THE TITLE OF YOUR THESIS:
USE LINE BREAKS IF NEEDED

Your M. Name

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 Department of Computer Science.

Chapel Hill
201X

Approved by:

Committee Member 1

Committee Member 2

Committee Member 3

Committee Member 4

Committee Member 5

Committee Member 6

ABSTRACT

YOUR M. NAME: Your Title in Title Font, but not in all Caps
(Under the direction of Your Boss)

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

Dedication…

ACKNOWLEDGEMENTS

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

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

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

CHAPTER 1

Introduction

Network data appears widely across fields as a data structure for modeling relational information between a set of entities. In recent years, networks have become an indispensable data mining tool, as they allow for tasks such as, data visualization, clustering, and predictive modeling. Motivated by problems in fields, such as, biology, medicine, neuroscience, social science, and epidemiology, the field of network analysis has gained popularity and seeks to develop tools for understanding the associated network data. The development of these tools is rooted in a combination of techniques from statistics, computer science, physics, and mathematics. In this thesis, we will provide a comprehensive overview of networks and analysis techniques and introduce three new models/methods that will expand the types of network data that we asre able to collect and interpret.

1.1 Network Notation,Representation, and Summarization

1.1.1 Representing relational information

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

Such a network with edges simply representing whether or not a pair of nodes interact is an example of an *undirected,unweighted* network. We will use an undirected network to introduce two forms of representations for networks. For a set of N nodes, we define the $N \times N$ network adjacency

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

matrix, $\mathbf{A} = \{a_{ij}\}$. For a pair of nodes i and j, its corresponding adjacency matrix entry a_{ij} is defined as follows,

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

.

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

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

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.

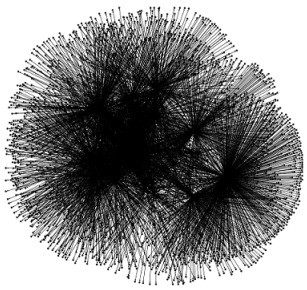

Figure 1.2: **Hairball network.** Networks are often noisy data structures and lack an immediate straight forward structural interpretation. Image from `https://cs.umd.edu`

1.1.2 Network Summary Statistics

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

1.1.2.1 Building a network to illustrate summary statistics

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

To illustrate these quantities, we will compute them on the network shown in Figure 1.3. This network is constructed from a single cell mass cytometry dataset released publicly in the Cytofkit R package (Chen et al., 2016). Each node represents a single cell and is represented with 50 features profiled with mass cytometry. We created a network from this dataset by selecting 500 cells and building a k-nearest neighbor network with $k = 5$. This means that for a node i, we found its 5 nearest neighbors according to Euclidean distance, and connected them all to node i.

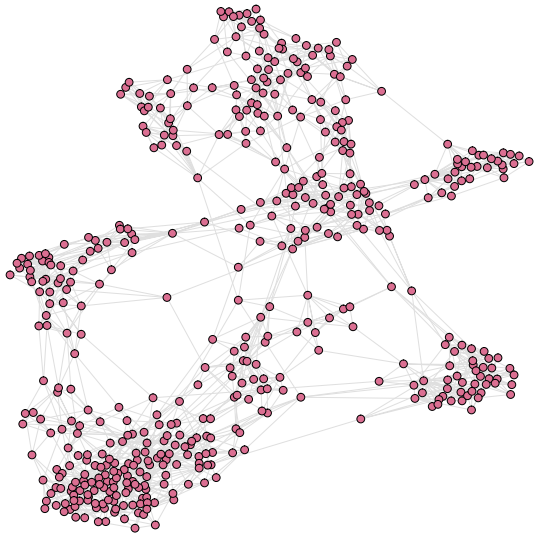

Figure 1.3: **Network of single cells.** We constructed a network from mass cytometry profiling in single cell data. Each node is a single cell and is connected to its 5 nearest neighbors.

1.1.2.2 Degree Distribution

The first most basic summary statistic is known as *degree*. Here, we will define a variety of summary statistics and quantities that can be computed on a network that give insight into the network's structure. Given the adjacency matrix for an undirected network, $\mathbf{A}$, the degree of node i, degree(i) is computed as,

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

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

In the case of an undirected, unweighted network, the degree of node i counts its number of neighbors, while in the undirected, weighted context, degree encodes the total edge weight incident to node i. Collectively examining the distribution of degrees for a network is known as the *degree distribution*. Understanding the degree distribution provides insight into the network type and

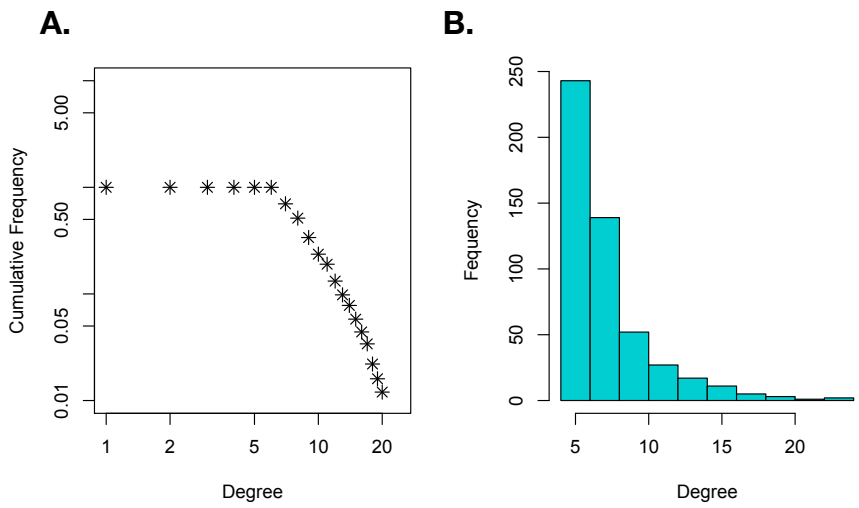

Figure 1.4: **Degree distribution for single cell network.** We visualize the trends in node degree in the single cell network presented in Figure 1.3. **A.** We compute a cumulative distribution plot for degree. **B.** Node degrees can also be visualized with a simple histogram.

structural organization. A node's degree is often highly related to its importance in the network, which provides a nice transition to the next set of summary statistics: centrality and hub scores.

1.1.2.3 Centrality and network hubs

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

Degree centrality

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

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

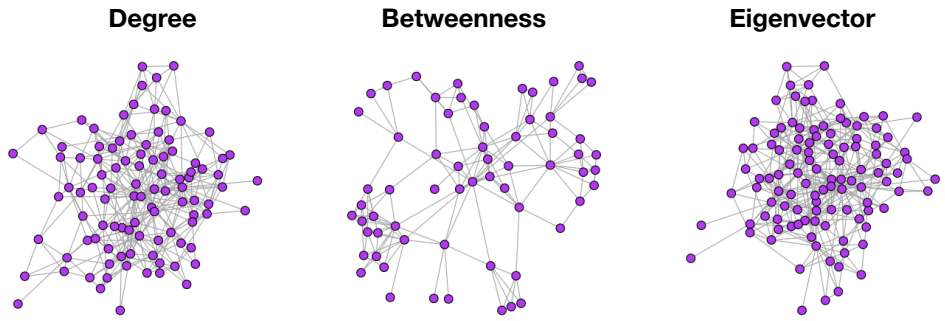

Figure 1.5: **Centralities on single cell network.** The second order ego network for the highest centrality nodes in the single cell network according to degree, betweenness, and eigenvector in the left, right, and center, respectively.

