communities (or inferred phenotypes) under the Louvain algorithm, producing a partition of the network, $\mathbf{z}$. We then created a partition, $\tilde{\mathbf{z}}$ from each attribute in isolation. For each attribute and its induced partition of the nodes, $\tilde{\mathbf{z}}$, normalized mutual information (NMI) was used to measure the discriminative power of the marker in distinguishing network communities, or NMI($\tilde{\mathbf{z}}, \mathbf{z}$). We expected that our p-value should align with this NMI measure in that markers leading to high NMI between the induced $\tilde{\mathbf{z}}$ and $\mathbf{z}$ should have more significant p-values. **A.** We used a histogram to visualize the distribution of NMI values across the 51 possible markers, with many of them leading to low NMI (between 0 and 0.1). **B.** Similar to **A.**, we visualized the empirical p-value for the 51 possible markers. **C.** We compared the relationship between the empirical p-value (vertical axis) and NMI($\tilde{\mathbf{z}}, \mathbf{z}$) (horizontal axis) across the 51 possible markers. As expected, we observed these quantities to be anti-correlated in that more significant (lower) empirical p-values were obtained for higher values of NMI($\tilde{\mathbf{z}}, \mathbf{z}$).

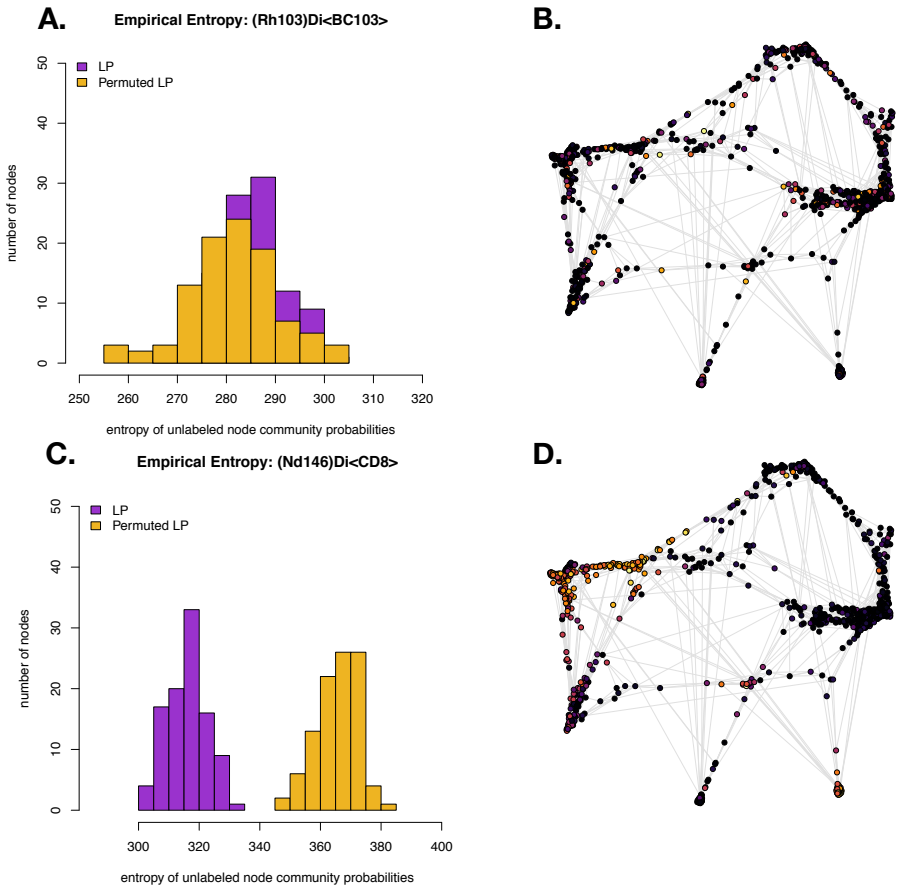

Figure 5.6: **Validation with a well and poorly aligned markers**. We used two markers with different correlation strength with communities as another validation of the computed entropy under label propagations. First, we defined a labeling of the nodes, $\tilde{\mathbf{z}}$ based on marker (Rh103)Di$<$ BC103 $>$ that did not vary across communities in its expression, and hence not discriminate between the communities. **A.** We visualized the distribution of $\mathcal{E}$ (purple), in comparison to $\mathcal{E}_{\text{perm}}$ (gold). Since this marker has low discriminative power, we expected the shown overlap between $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$. **B.** We plotted the network of the 1000 single cells and colored nodes by their expression of (Rh103)Di$<$ BC103 $>$, with lighter colors indicating higher expression. It is difficult to notice clustering in this network between cells with similar expression values. **C.** Conversely to the result shown in (**A.**), we chose a marker with high discriminative power, (Nd146)Di$<$ CD8 $>$. Again, we show the distribution of $\mathcal{E}$ (purple), in comparison to $\mathcal{E}_{\text{perm}}$ (gold). Since this marker has good discriminative power, $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$ do not overlap. **D.** We plotted the network of single cells, with nodes colored according to the intensity of (Nd146)Di$<$ CD8 $>$, with lighter colors indicating higher expression.

