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 (138), clustering (51), and prediction tasks (146; 45). Motivated by problems in fields such as, biology (78), medicine (8), neuroscience (17), and social science (57), the field of network analysis has gained popularity and seeks to develop tools for understanding the associated network data. The main objectives in creating tools for the analysis of network data is to enable effective modeling, prediction, and data interpretation. In this thesis, we present three new methods that enable a more thorough understanding of the structural organization patterns in networks through *community detection*. The objective of community detection is to partition the network nodes into *communities*, such that members of a community have similar connectivity patterns. With an increasing amount of more challenging types of network data, such as those containing multiple relational definitions between a set of nodes, standard community detection approaches are often insufficient. In this thesis, we will look in depth at how to handle communities in networks that are *multilayer*, *large*, and *attributed*. We then present several case studies in each of the developed methods in applications such as, microbiome analysis, protein interaction network understanding, and mining in social networks. We show that the successful identification of communities in these types of networks allows the network to be used for tasks such as, efficient summarization, prediction, and classification.

In this introduction, we first present notation, terminology, and useful concepts for working with networks. We then provide a detailed discussion about community detection, highlighting not only the main methods studied in this thesis, but also the recently developed novel and state-of-the-art approaches. Next, we provide several examples of how community detection is used as an important tool in computational biology, as it assists in tasks such as, clustering, biological interpretation, and prioritizing further experiments. Finally, we discuss challenges in the field of community detection and how this work addresses some of these problems.

1.1 Network Notation and Basic Summarization

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

1.1.1 Representing relational information

Humans frequently benefit from network applications for tasks such as, viewing relevant queries from a google search, enjoying a suggested movie on Netflix, or interacting on a social network platform. The basic building blocks of networks are nodes, representing entities in a systems, and edges, encoding connections their physical or inferred connection or similarity. Figure 1.1 shows a collaboration network between the six people that made the work in this thesis possible. An edge between a pair of people indicates if they have written a paper together.

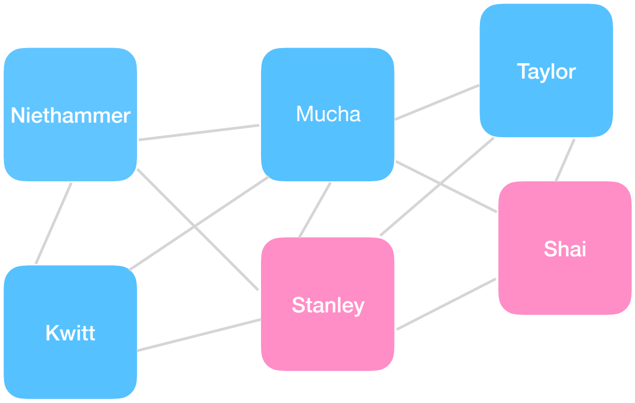

Figure 1.1: **A simple network example (coauthorship).** A co-authorship network with an edge between a pair of people if they have written a paper together.

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

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

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

In the *weighted* case of undirected networks, edge weights are some real number and are frequently quantities such as correlation or pairwise similarity. A simple extension of $\mathbf{A}$ to an undirected, weighted network where w is the edge weight between nodes i and j, computes the adjacency matrix entry a_{ij} as,

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

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

1.1.2 Network Summary Statistics

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

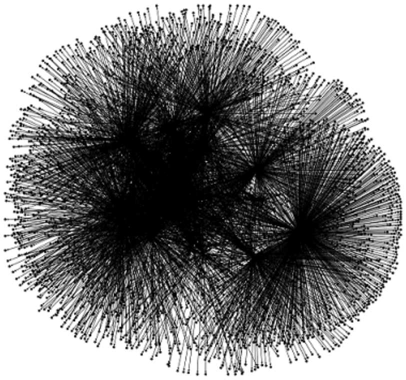

Figure 1.2: **Hairball network.** Networks are often noisy data structures and lack an immediate straight forward structural interpretation. *Image from* `https://cs.umd.edu`.

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

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

To illustrate these quantities in an applied context, we will compute them on an example network shown in Figure 1.3. This network is constructed from a single cell mass cytometry dataset, which was originally described in Ref. (148) and released publicly and processed using the Cytofkit R package (31). Each node represents a single cell and is represented with 52 features for a mass cytometry analysis. Briefly, mass cytometry is a technique to simultaneously measure multiple immunological features in a cell or tissue (19). From this data matrix, we created a network by selecting 500 cells and building a k-nearest neighbor network with $k = 5$. This means that for a node i, we found its 5 nearest neighbors according to Euclidean distance, and connected them all to node i.