Betweenness centrality

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

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

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

Eigenvector centrality

The idea behind eigenvector centrality is that a node should be prioritized not only based on its degree, but the degree of the nodes it connects to. That is, a node connected to other 'important' or high degree nodes should be ranked higher than one connected to many low degree nodes. The eigenvector centrality for node i, can be computed using the spectra of the adjacency matrix, $\boldsymbol{A}$. 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

in comparison to the rest of the nodes. Moreover, the i-th entry of $\mathbf{x}$ gives the eigenvector centrality for node i.

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

1.2 Conceptual Overview of Community Detection

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

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

Community detection is a well-studied sub-domain of network science. The interested reader can refer to one of the comprehensive review articles (Lancichinetti and Fortunato, 2009; Fortunato and Hric, 2016; Shai et al., 2017)

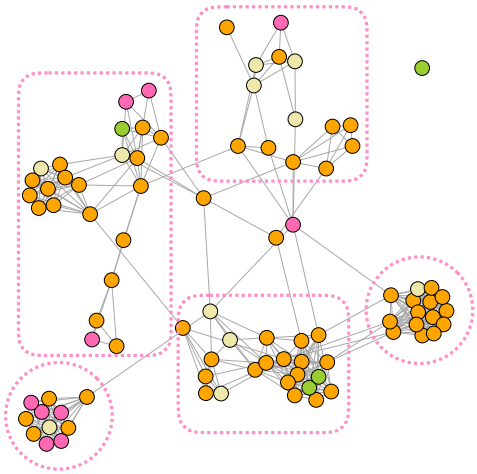

Figure 1.6: **Assortative Community Structure.** Nodes are tightly connected to each other and more sparsely connected to the rest of the network. Each community is outlined with a pink dotted line.

1.3 Introduction to community detection

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

1.3.1 Quality function maximization with modularity

For quality function optimization, one writes down a quantity to optimize that seeks to identify a partition of the network into nodes that is representative of the network structure. The most common quality function for this task is known as modularity (Newman, 2006a). Intuitively, modularity defines a null model for network that doesn't have prominent organizational structure. In particular, this null model is a random graph model, known as the configuration model (Bender and Canfield, 1978). 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 crreated 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 number of edges for nodes i and j, respectively, and M is the total number of edges in the network.

Modularity was introduced in 2004 by Newman and Girvan (Newman and Girvan, 2004). 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 (Reichardt and Bornholdt, 2006) that controls the scale of community size. Large values of γ favor more small communities while smaller value enforce for fewer large communities.

In order to determine $\mathbf{z}$, the most computationally efficient approach is known as the Louvain algorithm (Blondel et al., 2008). The Louvain algorithm is an agglomerative heuristic, which initially starts with each node in its own community and in the first match merges pairs of nodes if their merge leads to an increase in modularity. Each group of nodes assembled after this first pass becomes a new node in the network and a new weighted network is created between the set of new nodes. The weight on the edges of the new network are the number of edges from the original network that go between the sets of merged nodes. This process is continues iteratively until the modularity no longer increases. The reason that this approach is so computationally tractable is because the gain in modularity, ΔQ of merging two groups of nodes can be explicitly computed in closed form.

Modularity has shown to be effective in applications from neuroscience (Meunier et al., 2009) to image segmentation (Browet et al., 2011). It has also shown to be effective in clustering high dimensional data that has been used to create a network. In 1.7, we used tSNE **??** to project the 50-dimensional single cell data into 2 dimensions. Points are colored by their cluster assignment according to k- We first performed k-means on the original 50 dimensional data (left) and Louvain community detection on the 5 nearest neighbor network representation (right). One benefit of the

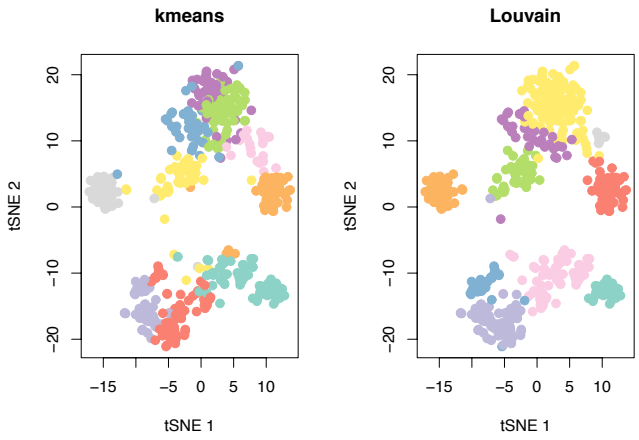

Figure 1.7: **A comparison of k-means and the Louvain algorithm** A comparison of the results of clustering the single cell dataset by visualizing the original 50 dimensional data with a 2-dimensional projection with tSNE. Points are colored by their cluster membership under k-means on the original data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

Louvain algorithm is that it does not require specifying the number of cluster. Moreover, in this example, the Louvain algorithm maximized modularity by partitioning the network into 10 clusters. Moreover, we also clustered the original data with 10 clusters. From these two partitions, we observe that creating a network representation of the data before clustering assists in identifying smaller, less prominent clusters.

1.3.2 Identifying communities with probabilistic approaches

This approach will be only briefly introduced here, as it will be explored more in depth in subsequent chapters. Probabilistic community detection methods aim to find a partition of the network through likelihood optimization. Intuitively, the goal is to study the generative process of the node edges in terms of inferred community assignments. For example, given nodes i and j, one may model $P(a_{ij} = 1)$ as $g(z_i, z_j)$, where $g(\cdot)$ is some rule based on the node-to-community assignments. Two common probabilistic community detection models are the stochastic block model (Snijders and Nowicki, 1997a) and the affiliation model (Yang and Leskovec, 2012). The definition and description of these models and inference techniques are described in depth in chapter 2.

1.3.3 Deep Learning Approaches

In recent years, deep learning has begun to revolutionize many fields, including network analysis. Perozzi *et al.*, pioneered the use of deep learning in community detection with the development of DEEPWALK (Perozzi et al., 2014) to learn a latent space representation of nodes in a lower dimensional space (i.e. an emedding). Once the network is embedded in a lower dimensional space, simple clustering techniques, such as k-means (Hartigan and Wong, 1979) 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 (Noh and Rieger, 2004; Gleich, 2015). A random walk on a network involves choosing a starting node and traversing the network by hopping between adjacent nodes. The DEEPWALK approach seeks to learn an embedding of the nodes that preserves the sets of nodes traversed in a random walk. To do this, the authors used Word2Vec, a tool from natural language understanding that allow for the specification of a node embedding that enable accurate prediction of a word's context, given the word (Mikolov et al., 2013). To adapt this context to networks, a random walk is treated as a sentence and nodes are treated as a word within the sentence. Moreover, the analogous task to the problem in text data to a network is to accurately assign a probability predict a set of nodes likely to be seen with the node of interest. Moreover, this problem is solved using the same optimization approach as Word2Vec

Based on the success of DEEPWALK, the method was followed up with Node2Vec in 2016 (Grover and Leskovec, 2016). 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 homohpily (Kossinets and Watts, 2009). 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 (Lorrain and White, 1971). For example, a grouping of nodes that take into account their degree, with the community assignments being highly related to node degree. To modify the random walk so that it leads to a model that gives flexibility in the nature of retrieved communities, the authors introduced a search bias term, which controls whether the random walk 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 source, the resulting community aligns with the homophily hyptohesis. A random walk performed in a breadth first manner that restricts the path to nodes neighboring the source and tends to capture nodes based on structural equivalence (i.e. a hub, or highly connected node).