In Figure 5.5A. we show the distribution of NMIs computed between $\mathbf{z}$ and $\tilde{\mathbf{z}}$ for each of the 51 potential markers. We observe a fairly broad range or marker qualities represented. Similarly, we applied our LP task for $T = 500$ trials. Figure 5.5B. shows the distribution of empirical p-values from

our LP method. We noticed that there are approximately 25 markers with a low p-value (between 0 and 0.2), according to our LP test. The majority of the markers do not have significant p-values, which is also reflected in Figure 5.5A. for the markers having an NMI < 0.1. Finally, in Figure 5.5C., we examined the relationship between the NMI between $\mathbf{z}$ and $\tilde{\mathbf{z}}$ and the empirical p-value, across each of the 51 markers. As expected, these quantities are highly related, with high values of NMI corresponding to lower, more significant p-values.

To visualize how particular markers correlated with communities in the network, through their induced partition, $\tilde{\mathbf{z}}$,we chose a marker with low empirical p-value, (Nd146)Di$<$ CD8 $>$, and a marker with high empirical p-value, (Rh103)Di$<$ BC103 $>$. In Figure 5.6, we sought to visualize both the entropy distributions, $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$, as well as the communities colored by the expression of each marker. In Figure 5.6 A. and C., we show the empirical entropy distributions, $\mathcal{E}$ (purple) and $\mathcal{E}_{\text{perm}}$ (gold). For the marker with significant empirical p-value, (Nd146)Di$<$ CD8 $>$ (Figure 5.6C., the the empirical entropy distributions do not overlap. Conversely, for the poorly predicted marker, (Rh103)Di$<$ BC103 $>$, there is strong overlap between the empirical entropy distributions (Figure 5.6A.). In Figure 5.6 C. and D., we colored the communities identified by Louvain by the intensities of the markers, (Rh103)Di$<$ BC103 $>$ and (Nd146)Di$<$ CD8 $>$, respectively. Light colors indicate a high marker expression, while darker colors indicate low expression. In Figure 5.6B., where nodes are colored according to (Rh103)Di$<$ BC103 $>$, we see that nodes with high intensity expression are scattered around the network and not confided to a particular community. However, in Figure 5.6D., where nodes are colored by (Nd146)Di$<$ CD8 $>$, many high expression markers are confined to the same community. This analysis further confirms that our empirical p-value is identifying markers with discriminative power because they are well-correlated with certain communities.

As an final experiment, we sought to see if the markers with significant empirical p-values (implying that they are effective in distinguishing cellular phenotypes) did indeed vary across communities in the network, through their induced partition, $\tilde{\mathbf{z}}$. To do this, we selected 10 markers from the 51 measured features of the single-cell data. In particular, we looked at the 5 most and least significant markers, in terms of the computed empirical p-value. For each of these 10 markers, we computed the mean marker expression across each of the 10 communities, identified by applying the Louvain algorithm (23) algorithm to $\mathbf{A}$. We then plotted the mean marker expression across communities for the 5 most and least significant markers in Figure 5.7. The least significant markers