1.1.2.2 Degree Distribution

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

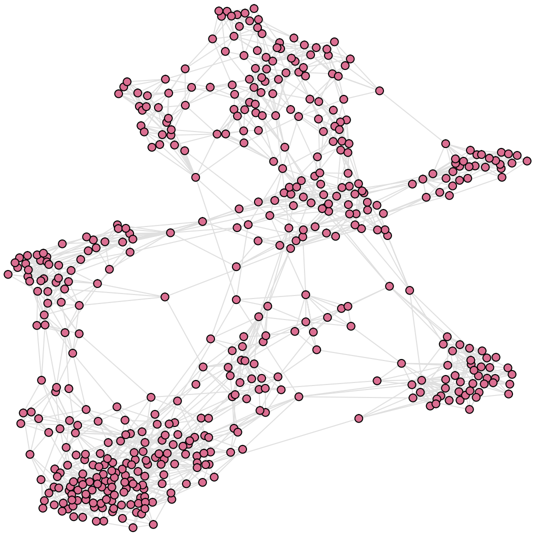

Figure 1.3: **Network of single cells.** We constructed a network from mass cytometry profiling among 500 cells in single cell dataset. Each cell has 52 measured immune features. In this network, each node is a single cell and is connected to its 5 nearest neighbors.

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

In the case of an undirected, unweighted network, the degree of node i counts its number of neighbors, while in the undirected, weighted context, degree encodes the total edge weight incident to node i. Collectively examining the distribution of degrees for a network is known as the *degree distribution*. Understanding the degree distribution provides insight into the network type and structural organization. We visualize degree distribution in Figure 1.4 using a cumulative distribution plot (A.) and a simple histogram (B.). Since this network was constructed with a 5-nearest neighbor rule, we see this reflected in the degree distribution, with all nodes having degree 5 or more. A few nodes have significantly higher degree (> 10) and represent single cells who is a nearest neighbor to many of the other cells in the original 52 dimensional space. A node's degree is often highly related to its importance in the network, which provides a nice transition to the next set of summary statistics, network centrality measures.

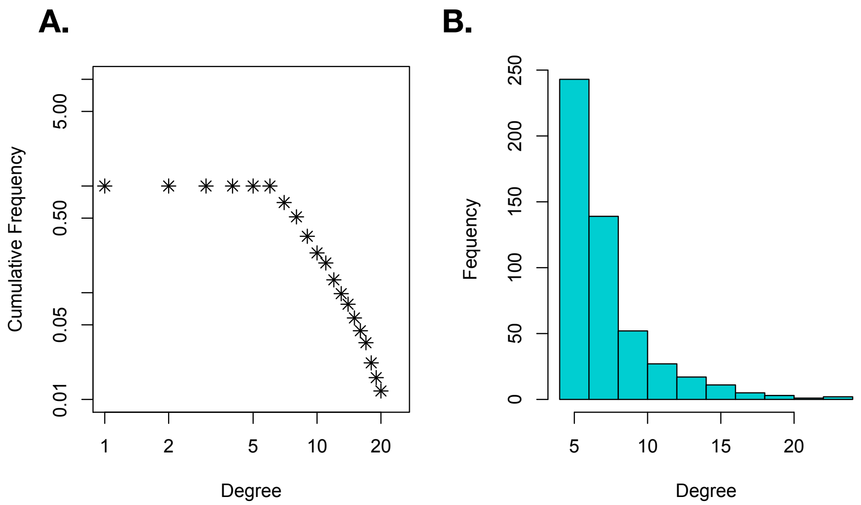

Figure 1.4: **Degree distribution for the single cell network.** We visualize the degree distribution in the single cell network presented in Figure 1.3. **A.** We compute a cumulative distribution plot for degree. **B.** Node degrees can also be visualized with a simple histogram.

1.1.2.3 Centrality

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

Degree centrality

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

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

Betweenness centrality

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

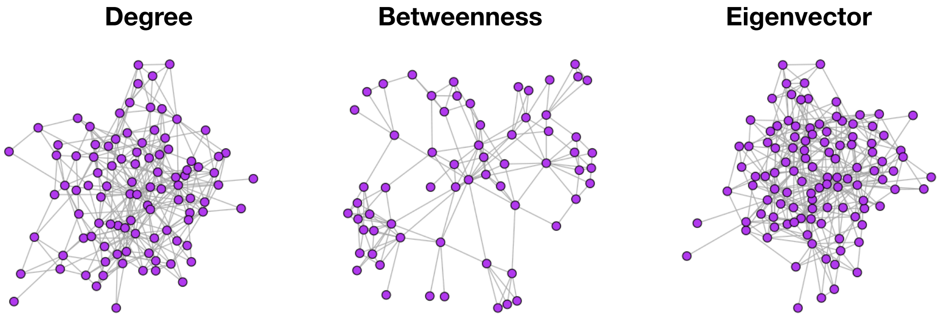

Figure 1.5: **Centralities on the single cell network.** The second order ego network for the highest centrality nodes in the single cell network according to degree, betweenness, and eigenvector in the left, center, and right plots, respectively. These plots are meant to emphasize how each of these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

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

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

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

Eigenvector centrality

The idea behind eigenvector centrality is that a node should be prioritized not only based on its degree, but the degree of its neighboring nodes. That is, a node connected to other 'important' or high degree nodes should be ranked higher than one connected to many low degree nodes [1]. The eigenvector centrality for node i, can be computed using the spectra of the adjacency matrix, $\boldsymbol{A}$. In particular, the vector of centralities, $\mathbf{x}$ is the one satisfying the eigenvector equation,