1.3.4 Spectral community detection methods

1.3.5 Higher order network analysis

1.4 Community detection in computational biology

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

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

1.4.1 Immunological profiling to establish a pregnancy immune clock

A study lead by Aghaeepour *et al.*, demonstrated that there is a typical timing of immunological events in a healthy, term, human pregnancy (Aghaeepour et al., 2017). Immunological profiling was performed on a training cohort of 18 women, using a technology called mass cytometry (Bendall et al., 2012) was used to quantify various features of the immune system, such as, cell type abundances, signaling activity. From this set of measured immune features, a correlation network from the training cohort to identify which immune features were potentially related or working

together. Simultaneously, a regression model was training to identify immune features associated with increased gestational age. When communities were identified in the network of immune features, there were two important observations. First, immune features of the same type (i.e. cell signaling vs. cell frequency) were aligned with community labels. Second, sets of features associated with a particular gestation age often fell in the same community, indicating their synchronous activity during the pregnancy. Finally, after identifying influential nodes in their ability to predict stage in pregnancy, according to the regression model, the communities of these nodes were more closely examined to uncover further insight into the immunological mechanisms occuring throughout the pregnancy time course.

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

The microbiome refers to the collection of bacterial species that populate an organism's gut. Microbiome analysis has recently gained attention, as its biological implications are large for health and disease. A 2017 review article presented the idea that the development of network analysis approaches for microbiome data is under explored and has great potential for advancing biological understanding and interpretation of these data (Layeghifard et al., 2017). 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 (Baldassano and Bassett, 2016). Communities were identified in the healthy and diseased networks, using classic modularity maximization (Girvan and Newman, 2002). After identifying a community structure for each network, the similarity of these partitions was quantified with the Rand index (Traud et al., 2011), which showed to be statistically significant under a permutation test. This observation allowed the authors to understand that the core structure from a healthy microbiome was conserved even in diseased patients, but allowed for more careful probing of the subtle differences. First, the functional roles of the members of each community were interrogated. Some interesting co-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 (van den Heuvel and Sporns, 2013). 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. 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 (Aghaeepour et al., 2013). A network-based approach to this problem, known as SamSPECTRAL was introduced in 2010 by Zare *et al.* (Zare et al., 2010). In this approach, the authors seek to segment a population of cells into distinct populations of cells, through the

construction of a similarity network and identification of communities. In this network, the nodes are comprised of the cell types in a sample, and edges between nodes indicate the similarity between a pair of cells, based on the quantification of their emitted light. Because a high throughput biological sample could contain as many as biological points, this approach seeks to first create a smaller representation of the data, build a network on this smaller version, and ultimately segment the data this way.

To create a network of the flow cytometry data, a large subset of data points (cells) are first sampled and denoted as 'registered' nodes. The next step is to look at the collection of 'unregistered nodes' and ultimately assign them to their closed registered node neighbor. Iteratively, for each registered node, denoting one of these registered nodes as P the set of unregistered nodes within some defined distance h become registered to p. The set of unregistered nodes that were newly assigned to be registered are removed from the set of unregistered nodes. This process is repeated until there are no more unregistered nodes. Each set of nodes registered with the same label are denoted as a community (an inconvenient label, given a network will be constructed and communities will be identified). A weighted network is constructed between these registered communities with edge weights quantifying the similarity in the quantitative features (as measured with the flow cytometry machine) between a pair of a communities. Once this weighted graph is created, a spectral community detection method (Xiang and Gong, 2008) is applied to segment the network into 1 of K network communities. These is one final post-processing step, motivated by previous work in computational flow cytometry methods, to combine the agglomerate a pair of network communities if members if the community show similarity greater than a predefined threshold (in terms, again, of their measured flow cytometry properties). The usefulness of this approach is that it exhibited outstanding performance on datasets containing clusters of challenging shapes. For example, overlapping clusters, non-elliptical shaped clusters, or low-density clusters. To summarize, the SamSPECTRAL method shows how network communities can be used to automate a challenging computational task and enables biologists to better study and characterize their data.

1.4.4 Understanding genetic diversity of the malaria parasite genes

Rich genetic diversity in the *var* genes of the human malaria parasite has been shown to contribute to the complexity of the epidemiology of the infection and disease. The parasite can change which

of the *var* genes are expressed at any given time on the infected red blood cell, which prevents the antibody from recognizing and resisting the new protein. One diversity-generating mechanism is recombination, which is the exchange and shuffling of genetic information during mitosis and meiosis (Barry et al., 2007). 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, (Larremore et al., 2013) 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 (Karrer and Newman, 2011).

After identifying communities within each HVR network, the authors used two summary statistics to formulate their biological hypothesis. First, the variation of information (Rosenberg and Hirschberg, 2007) 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* (Newman, 2002) 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, (Larremore et al., 2013) 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, presented by Levin *et al.,* in 2015 (Levine et al., 2015). Single cell technologies allow for the profiling of cells individually within a sample. Recent attention and methods development have focused on the use of RNA sequencing and mass cytometry to give a high dimensional profile for a single cell. Single cell technologies have enabled for an advancement in the understanding of the pathobiology of cancer in that cells within a tumor have been shown to exhibit a large amount of heterogeneity at the single cell level. Furthermore, this heterogeneity has important functional and clinical significance (Marusyk et al., 2012). 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 (Tong et al., 2008). 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 Challenging problems in community detection

1.5.1 Temporal Networks

1.5.2 Multilayer networks

1.5.3 Network Comparison

1.5.4 Large Networks

1.5.5 Attributed Networks

1.6 Thesis Contribution and Outline

In this thesis, we seek to apply and develop methods for community detection that can be applied to social and biological networks. In particular, we develop three extensions of community detection to multilayer networks, large networks, and attributed networks. For each of these three challenges, we present a method, software, and results on a variety of different networks. The thesis is organized as follows: First, in chapter 2, we present a comprehensive overview of the stochastic block model and the associated inference techniques for working with probabilistic models. In chapters 3, 4, and 5, we introduce community detection methods in multilayer networks, large networks, and attributed networks, respectively. In chapter 6, we present an application of community detection for the understanding of microbiome composition in patients with burn inhalation injury. Finally, in chapter 7, we provide future directions for the discussed work.

with a deep description of probabilistic network models and inference techniques in chapter 2, a develop

CHAPTER 2

Probabilistic community detection models and inference techniques

In this section, we will present two probabilistic models for community structure, the stochastic block model and the affiliation model.

2.1 Probabilistic graphical models for statistical inference

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

Probabilistic graphical models enable efficient specification and manipulation of large probability distributions through semantic structures. Given a set of random variables, $\{A, B, C, D, E, F\}$, we seek to compute the joint distribution, $P(A, B, C, D, E, F)$. This joint distribution can be expressed with a directed acyclic graph (DAG), whose structure encodes dependencies between random variables. The DAG allows for the representation of the joint distribution in a factorized way, which is computationally useful. A DAG between the set of random variables, $\{A, B, C, D, E, F\}$ is shown in 2.1.