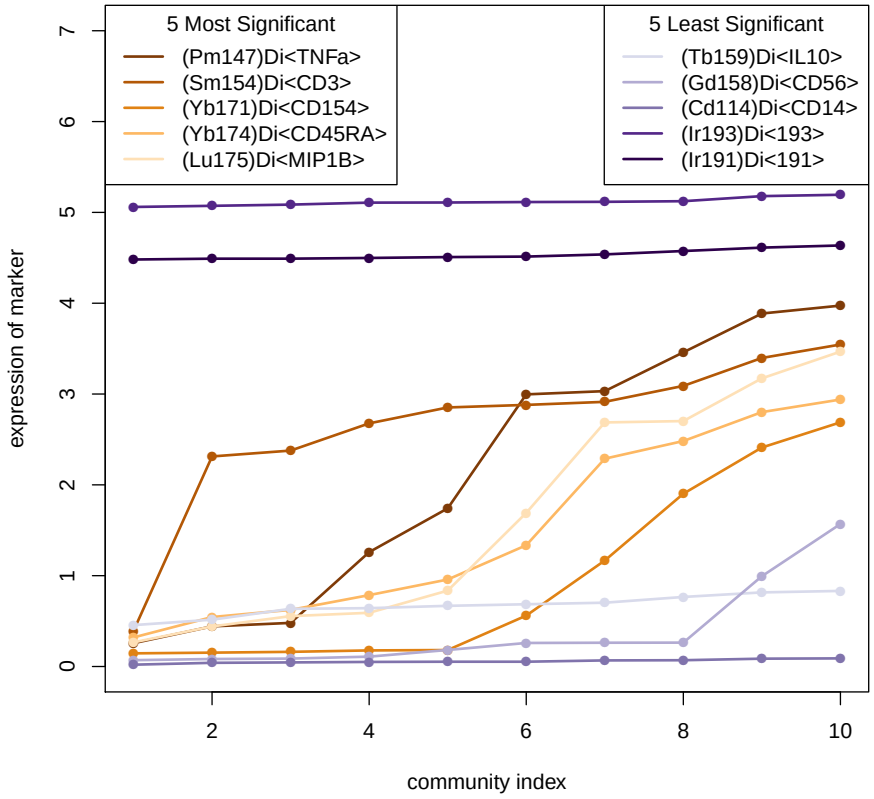

Figure 5.7: **Variation of markers with significant empirical p-values across communities**. We computed the empirical p-values induced by the partition $\tilde{\mathbf{z}}$ for each of the 51 markers and looked closely at the top and bottom 5 markers, as inferred through the empirical p-value. Since a quality marker in this case is said to be one that induces a labeled of the nodes, $\tilde{\mathbf{z}}$ similar to the result obtained under the Louvain algorithm $\mathbf{z}$, we expect the expression of such a marker to vary across communities. In this plot, we show the expression of each marker as a function of the community index. The family of orange-colored lines correspond to the top 5 predicted markers (according to empirical p-value). From all of these lines, the expression varies across communities. Conversely, we plotted the lowest-ranked markers (in terms of empirical p-value and their expression is relatively constant across all communities.

are shown in the family of blue lines and are relatively static across each of the 10 communities. In contrast, the orange family of lines corresponds to the markers for the top 5 most significant communities and do vary across communities. Since a marker with a significant low empirical p-value should correlate well with communities, this is the pattern we expected. The 5 poorly ranked markers clearly do not correlate with communities because their expression is constant across all communities.

5.4 Conclusion

In this paper, we introduced a label propagation based approach to determine how closely attributes align with network connectivity. To label propagation task seeks to predict the labels for an unlabeled set of nodes, according to the initial partition of the nodes according to the attribute information, $\tilde{\mathbf{z}}$. The label propagation task we adopt returns a probability distribution for each of the unlabeled nodes over the possible communities. The empirical p-value of our test is computed by comparing the empirical entropy distributions from our label propagation task, and a permuted label propagation task, denoted by $\mathcal{E}$ and $\mathcal{E}_{\text{perm}}$, respectively. The intuition is that if attributes are well aligned with network connectivity patterns, then the label propagation task should produce results that are more certain, and hence have lower entropy. Our results indicate that the computed entropy and empirical p-value are behaving as expected on synthetic example, where we designed the experiments in a way that we knew how well the attributes and connectivity correlated. We also show that our test is useful in the identification of important marker features for distinguishing communities in the single cell mass cytometry network. Here, features (markers) with low empirical p-value as features that vary across communities and hence give insight into what distinguishes the communities.

As future work, we can examine how the entropy and empirical p-value relate the the quality of communities identified across the variety of community detection approaches. For example, perhaps our test is more aligned with the assumptions of the stochastic block model than label propagation. Finally, similar to how we detected particular marker features that were aligned with the identified communities, perhaps we can use our tool as a feature selection method that can enable a meaningful network representation of the data.

CHAPTER 6

A network approach to understanding microbiome disruption in response to acute lung injury

This work is in collaboration with Dana Walsh.

To emphasize the usefulness of network analysis and community detection in biology, we present an application in studying the changes in the microbiome composition of patients with acute lung injury. Analogous to the discussions related to microbiome analysis earlier in this thesis, the profiling of the microbiome is accomplished through metagenomic sequencing and identifying OTUs (operational taxonomic units). Each OTU is treated as a taxonomic unit, or bacterial species and typical microbiome analyses look at the counts of a collection of OTUs across samples. In this work, we seek to study how the co-occurence patterns of OTUs differ between healthy patients and those with acute lung injury. We show that creating a network representation of this data and analyzing it with community detection is crucial for the ability to understand functional differences between two cohorts of patients.

6.1 Introduction

This study is a follow-up analysis to the work by Walsh *et al.*, which sought to profile the microbial composition of patients with acute lung injury (ALI) from smoke (146). In their work, Walsh *et al.,* show that patients with acute lung injury have enrichment for different taxa than healthy patients, however the pairwise co-occurence patterns between pairs of species were not investigated. The analysis of these co-occurence patterns are of interest because specific types of pairwise interactions between species lead to different functional outcomes (26). In these interactions, bacteria undergo processes, such as, exchanging intermediate compounds to make amino acids or lateral gene transfer. Pairwise interactions can be classified as being *mutualistic*, where all species in an interaction benefit.