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

Because centralities are non-zero, the solution must be an eigenvector with all positive entries. Since multiple eigenvalues (λ) correspond to non-zero eigenvectors, the eigenvector corresponding to the largest eigenvector is used and the centrality scores for each node reflect its relative importance

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

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

We visualized the results of each of these three presented centralities on the single cell network data in Figure 1.5. Under each of the centrality measures, we selected the the highest-ranked centrality node and focused in its local ego network. This is shown for degree, betweenness, and eigenvector centralities in the left, middle, and right panels respectively. In particular from these high centrality nodes, we visualized their corresponding order 2 ego networks. An ego network for node i is simply the subgraph of all nodes within two hops of node i. This visualization gives a sense of what kinds of connectivity patterns each centrality measure favors. For example, we see that degree and eigenvector centrality have similar ego networks, as they are capturing nodes with a lot of connections. However, the ego network of the high betweenness centrality node is serving as more as a bridge between densely connected parts of the network.

1.2 Introduction to community detection

While centrality measures allow for the prioritization of individual nodes in the network, it is also useful to look at sets of similar nodes in terms of how they are situated in the network. Each of these sets of similar nodes is known as a *community*. A community in a network is broadly defined as a set of nodes of who share something in common in terms of their connectivity patterns in the network. One can think of a community as a clustering problem on networks, where the objective is to define sets of nodes that maximize the within-community node similarity. The most basic type of community to understand is a network with assortative community structure. In this case, nodes are tightly connected to each other but more sparsely connected to the rest of the network. An example of a network with assortative community structure is shown in Figure 1.6. Communities in the network are outlined with pink dotted lines.

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

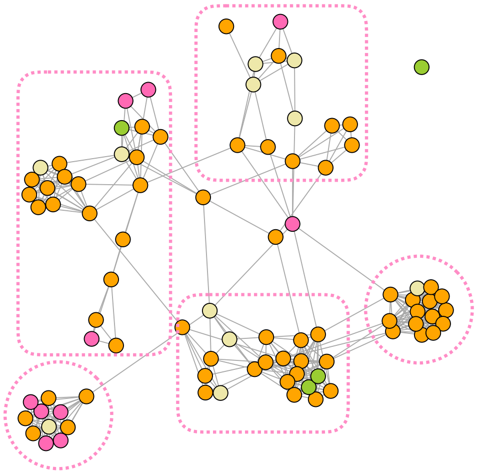

Figure 1.6: **Assortative Community Structure.** This network is an example of assortative community structure, where nodes are tightly connected to each other and more sparsely connected to the rest of the network. Each community is outlined with a pink dotted line.

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

In this section, we have only briefly introduced the history and intuition behind community detection. Since it is a well-developed domain of network science, the interested reader can refer to one of the comprehensive review articles (75; 51; 127; 94)

1.3 Community detection methods

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

1.3.1 Notation for Community Detection

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

1.3.2 Quality function maximization with modularity

In quality function optimization approaches one first specifies an objective function in terms of a partition of the nodes. The most common quality function for this task is known as modularity (100). Modularity first defines a null model for community structure where edges are places between groups randomly. With this as the starting point, the partition that optimizes modularity is the one that is maximally different from this null model. In particular, this null model is a random graph model, known as the configuration model (20). To generate an N-node network from the configuration model, one first specifies a fixed degree sequence, $D = \{k_i, k_2, \ldots, k_N\}$. From this sequence, nodes are connected with k_i stubs that will ultimately be connected together. Finally, the graph is constructed by randomly choosing pairs of the created stubs and joining them. Based on how this network was generated, it is easy to specify the probability that an edge exists between a pair of nodes, i and j, or $P(a_{ij} = 1)$.

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

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

Modularity was introduced in 2004 by Newman and Girvan (103). 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 (122) that controls the scale of community size. Large values of γ favor more small communities while smaller values enforce fewer large communities.

In order to determine $\mathbf{z}$, the most computationally efficient approach is known as the Louvain algorithm (23). The Louvain algorithm is an agglomerative heuristic, which initially starts with each node in its own community and in the first pass merges pairs of nodes if their merge leads to an increase in modularity. Each group of nodes assembled after this first pass becomes a new node in the network and a new weighted network is created between the set of new nodes. The weight on the edges of the new network are the number of edges from the original network that go between the sets of merged nodes. This process is continued iteratively until the modularity no longer increases. The reason that this approach is so computationally tractable is because the gain in modularity, ΔQ of merging two groups of nodes can be explicitly computed in closed form.