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

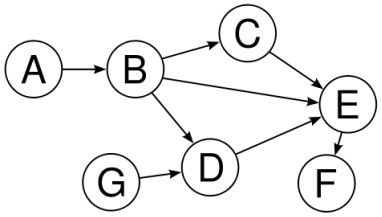

Figure 2.1: **Directed Acyclic Graph.** A directed acyclic graph (DAG) is formed based on dependency between random variable and allows for a fully factorized probability distribution.

$$P(\mathbf{X}) = \prod_{i=1:N} P(X_i \mid \mathbf{X}_{\pi_i}). \tag{2.1}$$

Here, π_i denotes the set of parents for node i. Using this information, we can write down the joint distribution for figure 2.1 as,

$$P(A, B, C, D, E, F) = P(A)P(B \mid A)P(C \mid B)P(D \mid B, G)P(E \mid D, B, C)P(F \mid E). \tag{2.2}$$

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.

2.2 Stochastic block model

2.2.1 Most general stochastic block model

For an undirected, unweighted network $\mathcal{G}$ with adjacency matrix, $\mathbf{A}$, we seek to partition each of the N nodes into one of K communities. We denote the 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 2.2 shows the dependency relationship between the node-to-community assignments. Here, the node-to-community assignments are treated as a latent variables because we seek to identify the $\mathbf{z}$ that makes the observed adjacency matrix, $\mathbf{A}$ the most likely. The crucial assumption of the stochastic block model is that

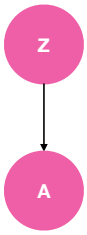

Figure 2.2: **SBM Graphical Model.** A graphical model is used to model the dependency between the node-to-community assignments, $\mathbf{z}$ and the observed network adjacency matrix, $\mathbf{A}$.

nodes within a community are connected to nodes within their community and to other communities in a characteristic way. To this end, the model fitting procedure requires learning a set of within and between community connection probabilities. Under this approach, edges are treated as independent and identically distributed and deciding whether or node an edge exists between a pair of nodes is the learned connection probability between the communities to which each of the nodes belong.

Using the factorization rules described in section 2.1, we can specify the complete data log likelihood between $\mathbf{z}$ and $\mathbf{A}$ as,

$$\log P(\mathbf{z}, \mathbf{A}) = \log(P(\mathbf{A} \mid \mathbf{z})) + \log(P(\mathbf{z})) \tag{2.3}$$

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

$$P(A_{ij} = 1) \sim \text{Bernoulli}(\Pi_{z_i,z_j}) \tag{2.4}$$

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,

$$\log(P(\mathbf{Z})) = \sum_i \sum_k Z_{ik} \log(\alpha_k). \tag{2.5}$$

Next,

$$\log(P(\mathbf{A} \mid \mathbf{Z})) = \sum_{i \neq j} \sum_{k<l} Z_{ik} Z_{il} [a_{ij} \log(\Pi_{kl}) + (1 - a_{ij}) \log(1 - \Pi_{kl})] \tag{2.6}$$

Optimizing the parameters of this incomplete data log likelihood requires computing the posterior $P(\mathbf{z} \mid \mathbf{A})$ but as shown by (Daudin et al., 2008) is intractable. To address this issue, the posterior can be recast using a factorized approximation. This is accomplished by optimizing a lower bound of $\mathcal{L}(\mathbf{A})$. We let $\mathcal{R}_A$ be an approximation of the posterior, $P(\mathbf{z} \mid \mathbf{A})$. To optimize the lower bound of $\log \mathcal{A}$, we seek the $\mathcal{R}_A$ that is as close as possible to $P(\mathbf{z} \mid \mathbf{A})$. In other words, we define the lower bound of $\mathcal{L}(\mathbf{A})$ as $\mathcal{T}(\mathcal{R}_A)$, with,

$$\mathcal{T}(\mathcal{R}_A) = \log \mathcal{L}(\mathbf{A}) - \mathrm{KL}[\mathcal{R}_A(\mathbf{z}), \mathbf{P}(\mathbf{z} \mid \mathbf{A})]. \tag{2.7}$$

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 (Jaakkola, 2001) as,

$$\mathcal{R}_A(\mathbf{z}) = \prod_i h(Z_i; \boldsymbol{\tau}_i). \tag{2.8}$$

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 (Daudin et al., 2008) *et al.*, show that the optimal estimate for τ_{ik} denoted $\hat{\tau}_{ik}$ satisfies

$$\hat{\tau}_{ik} \propto \alpha_k \prod_{j \neq i} \prod_l [\theta_{z_i, z_j}^{a_{ij}} (1 - \theta_{z_i, z_j})^{1 - a_{ij}}]^{\hat{\tau}_{ik}}. \tag{2.9}$$

Here, α_k notes the probability that a node belongs to community k. Furthermore, after computing the set of variational parameters, the updates for $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ that maximize $\mathcal{T}(\mathcal{R}_A)$ are also shown by Daudin *et al.,* (Daudin et al., 2008) to be,

$$\hat{\alpha}_k = \frac{1}{n} \sum_i \hat{\tau}_{ik} \qquad \theta_{ql} = \sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{jl} a_{ij} / \sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{jl} \tag{2.10}$$

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.* (Zhang et al., 2012) also show that belief propgation is very effective for this task (Murphy et al., 1999). Briefly, belief propagation is a message passing algorithm for parameter inference in probabilistic graphical models. Given that parameter learning offer requires computing marginal distributions for a set of variables with a very large number of possible configurations, belief propagation uses the graphical model to reduce the complexity of the problem. Using the belief propagation to infer latent node-to-community assignments and update the model parameters was shown to perform superperior to the variational appromixation

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.

2.2.2 Variants to the Classic Stochastic Block Model

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

Edge Weights

The majority of the stochastic block model literature considers unweighted networks simply because describing a probabilistic model to handle both edge existence and edge weight is a challenging task. In the classic stochastic block model, we are simply modeling whether an edge exists based on the inferred community memberships of the edge stubs. Since edge weights can come in a variety of forms (real-valued, count, etc.), it is difficult to immediately decide what distribution the edge weights should follow. In the past few years, this issue has been tackled in two papers (Aicher et al., 2014; Peixoto, 2018).

First, Aicher *et al.* developed a model and associated inference technique, for the weighted stochastic block model. Here, edge weights can be modeled by any exponential family distribution. The authors use a mixing parameter that allows for the control of the use of edge existence versus

edge weights when learning node-to-community assignments. This method requires having an estimate of the number of communities, K, but the paper provides an approach to use Bayes' factors between two competing values of K to determine which model is a better fit.The inference for fitting this model is performed through a variational bayes approach (Attias, 2000).

To avoid having intuition about K, Peixoto (Peixoto, 2018) developed a non parametric bayesian approaches that is capable of inferring K with no prior knowledge. The assumption of the model is also slightly different and assumes a hierarchical structure between communities. The inference is achieved through MCMC sampling.

Degree Heterogeneity

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

The restriction of single community membership

As it is often observed in social networks, the assumption that every node belongs to only a single community is restrictive. To address this issue, approaches have been developed to allow nodes to participate in a mixture of communities (Airoldi et al., 2008) or to overlapping groups (Latouche et al., 2011). Airoldi *et al.*, pioneered the development of the mixed membership stochastic block model (Airoldi et al., 2008), 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 $\boldsymbol{\theta}$ representing the between community connection probabilities.

- For each node p, draw a mixed membership vector $\pi_p \sim$ Dirchelet($\boldsymbol{\alpha}$)
- Then for each pair of nodes (p, q), draw $\mathbf{z}_{p\to q} \sim$ Multinomial(π_p), $\mathbf{z}_{q\to p} \sim$ Multinomial(π_q)
- Sample the edge between p and q as, A_{pq}, where $A_{pq} \sim$ 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.* (Latouche et al., 2011) addressed an important limitation of (Airoldi et al., 2008). 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 more structures of the network.