Alternatively, the interaction can be *antagonistic*, where some species benefit at the expense of others (83). The identification of important pairwise interactions between taxa offer the opportunity for therapeutic intervention.

6.1.1 Data Background

DNA was extracted from bronchial washings of 48 patients after 72 hours of hospitalization for burn and inhalation injury in the North Carolina Jaycee Burn Center. Patients were classified as having burn inhalation injury as hypoxemia according to the ratio of the partial pressure of arterial oxygen (denoted as 'P') to the fraction of inspired oxygen (denoted as 'F') ≤ 300. A. patient with a P/F ratio > 300 was assumed to have normal oxygenation levels. Those with P/F < 300 were classified as having acute lung injury (ALI). DNA was sequenced using the MiSeq (30) platform and bacterial species were identified using the MT-Toolbox pipeline (157). The output of MT-Toolbox was an OTU count table, profiling the abundances or counts of 372 OTUs for each of the 48 patients.

6.2 Network Analysis Methods

6.2.1 Creating Networks with SparCC

Given the OTU count table that we generated, the next objective was to construct two networks. The first network was the OTU co-occurence network constructed from the 24 patients with acute lung injury. The other network is the OTU co-occurence network between healthy patients. To construct these networks, we used SparCC (53), the method discussed in Chapter 2. As a recap, the objective of this method is to create sparse correlation networks between OTUs, based on their counts. Due to the count-based nature of the data, SparCC is a more appropriate approach for constructing correlation networks than Pearson correlation as it does not lead to as many spurious connections. The authors further point out that spurious correlations are often worse when the diversity of the sample (defined as some function of the number of OTUs present) is low. Hence, SparCC is a state-of-the-art method because it also takes this diversity issue into account. The correlation network in each case is between OTUs, according to their co-occurence patterns across patients in the associated cohort (healthy vs. acute lung injury).

The correlation networks returned by SparCC had both positive and negative edge weights. In

this analysis, we only consider positive edges, as most of the the community detection literature is not amenable to signed networks. We were further curious to see how we could more closely hone in on the important structures by thresholding the network, or removing edges less than some threshold weight. We based this threshold on two criteria. First, we sought to find a stable threshold where slight variations in threshold would not dramatically change the number of communities detected with a modularity-based community detection algorithm. Next, we also sought to identify the threshold producing a node-to-community partition similar to the results produced at an adjacent threshold. In summary, both of these methods seek to find a stable threshold that does not dramatically change the community structure. For both networks, this threshold turned out to be 0.14. In other words, we discarded edges with a weight less than 0.14. This produced 4 communities (identified through modularity optimization) in each network. We show the acute lung injury and healthy OTU co-occurence networks (left and right, respectively) in Figure 6.1. Here, nodes are colored by their community assignment. From even this early glance, we observe that the structures of these networks are quite different. This observation allowed us to further analyze the difference in biological function reflected in these different network structures.

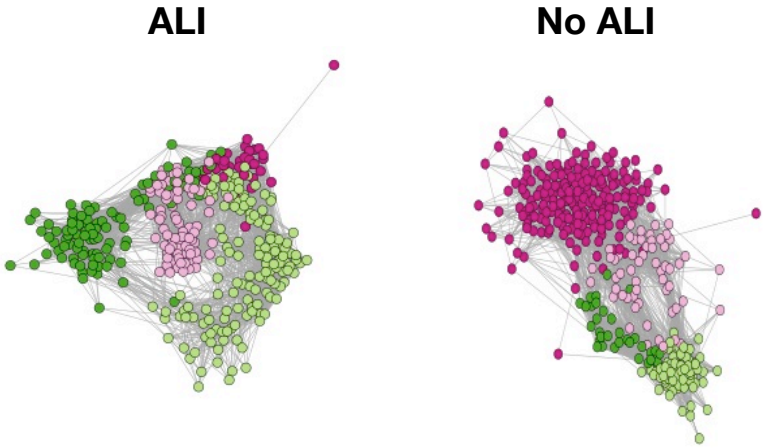

Figure 6.1: **Microbial co-occurence networks for each patient cohort**. We constructed networks with SparCC in the ALI and non-ALI cohort networks (left and right, respectively). Four communities were identified in each network. Nodes are colored by their community assignment.

	No ALI A	No ALI B	No ALI C	No ALI D
ALI 1	1	1	27	2
ALI 2	50	5	9	22
ALI 3	66	46	37	5
ALI 4	77	5	15	4

Table 6.1: **Comparing Networks in Each Patient Cohort**. We compare the OTUs in each pair of communities in the ALI and No ALI cohort networks. Large overlaps are denoted by pink shading in the table.