Modularity has shown to be effective in applications from neuroscience (92) to image segmentation (29). It has also shown to be effective in clustering high dimensional data that has been used to create a network. In Figure 1.7, we used tSNE (87) to project the 52-dimensional single cell data into 2 dimensions. Points are colored by their cluster assignment according to k-means. We first performed k-means on the original 51 dimensional data (left) and Louvain community detection on the 5 nearest neighbor network representation (right). One benefit of the Louvain algorithm is that it does not require specifying the number of clusters. Moreover, in this example, the Louvain algorithm maximized modularity by partitioning the network into 10 clusters. To compare the results under the same number of clusters, we also clustered the original data into 10 clusters. From these two partitions, we observe that creating a network representation of the data before clustering assists in identifying the smaller, less prominent clusters.

1.3.3 Identifying communities with probabilistic approaches

Probabilistic community detection methods aim to find a partition of the network through likelihood optimization. Intuitively, the goal is to study the generative process of the node edges in terms of the inferred community assignments. For example, given nodes i and j, one may model $P(a_{ij} = 1)$

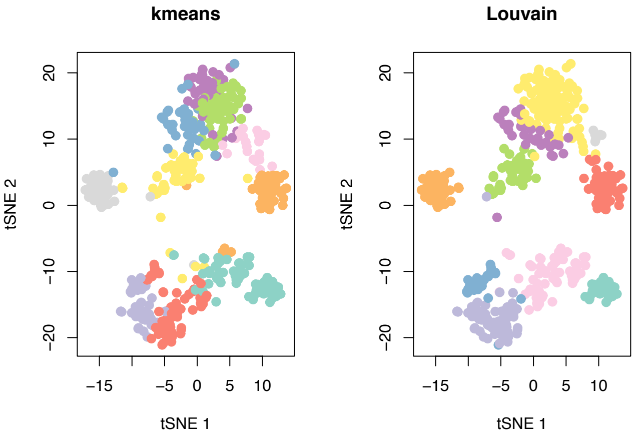

Figure 1.7: **A comparison of k-means and the Louvain algorithm on the single cell network.** A comparison of the results of clustering results on the the single cell dataset through k-means on the original 52-dimensional data (left) and by the Louvain algorithm on the nearest neighbor network (right).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

as $g(z_i, z_j)$, where $g(\cdot)$ is some rule based on the node-to-community assignments. Two common probabilistic community detection models are the stochastic block model (130) and the affiliation model (151). The definition and description of these models and inference techniques are described in depth in this section. To facilitate working with probabilistic models, we first introduce some notation and background on inference techniques.

1.3.3.1 Probabilistic graphical models for statistical inference

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

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

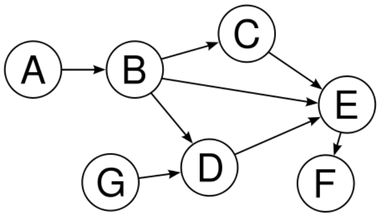

Figure 1.8: **Directed Acyclic Graph.** A directed acyclic graph (DAG) is formed based on dependency between random variable and allows for a fully factorized probability distribution. Nodes represent random variables and a directed edge from node i to node j indicates that node j depends on node i.

models (72) enable efficient specification and manipulation of large probability distributions through semantic structures.

As a brief example, given a set of random variables, $\{A, B, C, D, E, F\}$, we seek to compute the joint distribution, $P(A, B, C, D, E, F)$. This joint distribution can be expressed with a directed acyclic graph (DAG), whose structure encodes dependencies between random variables. The DAG allows for the representation of the joint distribution in a factorized way, which is computationally useful. A DAG between the set of random variables, $\{A, B, C, D, E, F\}$ is shown in Figure 1.8. Each node in the graphical model represents an random variable and a directed edge from node i to node j implies that node j depends on node i.

To translate a DAG between a set of N random variables, $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ (also in this context referred to as a Bayesian network) to its joint distribution, we rely on the chain rule for Bayesian networks (72), which specifies that a DAG factors according to its parent/child relationships with,

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

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

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

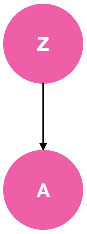

Figure 1.9: **SBM Graphical Model.** A graphical model is used to model the dependency between the node-to-community assignments, $\mathbf{z}$ and the observed network adjacency matrix, $\mathbf{A}$.

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

1.3.3.2 Stochastic Block Model

In this section, we introduce the most popular probabilistic model for community structure, known as the Stochastic Block Model (131). This model is popular and has been studied extensively, due to its simplicity and intuitive interpretation. The crucial assumption of the stochastic block model is that nodes within a community are connected to nodes within their community and to other communities in a characteristic way. For an undirected, unweighted network with adjacency matrix $\mathbf{A}$, we seek to partition each of the N nodes into one of K communities. We denote the the node-to-community assignments as $\mathbf{z}$, with z_i specifying the community assignment of node i. Here, $\mathbf{z}$ is a latent variable, with each entry taking on 1 of K states, or one of K community assignments. Figure 1.9 shows the dependency relationship between the node-to-community assignments ($\mathbf{z}$) and the network's adjacency matrix ($\mathbf{A}$). Here, the node-to-community assignments are treated as a latent variables because we seek to identify the $\mathbf{z}$ that makes the observed adjacency matrix, $\mathbf{A}$ the most likely. To model the objective that members within and between communities connect in characteristic ways, the model fitting procedure requires learning a set of within and between community connection probabilities. Under this approach, edges are treated as independent and identically distributed and deciding whether or node an edge exists between a pair of nodes is the learned connection probability between the communities to which each of the nodes belong.