2.3 Affiliation model and inference

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

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

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

CHAPTER 3

A multilayer stochastic block model

In this chapter we present the strata multilayer stochastic block model (sMLSBM). The sMLSBM method and inference described here is described in *Clustering Network Layer with the Strata Multilayer Stochastic Block Model* (Stanley et al., 2016). The goal in developing this method is two-fold. First, we seek to develop an approach to cluster network layers within a multilayer network. Second, we wish to develop a novel extension to the stochastic block model to handle the information contained across network layers and determine which subsets of the network layers are are likely to be samples from the same stochastic block model.

3.1 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 (Zitnik and Leskovec, 2017), or the desire to study multiple microbial co-occurence networks in different sites of the body (Turnbaugh et al., 2007). Multilayer networks provide a framework to do this, in that each relational definition leads to a new layer in the network (Kivelä et al., 2014; Boccaletti et al., 2014; De Domenico et al., 2013). 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 (Costanzo et al., 2010), in understanding underlying relationships and community structure across social networks (Greene and Cunningham, 2013),

and in the analysis of temporal networks (Mucha et al., 2010). Furthermore, recent advances in the mathematical foundations for multilayer networks have made analysis of these types of data more feasible. In particular, (De Domenico et al., 2013) 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 (Mucha et al., 2010). 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 (De Domenico et al., 2015b) 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 (Madeira and Oliveira, 2004) or co-clustering (Dhillon, 2001) 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 (Dhillon, 2001), 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.

3.2 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 (Ugander et al., 2013; Tsuda and Kudo, 2006); multiscale properties such as community structure (Onnela et al., 2012; Ni et al., 2015; Iacovacci et al., 2015), the spectra of network-related matrices (Brandes et al.,

2009) and by defining latent roles (Brandes et al., 2011). 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 (Porter et al., 2009a; Fortunato, 2010). Although there are many definitions for what constitutes a "community" (Rombach et al., 2014), 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 (Newman, 2006b) and fitting a generative probabilistic model (Jacobs and Clauset, 2014). Because each of these approaches present computational challenges for efficiently detecting communities, numerous heuristics exist for developing practical algorithms (Porter et al., 2009b; Fortunato, 2010; Leskovec et al., 2009; Clauset et al., 2007; Newman, 2006c).

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 (Ni et al., 2015), 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 (Iacovacci et al., 2015), 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) (Snijders and Nowicki, 1997b), 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. (Rombach et al., 2014; Aicher et al., 2015a)). 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 (Jacobs and Clauset, 2014; Guimerà and Sales-Pardo, 2009). 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.

3.3 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 (Mucha et al., 2010) and studying dynamics (De Domenico et al., 2015a) 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 (Han et al., 2015; Paul and Chen, 2015; Barbillon et al., 2015; Valles-Catala et al., 2014; Peixoto, 2015). In this context, the general assumption is that there are shared patterns in community structure across the layers of a multilayer network, and the goal is to define and identify a stochastic block model that captures this structure. These works have explored

many types of applications that can arise involving multilayer networks, and have therefore given rise to several complementary models for multilayer stochastic block models (MLSBMs). We now briefly summarize this previous work that is very related, but notably different, from the model we study herein.

In Refs. (Han et al., 2015; Paul and Chen, 2015; Barbillon et al., 2015), 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. (Han et al., 2015) 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 (**?**) to the multilayer setting.

Ref. (Han et al., 2015) was followed up by Ref. (Paul and Chen, 2015), 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. (Barbillon et al., 2015) is similar to Refs. (Han et al., 2015) and (Paul and Chen, 2015) 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. (Valles-Catala et al., 2014) and (Peixoto, 2015), which provide techniques to determine whether a single layer network is the result of an aggregation procedure in a multilayer network. In Ref. (Valles-Catala et al., 2014), 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. (Peixoto, 2015) 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 (De Domenico et al., 2015b), 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.

3.4 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

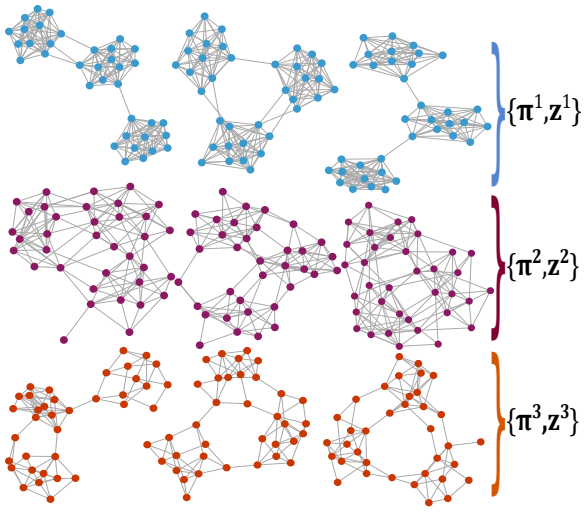

Figure 3.1: **Objective of strata multilayer stochastic block model (sMLSBM)**. Each of the $L = 9$ networks here represents a layer in a multilayer network. Every network layer has $N = 36$ nodes that are consistent across all layers. There are $S = 3$ strata as indicated by the three rows and the colors 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.

3.5 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 (De Domenico et al., 2015b). 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.

3.6 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,

$$p(\mathcal{G} \mid \mathbf{\Pi}) = \sum_{\mathcal{Z}} \sum_{\mathbf{Y}} p(\mathcal{G}, \mathcal{Z}, \mathbf{Y} \mid \mathbf{\Pi}). \tag{3.1}$$

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. 3.2, 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 (**?**). 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

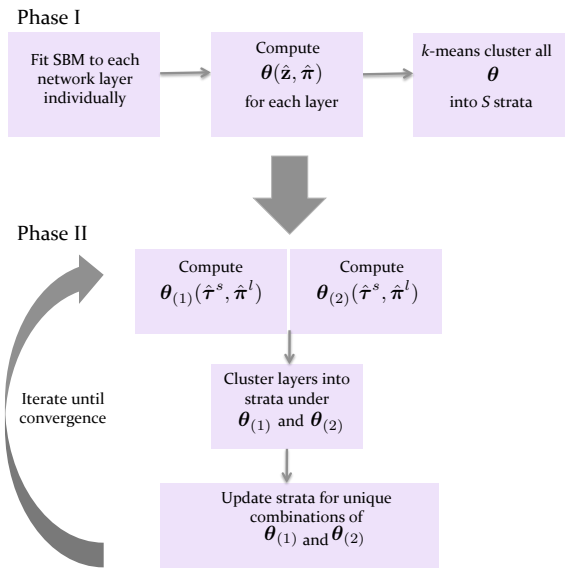

Figure 3.2: **Schematic illustration of our algorithm**: Our algorithm for fitting an sMLSBM is broken up into two phases: an initialization phase to cluster layers into strata, and an iterative phase that allows learning of node-to-community and layer-to-strata assignments.