6.3 Results

6.3.1 Community overlap between network

After constructing the network for each cohort, we first evaluated the similarity in all pairs of communities across both networks, and used bioinformatics tools to further uncover the biological differences. We denote the communities in the ALI network by ALI 1, ALI 2, ALI 3, and ALI 4. Similarly, we denote the four communities in the No ALI network by No ALI 1, No ALI 2, No ALI 3, and No ALI 4. In table 6.1, we show the contingency table used to compare the communities in the two networks. Each entry counts the number of OTUs shared between the community pair. We denoted the large overlaps (i.e. sharing many common OTUs) by pink shading in the table. In particular, we highlight the similarity between ALI 4 and No ALI A, ALI 3 and No ALI B, ALI 1 and No ALI C, and ALI 2 and No ALI D.

6.3.2 Evaluating functional differences

Next, we sought to study functional differences in the airway microbiota between patients with and without acute lung injury. In other words, each of the OTUs contains different genes, which leads to different functions (i.e. many OTUs contain genes that encode glycoside hydrolase activity). Moreover, we hypothesized that there would be a difference in the functions of the communities between the ALI and non ALI networks. To investigate this, we used PICRUSt (78), a bioinformatic approach used to predict the function of each community. PICRUSt works by looking at each community and the known genetic information about the OTUs assigned to that community and determining the enrichment of particular functions.

6.3.3 Classifying each community according to predicted function

PICRUSt originally returned 6,911 unique functions according to the communities across both networks. We were interested to see if we could train a classifier to predict an OTUs community assignment based on its inferred function, according to PICRUSt. In other words, each OTU has several predicted functions, according to its genetic content and we wished to test if we could predict an OTUs community assignment in each network based on the presence or absence of certain functions. From the 6,911 features returned by PICRUSt, we reduced the set by filtering out the functions that were not associated with any OTU. We further filtered out functions if they had a small ratio of within-class variance to between-class variance, meaning that we only wanted features that varied between classes. Using a ratio threshold of 0.05 brought the number of features in our model to 328. A random forest model was trained with half of the data for the ALI and No ALI datasets independently, using the 328 PICRUSt functions as the features. To visualize which features were most predictive in being able to classify an OTU into a community, based on function, we measured its importance their importance based on their Gini importance (91). In Figure 6.2, the biological functions are presented from top to bottom for the ALI and No ALI networks (left and right, respectively) in terms of their Gini score.

When these highly ranked predictors were compared between the ALI and no ALI networks, there was very little observed overlap between the highly ranked features between the ALI and no ALI networks, which suggested that acute lung injury is severely altering the microbiome and its function.

6.4 Discussion

In this chapter we presented preliminary work toward a better understand the disruption and evolution to the microbiome in response to acute lung injury. Recognizing the importance of not just measuring the abundances of OTUs in a sample, but also their co-occuring species, offers more in-depth insight into the underlying biological processes and functions. Constructing networks with SparCC for each of the ALI and No ALI cohorts allows us to probe the structural organization of these two different networks. The PICRUSt bionformatics tool and classification task provided important insight into

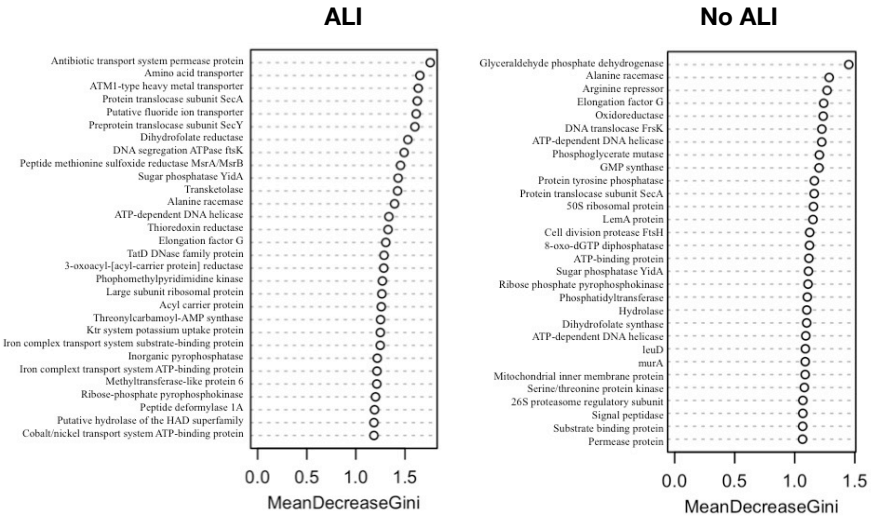

Figure 6.2: **Predictive functions for community classification**. We used a set of 328 filtered functions to predict OTU-to-community assignment in the ALI and No ALI networks. Here we show show the functions identified as the most strong predictors for each community in the ALI and No ALI networks (left and right, respectively). Functions with more discriminative ability in classification from the random forest classifier are ranked higher on the list.

the biological functions that are most indicative of each community in the ALI and No ALI networks. As shown in Figure 6.2, the antibiotic transport system permease protein was ranked to be the most predictive in terms of classifying the OTUs into communities in the ALI network. This might suggest that patients with ALI are better able to transport antibiotics out of the cell, making them more resistant to treatment. This observation suggests that future work should be done to determine if *Prevotella melaninogenica*, an enriched species shown in the initial study by Walsh *et al.* (146), contains the antibiotic transport system permease protein and gives bacteria an advantage in resisting antibiotic treatment and remaining in the airways of patients with ALI.