Using the factorization rules described in section 1.3.3.1, we can specify the complete data log likelihood between $\mathbf{z}$ and $\mathbf{A}$ as,

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

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

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

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

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

First,

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

Next,

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

Optimizing the parameters of this incomplete data log likelihood requires computing the posterior $P(\mathbf{z} \mid \mathbf{A})$ but unfortunately is intractable, as shown by Daudin *et al.* (37). To address this issue, the posterior can be recast using a factorized approximation. This is accomplished by optimizing a lower bound of $\mathcal{L}(\mathbf{A})$. We let $\mathcal{R}_A$ be an approximation of the posterior, $P(\mathbf{z} \mid \mathbf{A})$. To optimize the lower bound of $\mathcal{L}(\mathcal{A})$, we seek the $\mathcal{R}_A$ that is as close as possible to $P(\mathbf{z} \mid \mathbf{A})$. In other words, we define the lower bound of $\mathcal{L}(\mathbf{A})$ as $\mathcal{T}(\mathcal{R}_A)$, with,

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

Here KL denoted the Kullback-Leibler divergence (KL divergence) and the best approximation will be the value that makes the KL divergence the smallest. Jaakkola *et al.*, present a mean field approximation for the posterior distribution (65) as,

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

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

Daudin *et al.* (37) , show that the optimal estimate for τ_{ik} denoted $\hat{\tau}_{ik}$ satisfies

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

Here, α_k notes the probability that a node belongs to community k. Furthermore, after computing the set of variational parameters, the updates for $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ that maximize $\mathcal{T}(\mathcal{R}_A)$ are also shown by Daudin *et al.,* (37) to be,

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

We have presented this variational approach for performing SBM parameter inference and likelihood optimization because this approach was appropriate for the work presented in this thesis. Variational inference is just one approach that can be applied to learn model parameters and was but a study by Zhang *et al.* (160) also show that belief propagation (96) is very effective for this task. Briefly, belief propagation is a message passing algorithm for parameter inference in probabilistic graphical models. Given that parameter learning offer requires computing marginal distributions for a set of variables with a very large number of possible configurations, belief propagation uses the graphical model to reduce the complexity of the problem.

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

1.3.3.3 Variants to the Classic Stochastic Block Model

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

Edge Weights

The majority of the stochastic block model literature considers unweighted networks simply because describing a probabilistic model to handle both edge existence and edge weight is a challenging task. In the classic stochastic block model, we are simply modeling whether an edge exists based on the inferred community memberships of the edge stubs. Since edge weights can come in a variety of forms (real-valued, count, etc.), it is difficult to immediately decide what distribution the edge weights should follow. In the past few years, this issue has been tackled in two papers (9; 114).

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

To avoid having intuition about K, Peixoto (114) developed a non parametric bayesian approaches that is capable of inferring K with no prior knowledge. The assumption of the model is also slightly different and assumes a hierarchical structure between communities. The inference is achieved through Markov Chain Monte Carlo (MCMC) sampling.

Degree Heterogeneity

Based on the variety of network structures and types, the assumption that the classic stochastic block model is an appropriate model for even classic unweighted data is often invalid. That is, for some networks, the fitted model may not actually be a good fit, in that samples from the learned model are substantially different from the network. Work by Karrer *et al.* (68), introduced a simple extension to the classic stochastic block model, known as the degree corrected stochastic block model. This model is informed by degree distribution as a proxy for the network structure. In networks where there is a wide degree distribution (i.e. many high degree nodes and many low degree nodes), stochastic block models inference tends to partition the nodes into communities of high degree and low degree nodes. The approach for adapting the SBM to this setting is to slightly modify the learned $K \times K$ matrix, $\boldsymbol{\pi}$ slightly. Here, π_{ij} described the number of edges between nodes i and j. Further, these edges counts are modeled as poisson random variables. The likelihood of the observed network under this poisson assumption takes into each node's degree.

The restriction of single community membership

As it is often observed in social networks, the assumption that every node belongs to only a single community is restrictive. To address this issue, approaches have been developed to allow nodes to participate in a mixture of communities (11) or be members of overlapping communities (79). Airoldi *et al.*, pioneered the development of the mixed membership stochastic block model (11), where instead of modeling a node's membership in each community in a binary manner, the authors allow a node to belong to multiple communities. The generative process for this approach for modeling the existence of an edge between nodes p and q in a network with K possible communities and the $K \times K$ matrix, $\boldsymbol{\theta}$ representing the between community connection probabilities.