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 (Tibshirani et al., 2001). 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. (Daudin et al., 2008) to a multilayer context. Note that this is similar to Ref. (Han et al., 2015), 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,

$$p(\mathcal{A}^s, \mathbf{Z}^s) = p(\mathcal{A}^s \mid \mathbf{Z}^s) p(\mathbf{Z}^s), \tag{3.2}$$

where

$$p(\mathcal{A}^s \mid \mathbf{Z}^s) = \prod_{l \in \mathcal{L}^s} \prod_{i<j} \prod_{mn} {\pi^s_{mn}}^{A^l_{ij}} (1 - \pi^s_{mn})^{(1-A^l_{ij})}. \tag{3.3}$$

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

$$p(\mathbf{Z}^s) = \prod_i \prod_m {\alpha^s_m}^{(Z^s_{im})}. \tag{3.4}$$

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

$$\begin{aligned} \log P(\mathcal{A}^s, \mathbf{Z}^s) &= \log(P(\mathbf{Z}^s)) + \log(P(\mathcal{A}^s \mid \mathbf{Z}^s)) \\ &= \sum_i \sum_m Z^s_{im} \log(\alpha^s_m) \\ &+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} A^l_{ij} \log(\pi^s_{mn}) \\ &+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} (1 - A^l_{ij}) \log(1 - \pi^s_{mn}). \end{aligned} \tag{3.5}$$

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 (Dempster et al., 1977). Doing so requires the ability to compute $P(\mathbf{Z}^s \mid \mathcal{A}^s)$; however, Ref. (Daudin et al., 2008) 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 (Han et al., 2015; Barbillon et al., 2015; Daudin et al., 2008). 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

$$R_{\mathcal{A}^s}(\mathbf{Z}^s) = \prod_i h(\mathbf{Z}^s_{i\cdot}; \boldsymbol{\tau}_{i\cdot}). \tag{3.6}$$

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

$$\mathcal{J}(R_{\mathcal{A}^s}) = \ell\ell(\mathcal{A}^s) - \mathrm{KL}(R_{\mathcal{A}^s}(\mathbf{Z}^s), P(\mathbf{Z}^s \mid \mathcal{A}^s)), \tag{3.7}$$

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

$$\begin{aligned}
\mathcal{J}(R_{\mathcal{A}^s}) = &\sum_i \sum_m \tau^s_{im} \log(\alpha^s_m) \\
&+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} \tau^s_{im} \tau^s_{jn} [A^l_{ij} \log(\pi^s_{mn})] \\
&+ \sum_{l \in \mathcal{L}^s} \sum_{i<j} \sum_{mn} \tau^s_{im} \tau^s_{jn} [(1 - A^l_{ij}) \log(1 - \pi^s_{mn})] \\
&- \sum_i \sum_m \tau^s_{im} \log(\tau^s_{im}).
\end{aligned} \tag{3.8}$$

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:

$$\hat{\alpha}^s_m = \sum_i \hat{\tau}^s_{im} / N \,, \tag{3.9}$$

$$\hat{\pi}^s_{qt} = \frac{\sum_{l \in \mathcal{L}^s} \sum_{i<j} \hat{\tau}^s_{im} \hat{\tau}^s_{jn} A^l_{ij}}{\sum_{l \in \mathcal{L}^s} \sum_{i<j} \hat{\tau}^s_{im} \hat{\tau}^s_{jn}} \,, \tag{3.10}$$

$$\hat{\tau}^s_{im} \propto \hat{\alpha}^s_m \prod_{l \in \mathcal{L}^s} \prod_{i<j} \prod_n [\hat{\pi}^s_{mn}{}^{A^l_{ij}} (1 - \hat{\pi}^s_{mn})^{1 - A^l_{ij}}]^{\hat{\tau}^s_{jn}} \,. \tag{3.11}$$

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

$$\boldsymbol{\theta}^l_{(1)} = \boldsymbol{\theta}^l(\overline{\boldsymbol{\tau}^s}, \tilde{\boldsymbol{\pi}}^l), \tag{3.12}$$

$$\boldsymbol{\theta}^l_{(2)} = \boldsymbol{\theta}^l(\tilde{\boldsymbol{\tau}}^l, \overline{\boldsymbol{\pi}^s}) \tag{3.13}$$

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.

3.7 Synthetic Examples

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

3.7.1 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

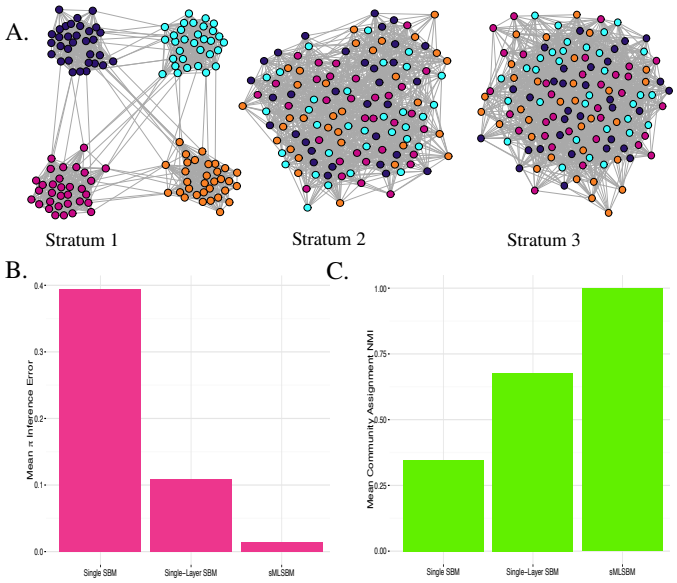

Figure 3.3: **Synthetic experiment comparing sMLSBM to other SBMs. A**. We specified a model with $S = 3$ strata and $L = 10$ layers per stratum. A representative layer from each stratum is plotted. Note that nodes in all networks are colored according to their community membership in stratum 1. Each 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 $\text{vec}(\overline{\boldsymbol{\pi}^{y_l}})$ and its true value, $\text{vec}(\boldsymbol{\pi}^l)$. Note that $\text{vec}(\mathbf{X})$ is the $\frac{K(K+1)}{2}$ length vector representing the

lower triangle of the matrix $\mathbf{X}$. Moreover, to quantify error, we compute $||\text{vec}(\boldsymbol{\pi}^l) - \text{vec}(\overline{\boldsymbol{\pi}^{s_l}})||_2$. 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) (Danon et al., 2005) 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.

3.7.2 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. 3.7.1, 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 (Decelle et al., 2011a). 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 $N(p_{in}^2 - p_{out}^2) = 10$, which represents the point where the two strata are indistinguishable since $(p_{in}^1, p_{out}^1) = (p_{in}^2, p_{out}^2)$. In Fig. 4(A), we show the NMI between the true layer-to-strata assignments and those inferred by sMLSBM, or 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. 3.7.1. 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.

3.8 Human Microbiome Project Example

As an application of sMLSBM, we consider correlation networks constructed from data from the Human Microbiome Project (Turnbaugh et al., 2007). For various sites on the body, the human microbiome project has successfully collected multiple human samples in order to better understand interactions between bacterial species. In this context, network inference is particularly interesting,

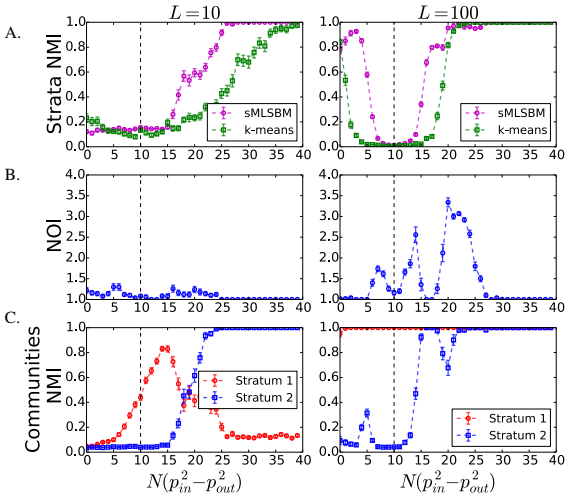

Figure 3.4: **Synthetic experiment with two strata.** We conducted numerical experiments with multilayer networks with $N = 128$ nodes, mean degree $c = 16$, $S = 2$ strata and $K^1 = K^2 = 4$ communities. The networks contained either $L = 10$ (left column) or $L = 100$ layers (right column), which 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).

