TESTING-BASED COMMUNITY DETECTION METHODS FOR COMPLEX NETWORKS

John J. Palowitch

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 Statistics and Operations Research.

Chapel Hill
2017

Approved by:

James S. Marron

Yufeng Liu

Peter J. Mucha

Shankar Bhamidi

Andrew B. Nobel

ABSTRACT

JOHN J. PALOWITCH: Testing-Based Community Detection Methods for Complex Networks
(Under the direction of Andrew B. Nobel)

Community detection is an exploratory method of grouping strongly connected nodes in a network, in most cases using only the network edge structure as a guide. Using discovered communities for downstream analyses can be crucial for real-world decision-making and inference. Recent approaches to community detection include *testing*-based community *extraction*, a process in which communities are refined one-by-one via analysis of graph statistics. However, to date, testing-based extraction methods are tied to the configuration model as a null, which applies only to single-layer, binary graphs.

In this thesis, testing-based extraction is generalized to arbitrary networks types with a framework called Node-Set Testing (NST). The NST framework defines the broader statistical elements of an approach that uses hypothesis testing to detect communities in complex networks. The NST framework is applied to (i) weighted networks and (ii) bipartite correlation networks, resulting in novel community detection algorithms. In particular, new null models and test statistics are specified to apply iterative hypothesis-testing algorithms on these types of networks. Detailed analyses of the empirical and theoretical properties of the proposed methods are provided.

Other chapters in this thesis, while not explicitly involving *testing*-based algorithms, support the discussion of community detection in heterogeneous networks. One chapter provides a consistency analysis of a significance-based score for community extraction in multilayer networks. In another chapter, preceding the discussion of the NST method for bipartite correlation networks, an application area called eQTL analysis is discussed. In particular, a new model for estimating the effect size and regression correlation of the links in an eQTL network is introduced and studied.

To Marilyn Maiden Palowitch and Carl Joseph Palowitch, my parents;
to whom I attribute my curiosity, dedication, and enthrallment with understanding;
and who, were it not for the degree that led to this thesis, would probably be
a few years closer to having grandchildren.

ACKNOWLEDGEMENTS

I have an enormous amount of gratitude for my collaborators and their professional and personal contributions to the work presented in this thesis. First and foremost, I thank my advisor Dr. Andrew B. Nobel. Dr. Nobel's near-endless energy and enthusiasm, combined with a genuine regard for and honest criticism of all my ideas, created an environment in which my needs for creative exploration and rigorous training were well-satisfied. I also thank my secondary advisor Dr. Shankar Bhamidi. Dr. Bhamidi continually challenged me to think more intuitively about complex problems, and provided me with fantastic advice about the relationships between research directions, career goals, and life endeavors.

I thank Dr. James S. Marron for a year's worth of immersion in his diverse and interesting research directions. I thank Dr. Perry Haaland under whom I worked a semester-long internship at Becton-Dickinson in Research Triangle Park. My collaboration with Dr. Marron and Dr. Haaland was an indispensable part of my growth as a statistician and data scientist. I am especially grateful for the presence of Dr. Peter J. Mucha and Dr. Yufeng Liu on my committee, as Dr. Mucha is technically on sabbatical this semester, and Dr. Liu is a new parent (congratulations!). Over the past year, I have met with Dr. Mucha and one of his postdoctoral researchers Dr. Dane Taylor to discuss interesting new directions in network science: I have greatly appreciated these conversations.

A large chapter in this thesis is devoted to my work on the Genotype Tissue Expression (GTEx) project. Along with Dr. Nobel, the other P.I. on our working group was Dr. Fred A. Wright, the Director of the Bioinformatics Research sCenter at NCSU. Dr. Wright's approach to research was a continual source of inspiration. His ways of thinking about modeling and significance were (and remain) formative to my statistical intuitions. Also in our working group was Dr. Andrey Shabalin, assistant professor at VCU. Dr. Shabalin was the main contributer to the software implementations of our new eQTL model, and was an indispensable collaborator in the refinement of the model's theoretical underpinnings. I am grateful for the opportunity to have worked alongside Andrey, as his experience and advice greatly improved my ability to think systematically about programming

and statistical computation. I thank Dr. Yihui Zhou, research assistant professor at NCSU, who contributed a thorough and somewhat self-contained simulation analysis of our method. Though it does not appear in this thesis, her analysis was crucial to the publication viability of our paper.

Lastly, I thank some of my peers in the STOR department. Soon-Dr. Kelly Bodwin was my research co-conspirator during the past four years. I can say without a doubt that many aspects of projects in this thesis were improved because of our conversations. Kelly shared many of my attitudes toward and approaches to research, making her camaraderie an indispensable part of my degree. The entirety of Chapter 2 is joint work with Kelly. Dr. James Wilson, a graduate of this department and now tenure-track assistant professor at USF, was co-first-author with me on the paper containing the work featured in Chapter 4. The work presented in Section 4.3 is primarily his contribution. During his time here, conversations with James and collaborations on simulations spurred on my thoughts about both the consistency of community extraction methods, and the general testing-based approach in Chapter 2. This past year, Miheer Dewaskar, a first-year STOR graduate student, joined the project discussed in the "future work" portion Chapter 5, and has contributed to computational aspects of the method introduced therein.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