- For each node p, draw a mixed membership vector $\pi_p \sim$ 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.* (79) addressed an important limitation of (11). Since the probability of an edge between a pair of nodes p and q depends on a single draw of $\mathbf{z}_{p \to q}$ and $\mathbf{z}_{q \to p}$, the class memberships of nodes p and q towards other nodes in the network are ignored. Moreover, this model adapts the mixed membership stochastic block model to incorporate a higher order resolution of structure by considering each node in the context of its neighbors.

1.3.3.4 Affiliation model and inference

We have previously discussed extensions of the stochastic block model that account for the assumption that nodes can belong to multiple communities. Another interesting idea is the idea of *pluralistic homophily*, where the probability that two individuals are connected is related to the affiliations of the nodes (49). In other words, the more groups a pair of nodes share, the more likely they are to have a connection. For example, if two two people were graduate students, studying computational biology at the same university, they are more likely to be connected than a pair of graduate students studying different subjects at the same university. A state-of-the-art method called BIGCLAM was presented for this task by Yang *et al.* in 2013 (152). The objective here is to model the connection probability between a pair of nodes based on the similarity in their learned affiliations towards communities. To do this, individual nodes are connected with communities with some number of links, with more links from a node to a community indicating that the node has a higher 'affilation' to that group. For a network with N nodes and c communities, the affiliation between nodes and communities is encoded by a matrix, **F**, where F_{uc} is the learned count of links (again encoding the affiliation), between node i and community c. Similarly, let F_u and F_v be the community affiliations for nodes u and v. Then the probability that an edge exists between nodes u and v, or $P(A_{uv} = 1)$ is modeled as,

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

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

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

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

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

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

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

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

1.3.4 Deep Learning Approaches

In recent years, deep learning has begun to revolutionize many fields, including network analysis. Perozzi *et al.*, pioneered the use of deep learning in community detection with the development of DEEPWALK (118) to learn a latent space representation of nodes in some d-dimensional Euclidean space (i.e. an emedding). Once the network is embedded in a lower dimensional space, simple clustering techniques, such as k-means (61) 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 (106; 56). A random walk on a network involves choosing a starting node and traversing the network by hopping between adjacent nodes. The DEEPWALK approach seeks to learn an embedding of the nodes that preserves the sets of nodes traversed in a random walk. To do this, the authors adapted the Word2Vec approach to this context. Word2Vec is a powerful tool from natural language understanding that allow for the specification of a node embedding that enables accurate prediction of a word's context, given the word (93). To adapt this context to networks, a random walk is treated as a sentence and nodes are treated as a word within the sentence. Moreover, the analogous task to the problem in text data to a network is to accurately predict a set of nodes likely to be seen on a random walk with the node of interest. The embeddings should preserve these learned rules. Moreover, this problem is solved using the same optimization approach as Word2Vec

Based on the success of DEEPWALK, the method was followed up with Node2Vec in 2016 (58). While node2vec also uses the random walk framework to specify the optimization problem, they modify how the random walk is performed to enable an embedding that captures different aspects of a potential network community. For example, one may describe a community by a set of nodes located close to each other in the network with many common neighbors and connections to common neighbors. This assumption is known as network homophily (73). 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 (86). For example, a community under structural equivalence might group could group nodes with similar degree or centrality. To modify the random walk so that it leads to a model that gives flexibility in the nature of retrieved communities, the authors introduced a search bias term, which controls whether the random walk in performed in a breadth-first or depth-first search parameter. If on a random walk, the path is traversed in a depth-first search, favoring the exploration of a larger area of the network far from the random walk source, the resulting community aligns with the homophily hypothesis. A random walk performed in a breadth first manner that restricts the path to nodes neighboring the source and tends to capture nodes based on structural equivalence (i.e. a hubs, high degree nodes, or low degree nodes).

1.3.5 Higher order network analysis

One of the most recent advances in community detection is clustering nodes based on 'higher order' features, or at the level of small subgraphs or *motifs*. The structural organization of the network is then interrogated by identifying clusters of network motifs. The flexibility and appeal of this framework is that different kinds of organizational patterns of the network can be identified, depending on the motif used. Seminal work using this approach was proposed by Benson *et al.* (21) . Given a motif, M, higher order clustering frameworks seek to identify a cluster of nodes S that minimize the following ratio,

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

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

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

2. Compute the spectral ordering (i.e. order the eigenvalues in descending order), δ, of the nodes from the normalized motif Laplacian of W_M. Note this motif Laplacian is the standard Laplacian matrix for W_M (91).

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

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

1.4 Community detection in computational biology

Analyzing networks with community detection has shown to be fruitful, particularly in biology and neuroscience applications. In this section, we will describe examples of analyses where the identification of communities provided insight and understanding for a biological problem.

Multiple experimental modalities exist that enable the collection and analysis of biological data. Understanding protein expression, gene expression, microbiome composition, metabolomic profiles, genomic mutations, and immune profiling are just a few of examples of biological data that is studied routinely for insight into human health. With most experimental platforms producing high dimensional data, it is crucial to have good tools for interpretation, visualization, and prediction. Machine learning techniques in computational biology have revolutionized prediction in biology and medicine (38; 12; 84). In this section, we outline particular examples of how the application of community detection to diverse types of biological data lead to improved scientific understanding and predictive ability.