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 (Faust et al., 2012). 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 (Friedman and Alm, 2012) 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. (Friedman and Alm, 2012), 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 (Friedman and Alm, 2012)). 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.

3.8.1 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 (De Domenico et al., 2015b). 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. (Ding and Schloss, 2014) offers a state-of-the-art clustering of body sites based on biological expertise. Here, the authors have advanced understanding of microbial community composition through the

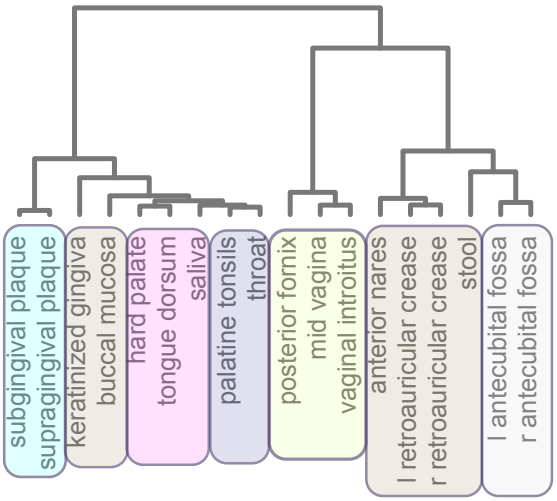

Figure 3.5: **Comparison of sMLSBM on the OTU interaction networks (Friedman and Alm, 2012) for each of the body sites to a reducibility hierarchy (De Domenico et al., 2015b).** As described in the text, we consider a multiplex network with $L = 18$ layers and $N = 213$ nodes, which we group here into $S = 6$ 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.

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 (Ding and Schloss, 2014) found a surprising correlation between samples from stool and oral cavity, which is reflected in our result.

3.8.2 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.

3.9 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 (De Domenico et al., 2015b). 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, there are a variety of ways to aggregate layers in a strata. See,

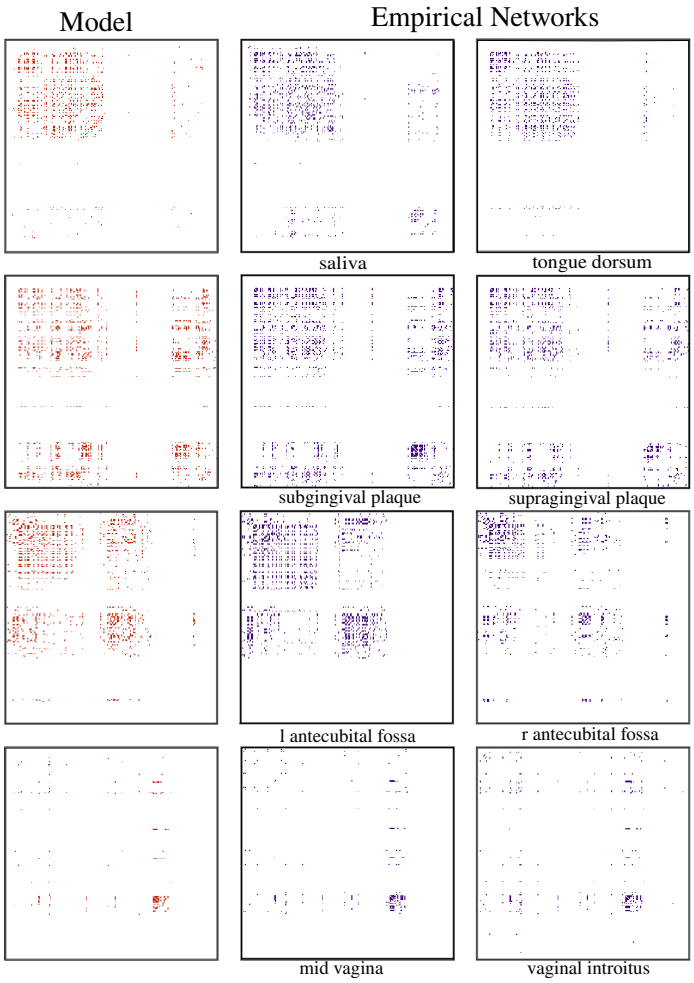

Figure 3.6: **Visualization of Strata in SparCC Networks.** We visualize the adjacency matrices for SparCC networks that encode microbiome interactions at body sites. In each panel, a colored dot at position (i, j) indicates the existence of an edge (i, j) in the corresponding network layer. The four 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.

for example, Ref. (Taylor et al., 2015), 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. (De Domenico et al., 2015b) (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 (De Domenico et al., 2015b) 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 (Aicher et al., 2015b) and (Wang and Wong, 1987), respectively. It would also be useful to extend to degree-corrected and overlapping (i.e., mixed-membership) communities (Karrer and , 2011), 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 (Zapién-Campos et al., 2015). 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.

3.10 Detectability in a single stratum

The development of sMLSBM motivated the analysis for how multiple layers can be collectively used to more accurately learn SBM model parameters in the single stratum case. That is, given a collection of sparse networks from a multilayer stochastic block model with one stratum, how can the layers most accurately be combined to give the most accurate definition of community structure. We investigate these questions in *Enhanced detectability of community structure in multilayer networks*

through layer aggregation (Taylor et al., 2015). In particular, we studied the detectability limitations of the stochastic block model for a multilayer network with 1 stratum using random matrix theory techniques.

3.10.1 Investigating detectability in a multilayer network

Community structure detectability has gained considerable attention (Lancichinetti and Fortunato, 2011; Reichardt and Leone, 2008; Hu et al., 2012; Decelle et al., 2011b; Nadakuditi and Newman, 2012; Abbe et al., 2016) 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 (Radicchi, 2013), hierarchical structure (Peixoto, 2013; Sarkar et al., 2013), and in temporal networks (Ghasemian et al., 2016), 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, $\tilde{L}$.

3.10.2 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 (Decelle et al., 2011b; Nadakuditi and Newman, 2012), characterized by the solution curve (Δ^*, ρ) to

$$N\Delta = \sqrt{4N\rho}, \tag{3.14}$$

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 3.14 was derived for sparse networks (i.e. constant ρN so that $\rho = O(N^{-1})$) and was obtained using both a Bayesian analysis (Decelle et al., 2011b) and random matrix theory (Nadakuditi and Newman, 2012).

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.

3.10.3 Using random matrix theory to study detectability

Since node-to-community assignments, $\mathbf{z}$ can be inferred with spectral method, random matrix theory (Benaych-Georges and Nadakuditi, 2011; Nadakuditi and Newman, 2013) is a useful approach for

studying partitioning and phase transitions in detectability (i.e. node-to-community assignment accuracy) (Nadakuditi and Newman, 2012; Peixoto, 2013; Sarkar et al., 2013). 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, $\bar{B_{ij}} = \bar{A}_{ij} - \rho L$ (Newman and Girvan, 2004).

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

$$f(a; L, p) = \binom{L}{a} p^a (1-p)^{L-a} \tag{3.15}$$

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,

$$P(\lambda) = \frac{\sqrt{\lambda_2^2 - \lambda^2}}{\pi \lambda_2^2 / 2} \tag{3.16}$$

CHAPTER 4

Network compression for community detection with super nodes

CHAPTER 5

An attributed stochastic block model

CHAPTER 6

Community detection for understanding burn inhalation injury

CHAPTER 7

Conclusion and future work

BIBLIOGRAPHY

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.