Overall, this work prioritized microbial interactions and potential biological mechanisms (specifically antibiotic resistance) that can be further investigated in the lab to understand the implications of acute lung injury on the microbial ecosystem in airways.

CHAPTER 7

Conclusion and Future Work

In this thesis, we presented four new methods that extend the capabilities of community detection in 3 types of challenging network data. In particular, we focused on multilayer networks (Chapter 2), large networks (Chapter 3) and attributed networks (Chapters 4 & 5). In this section, we will recap each of the developed methods, explain possible extensions to the work, and describe classes of problems or applications that may benefit from such an approach.

7.1 Strata Multilayer Stochastic Block Model

7.1.1 Recap

We first presented the strata multilayer stochastic block model (sMLSBM), an extension to the standard stochastic block model. In sMLSBM, the objective is to learn a collection of models that have most likely generated the layers in a multilayer network. Seminal approaches to fit an SBM to a multilayer network used all of the layers together to learn a single stochastic block model to describe a multilayer network. Moreover, our approach is novel because it assumes that there are potentially multiple stochastic block models that have given rise to the collection of networks. By learning a set of stochastic block models, with each model corresponding to a 'stratum', we have the ability to cluster networks and to understand the generative process that gave rise to each layer. Furthermore, we also gain insight into the generative process describing each stratum. We have shown this is useful for characterizing the networks across body sites from the Human Microbiome Project (shown in Figure 2.6). From this figure, it is clear that the community structure is distinct across strata and fitting the same model to all of the layers together would not provide a meaningful summary of the data. We can summarize the implications of this work as follows. First, sMLSBM is a clustering and

an SBM parameter learning method that provides insight into the differences among the individual network layers in a multilayer network. Within each stratum, we show that the SBM fit with the collective information of all networks in the stratum can be sampled to understand a representative connectivity pattern for members of that cluster. Finally, this work inspired the investigation of the detectability limits of the stochastic block model and how to most effectively combine the network layers within a stratum to more accurately identify community structure.

7.1.2 Future Work

The first two major limitations of sMLSBM is that it exists for handling unweighted networks and that fitting the model requires the specification of the number of strata, S. In reality, most networks are weighted, and weights are an important aspect of structural organization. As we have alluded to, the literature for dealing with weighted stochastic block models is still in its primitive phase, as it is not immediately clear how to handle edge weights in a statistically principled way. With increasing advances in the development of the weighted stochastic block model, and the ubiquitous presence of weighted network data, the ability to extend sMSLBM to weighted networks could be valuable.

Second, in all of the examples in this paper, we always had a rough idea about how many strata, S to use. In reality, if you are looking to cluster your networks and figure out how many stochastic block models describe your multilayer network, you might not know which S to choose up front. To temporarily mediate this issue, there are a couple of hacks that can be used. First, an SBM can be fit to each network layer individually and the corresponding adjacency probability matrices can be clustered with k-means, where the number of clusters can be specified with one of the many model selection criteria. Second, the adjacency probability matrices can be visualized using tSNE or PCA to guess how many clusters of networks exist in the data. Finally, since fitting sMLSBM is done in a way that maximizes likelihood, one could always compute fit multiple models with different S and compare the likelihoods of the fitted models.

One interesting application of multilayer networks in in the analysis of temporal network data, where each network layer corresponds to a point in time over the considered time series. It would be interesting to incorporate this time element in determining each stratum. That is, one can assume that network layers that are closer together in the temporal trajectory should be more likely to be members of the same stratum. If this were the case and strata were composed only of network layers

that appeared sequentially in time. The breaks between these temporally-contiguous sets of networks is known as change point detection and has been previously explored for probabilistic hierarchical networks by Peel *et al.,* (110).

Finally, sMLSBM currently operates on an assumption that each network layer belongs to a specific cluster. Given the great success in mixture models, perhaps it could be useful to model each network's membership to a cluster with a probability distribution, rather than a binary classification.

7.2 Super Nodes

7.2.1 Recap

Next, we introduced super nodes, which is a pre-processing step for large networks before applying community detection. We discusses that the benefits of a super node representation of the network is a decreased run time of community detection algorithms, decreased variability among outputs of the community detection methods, and more consistent labeling within a local region of the network (i.e. less label entropy within a neighborhood). We demonstrated our approach on several large social network datasets. Finally, we discussed the important issue in the development of community detection approaches that results are difficult to validate without ground truth. To quantify the quality of our super node pre-processing, we adapted normalized mutual information (NMI) and under segmentation error to this context.