1.4.1 Immunological profiling to establish a pregnancy immune clock

A study lead by Aghaeepour *et al.*, demonstrated that there is a typical timing of immunological events in a healthy, term, human pregnancy (8). Immunological profiling was performed on a training cohort of 18 women, using a technology called mass cytometry (19) was used to quantify various features of the immune system, such as, cell type abundances and signaling activity. A correlation network between the measured set of immune features in the training cohort was constructed to develop hypotheses about their interactions. In addition to the construction of the network, an elastic net regression model (162) was trained to identify immune features associated with increased gestational age. When communities were identified in the correlation network of the immune features, there were two important observations. First, immune features of the same type (i.e. cell signaling vs. cell frequency) were aligned with community labels. Second, sets of features associated with a particular gestation age often fell in the same community, indicating their synchronous activity during a particular time in the pregnancy. Finally, after identifying influential nodes in their ability to predict stage in pregnancy, according to the regression model, the communities of these nodes were more closely examined to uncover further insight into the immunological mechanisms occurring throughout the pregnancy time course.

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

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

The microbiome refers to the collection of bacterial species that populate an organism's gut. Microbiome analysis has recently gained attention, as its biological implications are large for health and disease (129). 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 (80). A network in this context is typically constructed based on some notion of co-occurence or correlation between microbial species, profiled across samples. A recent example where community detection played a key role in the biological understanding was introduced in 2017 and assessed the interplay between microbial co-occurence structural organization patterns between patients with and without inflammatory bowel disease (14). Communities were identified in the healthy and diseased networks, using classic modularity maximization (55). After identifying a community structure for each network, the similarity of these partitions was quantified with the Rand index (139), which showed to be statistically significant under a permutation test. This observation allowed the authors to understand that the core structure from a healthy microbiome was conserved even in diseased patients, but allowed for more careful probing of the subtle differences. First, the functional roles of the members in each community were interrogated. Some interesting co-occurence relationships within communities were identified, such as the loss of strong clustering, or association propensity between pro and anti-inflammatory species within the diseased networks. This interplay between pro and anti inflammatory species is thought to play a pivotal role in the maintenance of a healthy gut microbiome.

Next, the authors used the community structure of each network to study the differences in node roles (i.e. importance) between the healthy and IBD networks. Within the neuroscience community, there have been numerous efforts to characterize nodes, in terms of the role they play connecting communities or as an important node within a community (144). 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 (116). While this process was historically performed manually, there has been a significant amount of work to develop computational methods that can successfully segment cell populations, automatically (7). A network-based approach to this problem, known as SamSPECTRAL was introduced in 2010 by Zare *et al.* (158). In this approach, the authors seek to segment a population of cells into distinct subpopulations, through the construction of pairwise single cell similarity network. After constructing this network, communities detection can be applied to cluster the cells into sub populations. To recap, in this network, the nodes are comprised of the individual cells within a sample, and edges between nodes indicate the similarity between a pair of cells, based on the quantification of their emitted light. Another useful feature of SamSPECRAL is that it also does some preprocessing to reduce the size of the constructed network.

To create a network of the flow cytometry data, a large subset of data points (cells) are first sampled and denoted as ‘registered’ nodes. The next step is to look at the collection of ‘unregistered nodes’ and ultimately assign them to their closed registered node neighbor. Iteratively, the unregistered nodes are attempted to be registered or agglomerated with the set of registered nodes. For example, for one of the registered nodes, which can be denoted as p, the set of unregistered nodes within some defined distance h become registered to p. The set of unregistered nodes that were newly assigned to be registered are removed from the set of unregistered nodes. This process is repeated until there are no more unregistered nodes. Each set of nodes registered with the same label are denoted as a community (an inconvenient label, given a network will be constructed and communities will be identified). A weighted network is constructed between these registered communities with edge weights quantifying the similarity in the quantitative features (as quantified through flow cytometry) between a pair of a communities. Once this weighted graph is created, a spectral community detection method (149) is applied to segment the network into 1 of K network communities. These is one final post-processing step, motivated by previous work in computational flow cytometry methods, to combine the agglomerate a pair of network communities if members if the community show similarity greater than a predefined threshold (in terms, again, of their measured flow cytometry properties). The usefulness of this approach is that it exhibited outstanding performance on datasets containing clusters of challenging shapes. Some examples of challenging shapes are, overlapping clusters, non-elliptical shaped clusters, or low-density clusters. Ultimately the retrieved population of cells avoided manual segmentation and provided a solution to a computationally challenging task.