Aghaeepour, N., Finak, G., Hoos, H., Mosmann, T. R., Brinkman, R., Gottardo, R., Scheuermann, R. H., Consortium, F., Consortium, D., et al. (2013). Critical assessment of automated flow cytometry data analysis techniques. *Nature methods*, 10(3):228.

Aghaeepour, N., Ganio, E. A., Mcilwain, D., Tsai, A. S., Tingle, M., Van Gassen, S., Gaudilliere, D. K., Baca, Q., McNeil, L., Okada, R., et al. (2017). An immune clock of human pregnancy. *Science immunology*, 2(15):eaan2946.

Aicher, C., Jacobs, A. Z., and Clauset, A. (2014). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.

Aicher, C., Jacobs, A. Z., and Clauset, A. (2015a). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.

Aicher, C., Jacobs, A. Z., and Clauset, A. (2015b). Learning latent block structure in weighted networks. *Journal of Complex Networks*, 3(2):221–248.

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014.

Attias, H. (2000). A variational 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.

Benaych-Georges, F. and Nadakuditi, R. R. (2011). The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics*, 227(1):494–521.

Bendall, S. C., Nolan, G. P., Roederer, M., and Chattopadhyay, P. K. (2012). A deep profiler's guide to cytometry. *Trends in immunology*, 33(7):323–332.

Bender, E. A. and Canfield, E. R. (1978). The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C. I., Gómez-Gardeñes, J., Romance, M., Sendina-Nadal, I., Wang, Z., and Zanin, M. (2014). The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122.

Brandes, U., Lerner, J., and Nagel, U. (2011). Network ensemble clustering using latent roles. *Advances in Data Analysis and Classification*, 5(2):81–94.

Brandes, U., Lerner, J., Nagel, U., and Nick, B. (2009). Structural trends in network ensembles. In *Complex networks*, pages 83–97. Springer.

Browet, A., Absil, P.-A., and Van Dooren, P. (2011). Community detection for hierarchical image segmentation. In *IWCIA*, volume 11, pages 358–371. Springer.

Chen, H., Lau, M. C., Wong, M. T., Newell, E. W., Poidinger, M., and Chen, J. (2016). Cytofkit: a bioconductor package for an integrated mass cytometry data analysis pipeline. *PLoS computational biology*, 12(9):e1005112.

Clauset, A., Moore, C., and Newman, M. E. (2007). Structural inference of hierarchies in networks. In *Statistical network analysis: models, issues, and new directions*, pages 1–13. Springer.

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. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008.

Daudin, J.-J., Picard, F., and Robin, S. (2008). A mixture model for random graphs. *Statistics and computing*, 18(2):173–183.

De Domenico, M., Lancichinetti, A., Arenas, A., and Rosvall, M. (2015a). Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X*, 5(1):011027.

De Domenico, M., Nicosia, V., Arenas, A., and Latora, V. (2015b). Structural reducibility of multilayer networks. *Nature communications*, 6.

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

Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011a). Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701.

Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011b). Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM.

Ding, T. and Schloss, P. D. (2014). Dynamics and associations of microbial community types across the human body. *Nature*, 509(7500):357–360.

Faust, K., Sathirapongsasuti, J. F., Izard, J., Segata, N., Gevers, D., Raes, J., and Huttenhower, C. (2012). Microbial co-occurrence relationships in the human microbiome. *PLoS computational biology*, 8(7):e1002606.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

Fortunato, S. and Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, 659:1–44.

Friedman, J. and Alm, E. J. (2012). Inferring correlation networks from genomic survey data. *PLoS computational biology*, 8(9):e1002687.

Ghasemian, A., Zhang, P., Clauset, A., Moore, C., and Peel, L. (2016). Detectability thresholds and optimal algorithms for community structure in dynamic networks. *Physical Review X*, 6(3):031005.

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.

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*.

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.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *Journal of Complex Networks*, 2(3):203–271.

Kossinets, G. and Watts, D. J. (2009). Origins of homophily in an evolving social network. *American journal of sociology*, 115(2):405–450.

Lancichinetti, A. and Fortunato, S. (2009). Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117.

Lancichinetti, A. and Fortunato, S. (2011). Limits of modularity maximization in community detection. *Physical review E*, 84(6):066122.

Larremore, D. B., Clauset, A., and Buckee, C. O. (2013). A network approach to analyzing highly recombinant malaria parasite genes. *PLoS computational biology*, 9(10):e1003268.

Latouche, P., Birmelé, E., and Ambroise, C. (2011). Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, pages 309–336.

Layeghifard, M., Hwang, D. M., and Guttman, D. S. (2017). Disentangling interactions in the microbiome: a network perspective. *Trends in microbiology*, 25(3):217–228.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123.

Levine, J. H., Simonds, E. F., Bendall, S. C., Davis, K. L., El-ad, D. A., Tadmor, M. D., Litvin, O., Fienberg, H. G., Jager, A., Zunder, E. R., et al. (2015). Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197.

Lorrain, F. and White, H. C. (1971). Structural equivalence of individuals in social networks. *The Journal of mathematical sociology*, 1(1):49–80.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45.

Marusyk, A., Almendro, V., and Polyak, K. (2012). Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*, 12(5):323.

Meunier, D., Lambiotte, R., Fornito, A., Ersche, K. D., and Bullmore, E. T. (2009). Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878.

Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc.

Nadakuditi, R. R. and Newman, M. E. (2012). Graph spectra and the detectability 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 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*.

Peixoto, T. P. (2013). Eigenvalue spectra of modular networks. *Physical review letters*, 111(9):098701.

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.

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*.

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.

Stanley, N., Shai, S., Taylor, D., and Mucha, P. J. (2016). Clustering network layers with the strata multilayer stochastic block model. *IEEE transactions on network science and engineering*, 3(2):95–105.

Taylor, D., Shai, S., Stanley, N., and Mucha, P. J. (2015). Enhanced detectability of community structure in multilayer networks through layer aggregation. *arXiv preprint arXiv:1511.05271*.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Tong, H., Faloutsos, C., and Pan, J.-Y. (2008). Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, 14(3):327–346.

Traud, A. L., Kelsic, E. D., Mucha, P. J., and Porter, M. A. (2011). Comparing community structure to characteristics in online collegiate social networks. *SIAM review*, 53(3):526–543.

Tsuda, K. and Kudo, T. (2006). Clustering graphs by weighted substructure mining. In *Proceedings of the 23rd international conference on Machine learning*, pages 953–960. ACM.

Turnbaugh, P. J., Ley, R. E., Hamady, M., Fraser-Liggett, C. M., Knight, R., and Gordon, J. I. (2007). The human microbiome project. *Nature*, 449(7164):804–810.

Ugander, J., Backstrom, L., and Kleinberg, J. (2013). Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1307–1318. International World Wide Web Conferences Steering Committee.

Valles-Catala, T., Massucci, F. A., Guimera, R., and Sales-Pardo, M. (2014). stochastic block models reveal the multilayer structure of complex networks. *arXiv preprint arXiv:1411.1098*.

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

Wang, Y. J. and Wong, G. Y. (1987). Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19.

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

Yang, J. and Leskovec, J. (2012). Community-affiliation graph model for overlapping network community detection. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1170–1175. IEEE.

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.

Zhang, P., Krzakala, F., Reichardt, J., and Zdeborová, L. (2012). Comparative study for inference of hidden classes in stochastic block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2012(12):P12021.

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