Some limitations of this work is that it currently exists only for unweighted networks and we don't have evidence that the seeds chosen are the absolute best super node centers. To address both of these issues, it might be useful to use random walks to choose seeds and grow out super nodes. In choosing seeds, for example, a number of random walks can be started from a large number of randomly selected nodes. The seeds of our super nodes could be the set of nodes that many random walks pass through. Furthermore, the growing out process for the super nodes could assign the non-seed nodes to the seed contained in the majority of their random walks. Random walks are a powerful tool to compress networks due to their relationship with the graph Laplacian matrix (31; 92).

7.2.2 Future Work

Since we have explored attributed network data in this thesis, it may also be interesting to determine how attributes can be incorporated in defining a super node. This is briefly considered in References (156) and (86), which require some notion of node attributes. However, perhaps we can incorporate node information in our growing out step.

Given that one appeal of the super node representation of the network is that it reduces the large network to a much smaller size, it would be useful to figure out how to exploit this information for visualization, network summarization, and prediction tasks. More work should be done to determine how the summary statistics of the original network are carried over to the super node representation. For example, how does a quantity like modularity compare between a network and its super node representation? For prediction tasks such as link prediction, or collaborative filtering, it would be interesting to probe how information within and between super nodes can be used to maximize predictive accuracy.

7.3 Stochastic Block Models with Multiple Continuous Attributes

7.3.1 Recap

Finally, we presented an extension of the stochastic block model to handle networks, where the nodes have multiple continuous attributes. This work addresses the current limitations of some of the primitive developments in the attributed stochastic block models that are more inclined toward discrete or scalar attributes. To validate our model and inference approach, we show that the learned model can be used and obtain good performance on link prediction and collaborative filtering tasks. In particular, we demonstrate successful performance of these two tasks with the learned attributed SBM on a microbiome subject-similarity network and on a protein interaction network.

7.3.2 Future Work

Similar to the limitations of sMLSBM and super nodes, this method is also only applicable to unweighted networks and it is required to specify the number of communities expected to find. While the combination of attributes and connectivity in a weighted network is likely a challenging task, it

would be useful to develop a model selection criteria to decide how many communities to find. Next, the examples we have shown have included a few continuous attributes, but it might become difficult to estimate the covariance matrix as the number of attributes becomes large. The limitations of this should explored and it should also be investigated. Perhaps lower dimensional representations of the attributes could be a solution to this problem. Finally, we make a strong assumption in our model that connectivity and attributes are conditionally independent, given the class labels. Perhaps we can do more work in the future to determine how realistic this assumption is for a dataset of interest.

We also briefly alluded to the detectability limits of stochastic block models and how adding attributes that are principled (or aligned with the node-to-community assignments) can affect the ability to find communities. This question can be explored more rigorously and formally.

Finally, in the introduction, we introduced the affiliation model as a flexible probabilistic approach to model connections between nodes according to their community affiliations. Perhaps adapting the affiliation model to handle multiple continuous attributes could provide a more flexible notion of an attributed community.

7.4 Testing Alignment of Attributes and Connectivity

7.4.1 Recap

After introducing a stochastic block model for attributed networks, we sought to develop a test to quantitatively measure how closely attributes and connectivity align. Our test first partitions nodes into communities according to attribute information and then uses a sequence of label propagation tasks to predict node-to-community assignment probabilities. From these node-to-community assignment probabilities computed over the sequence of label propagation tasks, an empirical entropy distribution is constructed to reflect the uncertainty of these assignments. We then use a permutation test to construct an empirical p-value as a measure of the significance of the attribute and connectivity alignment. The intuition is that if the attributes and connectivity are well aligned, then the node-to-community assignment probabilities should be highly certain. High certainty in this probability distribution, or a clear propensity for assignment in a particular community corresponds to low entropy. We applied our test to several synthetic examples, including to BESTest, a related method.

Finally, by applying our method to a single cell mass cytometry dataset, we show that our method can be used for the identification of discriminative features that vary across communities.

7.4.2 Future Work

As we discussed in this thesis, there are numerous community detection approaches that can be augmented to incorporate attribute information. In the context of the empirical p-value computed by our test, it could be interesting to compare the results across attributed community detection methods. For example, we could consider 3 possible partitions of the nodes. We could either consider the partition of nodes according to connectivity information, attribute information, or both pieces of information together through an attributed community detection algorithm. From our empirical p-value, we gain insight into how closely attributed and connectivity align. We could expect that in cases with a low (significant) empirical p-value, the similarity between the partitions of the nodes with the connectivity and attribute information independently should be similar to the partition under the attributed community detection algorithm. Such analysis could provide insight into which attributed community detection approach can most effectively integrate both of these sources of information.

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. (2015). 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.

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831.

Attias, H. (2000). A variational baysian 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.