1.4.4 Understanding genetic diversity of the malaria parasite genes

Rich genetic diversity in the *var* genes of the human malaria parasite has been shown to contribute to the complexity of the epidemiology of the infection and disease. The parasite can change which of the *var* genes are expressed at any given time on the infected red blood cell, which prevents the antibody from recognizing and resisting the new protein. One diversity-generating mechanism is recombination, which is the exchange and shuffling of genetic information during mitosis and meiosis (16). The ability to understand genetic diversity is complicated by inadequate tools to uncover the phylogeny, or genetic relationship between sequences resulting from recombination events, in a scalable and statistically rigorous way. The typical analyses for evolutionary data assume a tree-like relationship between events, which is unrealistic for recombination data. To address this challenge, (78) 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 (69).

After identifying communities within each HVR network, the authors used two summary statistics to formulate their biological hypothesis. First, the variation of information (125) 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* (99) 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, (78) 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 (83). Single cell technologies allow for the profiling of cells individually within a sample. Recent attention and methods development have focused on the use of RNA sequencing and mass cytometry for high dimensional profiling of single cells. Single cell technologies have enabled for an advancement in the understanding of the pathobiology of cancer in that cells within a tumor have been shown to exhibit a large amount of heterogeneity at the single cell level. Furthermore, this heterogeneity has important functional and clinical significance (89). 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 (137).

Overall, the findings of this paper use community detection to allow for the analysis and understanding of tumor heterogeneity data that was not possible with standard high dimensional data analysis techniques. The authors suggest that this method is useful in characterizing primitive cancer cells and for the identification of cell biology features that define particular biological states and clinical outcomes.

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

Urinary tract infections are primarily caused by uropathogenic *E. coli* (UPEC). In their study Parker *et al.,* seek to better understand UPEC antibiotic resistance, which prevents patients from being treated for urinary tract infections. Using a cohort of 337 *E. coli* patient isolates, the authors looked closely at the virulence factor genes of these patients. Virulence factors are non conserved or are carried on mobile genetic elements and elicit biological functions that relate to uropathogenesis (i.e. the onset of a patient getting at UTI). The biological function of virulence factors are known and allow for the development of therapeutic agents. In the analysis, the presence or absence for each of

16 virulence factors was determined. A network was constructed between the 337 patient isolates, with each edge reflecting the pairwise similarity in their virulence factor profiles. Modularity based community detection was then applied to this network and partitioned it into 4 different communities. Most remarkably, each of the 4 communities was characterized by clinical isolated described by either a single or pair of virulence factor markers. These pairs of related virulence factors were then probed further to investigate their role in antibiotic resistance. This approach offers a n ew way to integrate genomic and individual patient information to determine which types of antibiotics might be most effective.

1.5 Thesis Contribution

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

1.5.1 Thesis Statement

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

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

1.5.2 Summary of the novelty of this work

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

Method	Brief Description	3 Similar Approaches	Novelty
sMLSBM (Chapter 2)	Stochastic block model for multilayer networks	MLSBM through aggregation methods (143), Restricted MLSBM (108), MLSBM (60)	Instead of fitting a consensus SBM to all layers, we learn a set of models that represents different clusters of layers.
SuperNode (Chapter 3)	Pre-processing a network into a smaller network of super nodes before applying community detection	Identify communities on core of network and propagate labels outward (115), Create super nodes with prior information about nodes expected to be together (155), Reduce the size of the network by systematically removing nodes and edges (54)	We recast the entire network as a network of super nodes and input this smaller version into community detection algorithms. We do not require prior knowledge or side information
Attribute SBM (Chapter 4)	Incorporate both network connectivity and continuous attributes into account when assigning nodes to communities	SBM inference with a single attribute (102), iLouvain Modifying modularity to incorporate attributes (33), CESNA: Affiliation model with attributes (154)	We adapt the SBM to handle multiple continuous attributes
Attribute Alignment Test (Chapter 5)	A test to quantify the extent to which attributes and connectivity align.	neoSBM + BESTest (attribute augmented SBM) (110). To our knowledge, there are not any related tests other than (110)	We use label propagation to generate an empirical p-value that quantifies how the attributes relate to connectivity. We do not require assuming that a stochastic block model is an appropriate representation of the network.

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

1.5.3 Relevant Publications

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

Community detection in multilayer networks

1. *Clustering Network Layers with the Strata Multilayer Stochastic Block Model*. **N. Stanley, S. Shai, D. Taylor, P.J. Mucha**. IEEE Transactions on Network Science and Engineering. 2016. http://ieeexplore.ieee.org/abstract/document/7442167/

2. *Enhanced Detectability of Community Structure in Multilayer Networks Through Layer Aggregation*. **D. Taylor, S. Shai, N. Stanley, P.J. Mucha**. Physical Review Letters. 2016. https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.116.228301

Community Detection (General)

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

Pre-Processing for Community Detection in Large Networks

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

Community Detection for Attributed Networks

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

1.5.4 Software

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

1. **sMLSBM: Fitting a multilayer stochastic block model**. https://github.com/stanleyn/sMLSBM

2. **SuperNode: For compressing a large network**. https://github.com/stanleyn/SuperNode

3. **Attributed SBM: For fitting an SBM with multiple continuous attributes**. https://github.com/stanleyn/AttributedSBM

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