Bassett, D. S., Wymbs, N. F., Porter, M. A., Mucha, P. J., Carlson, J. M., and Grafton, S. T. (2011). Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 108(18):7641–7646.

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.

Benson, A. R., Gleich, D. F., and Leskovec, J. (2016). Higher-order organization of complex networks. *Science*, 353(6295):163–166.

Betzel, R. F., Medaglia, J. D., and Bassett, D. S. (2018). Diversity of meso-scale architecture in human and non-human connectomes. *Nature Communications*, 9(1):346.

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.

Boon, E., Meehan, C. J., Whidden, C., Wong, D. H.-J., Langille, M. G., and Beiko, R. G. (2014). Interactions in the microbiome: communities of organisms and communities of genes. *FEMS microbiology reviews*, 38(1):90–118.

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.

Caporaso, J. G., Lauber, C. L., Walters, W. A., Berg-Lyons, D., Huntley, J., Fierer, N., Owens, S. M., Betley, J., Fraser, L., Bauer, M., et al. (2012). Ultra-high-throughput microbial community analysis on the illumina hiseq and miseq platforms. *The ISME journal*, 6(8):1621.

Chaudhuri, K., Chung, F., and Tsiatas, A. (2012). Spectral clustering of graphs with general degrees in the extended planted partition model. In *Conference on Learning Theory*, pages 35–1.

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.

Davis, M. M., Tato, C. M., and Furman, D. (2017). Systems immunology: just getting started. *Nature immunology*, 18(7):725.

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.

Feld, S. L. (1981). The focused organization of social ties. *American journal of sociology*, 86(5):1015–1035.

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.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

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.

Langille, M. G., Zaneveld, J., Caporaso, J. G., McDonald, D., Knights, D., Reyes, J. A., Clemente, J. C., Burkepile, D. E., Thurber, R. L. V., Knight, R., et al. (2013). Predictive functional profiling of microbial communities using 16s rrna marker gene sequences. *Nature biotechnology*, 31(9):814.

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.

Leung, T. and Poulin, R. (2008). Parasitism, commensalism, and mutualism: exploring the many shades of symbioses. *Vie et Milieu*, 58(2):107.

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.

Libbrecht, M. W. and Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321.

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.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

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.

Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., and Hamprecht, F. A. (2009). A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10(1):213.

Merris, R. (1994). Laplacian matrices of graphs: a survey. *Linear algebra and its applications*, 197:143–176.

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. (2010a). Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010b). 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. and Clauset, A. (2015). Detecting change points in the large-scale structure of evolving networks. In *AAAI*, pages 2914–2920.

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*.

Perfetto, S. P., Chattopadhyay, P. K., and Roederer, M. (2004). Seventeen-colour flow cytometry: unravelling the immune system. *Nature Reviews Immunology*, 4(8):648.

Perozzi, B. and Akoglu, L. (2018). Discovering communities and anomalies in attributed graphs: Interactive visual exploration and summarization. *ACM Trans. Knowl. Discov. Data*, 12(2):24:1–24:40.

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.

Shreiner, A. B., Kao, J. Y., and Young, V. B. (2015). The gut microbiome in health and in disease. *Current opinion in gastroenterology*, 31(1):69.

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., Frost, C., Mucha, P. J., and Porter, M. A. (2009). Visualization of communities in networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 19(4):041104.

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. (2016). Multilayer stochastic block models reveal the multilayer structure of complex networks. *Physical Review X*, 6(1):011036.

van den Heuvel, M. P. and Sporns, O. (2013). Network hubs in the human brain. *Trends in cognitive sciences*, 17(12):683–696.

Walsh, D. M., McCullough, S. D., Yourstone, S., Jones, S. W., Cairns, B. A., Jones, C. D., Jaspers, I., and Diaz-Sanchez, D. (2017). Alterations in airway microbiota in patients with pao2/fio2 ratio? 300 after burn and inhalation injury. *PloS one*, 12(3):e0173848.

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.

Wong, M. T., Chen, J., Narayanan, S., Lin, W., Anicete, R., Kiaang, H. T. K., De Lafaille, M. A. C., Poidinger, M., and Newell, E. W. (2015). Mapping the diversity of follicular helper t cells in human blood and tonsils using high-dimensional mass cytometry analysis. *Cell reports*, 11(11):1822–1833.

Xiang, T. and Gong, S. (2008). Spectral clustering with eigenvector selection. *Pattern Recognition*, 41(3):1012–1029.

Xie, J. and Szymanski, B. K. (2011). Community detection using a neighborhood strength driven label propagation algorithm. In *Network Science Workshop (NSW), 2011 IEEE*, pages 188–195. IEEE.

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. (2013). Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM.

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.

Yourstone, S. M., Lundberg, D. S., Dangl, J. L., and Jones, C. D. (2014). Mt-toolbox: improved amplicon sequencing using molecule tags. *BMC bioinformatics*, 15(1):284.

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.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation.

Zitnik, M. and Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.