n	Sample size of a network or Euclidean data set.
$[b]$	For a positive integer b, the index set $[b] := 1, \ldots, b$.
u, v	Indices for nodes, e.g. $u, v \in [n]$.
$\mathcal{C}$	A node-set cover: $\mathcal{C} := \{C_1, \ldots, C_K\}$ with $C_j \subset [n]$ for all $j \in [K]$.
A	Adjacency matrix. $A[u, v] = 1$ if and only if u and v are connected.
$\mathcal{G}$	A network $\mathcal{G} = ([n], A)$.
$v(u)$	The u-th component of an n-vector $\mathbf{v}$
v_T	Denotes the vector sum of $\mathbf{v}$, defined $v_T := \sum_{u \in [n]} v(u)$.
$\mathbf{d}$	The vector of node degrees, where $d(u) := \sum_{v \in [n]} A[u, v]$.
W	For weighted networks, the adjacency matrix of weights.
$\mathbf{s}$	The vector of node strengths, where $s(u) := \sum_{v \in [n]} W[u, v]$.
NST	Node-Set Testing
SCS	Stable Community Search
SBM, DCSBM	(Degree-Corrected) Stochastic Block Model
CCME	Continuous Configuration Model Extraction
ACME-eQTL	Additive-Contribution, Multiplicative-Error model for eQTL analysis

CHAPTER 1
Introduction

A network can be both a mathematical structure, having an abstract set of nodes and edges, and a natural phenomenon, consisting of objects and their interactions. Sometimes there is, in some sense, an isomorphism between a physical network and its mathematical representation. Road networks, electric networks, and internet networks all have a set of physical, fixed nodes and links that can be identified with a binary graph. Analyses of these networks often involve questions about logistics and operations, like finding efficient ways to schedule travel routes, packet transfers, or computing jobs.

Most often, however, natural systems are only *represented* by a mathematical network. For instance, a gene interaction network is a set of genes and their regulatory relationships. The existence of such networks are acknowledged through a build-up of statistical evidence for correlations between genomic regions. However, gene networks do not necessarily correspond to biophysical pathways. In general, networks that *represent* a system are studied in a more scientific manner. Nodes are treated as data objects, and their emergent relationships seen as stochastic, potentially indicative of underlying group dynamical structure in the system under analysis.

The topics in this thesis focus on the scientific and statistical analysis of real-world systems through network data objects. In particular, the statistical methods this work introduces apply to *complex* networks. The term "complex network" initially arose when natural networks of interest became large and quite unlike standard random graph models of the mid-to-late twentieth century. Hence, the study of complex networks is defined by a focus on large-scale, heterogeneous properties of networks, like topology, clustering, and degree distributions (Albert and Barabási, 2002).

Another facet of the study of complex networks is the increasing availability of a wide variety of network data attributes (Newman, 2003a). Many networks of interest have weighted, labeled, signed, or directed edges. Others feature *layers* of networks, each layer with a registered node set, but with an edge structure corresponding to a unique experimental unit or a point in a time series.

Still others have hierarchical levels of sub-networks. Each separate data attribute in a network carries rich information of potential import to any analysis question (Boccaletti et al., 2006). As such, methods introduced in this thesis will have a particular focus on networks with more-than-one type of network data.

Myriad descriptions of studies involving networks with more than one attribute are given in the aforementioned references; here are just a few more notable examples:

1. Almaas et al. (2004) studied the molecular reaction network of the *E. coli* metabolism, with edges weighted by the reaction fluxes, providing deeper insights into bacterial metabolic organization and regulation.

2. Barrat et al. (2004) analyzed the collaboration network of authors contributing to the condensed-matter physics electronic archive between 1995 and 1998. The network contained $n = 12,722$ authors, with edges weighted by number of collaborations. Their analysis revealed, quantitatively, some intuitive findings about the relationship between academic collaborative *structure* and *volume*, for instance that highly published authors produce a plurality of their total output with a stable research group.

3. Ansari et al. (2011) developed flexible parametric models for multi-layer networks. In their work, they analyzed network data from a Swiss social media platform that connects musicians with listeners as well as with other musicians. The network data had three layers, each with different attributes: a binary, undirected friendship layer; a binary, directed messaging layer; and a weighted, directed music downloading layer. They showed that modeling diverse attributes separately, accounting for specific data types, greatly improved predictive performance.

Efforts to extend network data analysis techniques to incorporate all data attributes in networks like those mentioned above have comprised significant arms of network science methodology research (e.g. Newman (2004a) and Opsahl and Panzarasa (2009) for weighted networks, Mucha et al. (2010) and Kivelä et al. (2014) for time-series and multilayer networks, Battiston et al. (2014) for weighted and un-weighted multi-layer networks, Leskovec et al. (2010a) for signed, directed networks). In general, methods for complex networks have driven advances in areas as diverse as social science,

systems biology, life sciences, marketing, and computer science (cf. Palla et al. (2007); Barabasi and Oltvai (2004); Lusseau and Newman (2004); Guimera and Amaral (2005); Reichardt and Bornholdt (2007a); Andersen et al. (2012)). Surveys of the network science and methodology literature have been provided by Newman (2003b) and Jacobs and Clauset (2014), among others.

The primary, specific focus of this dissertation is community detection methods for complex networks, in particular networks with data attributes beyond binary edges. A common, loose definition of a community is a subset of nodes that are more connected internally than externally. Community detection is one of the most important network analysis techniques, as communities often reflect intrinsic structure of the system the network represents. Finding communities among objects of study in a data set can support exploratory analysis and provide an important starting point for further scientific inquiry or decision-making (Danon et al., 2005). For example:

1. Hundreds of studies in computational biology have used community detection to find subsets of genomic loci, genes, cells, or micro-organisms with high levels of interaction. These subsets can be focal points for a variety of downstream analyses, and can spur new insights about the underlying mechanics of genomes (Chen and Yuan, 2006; Cabreros et al., 2015; Platig et al., 2015; Fan et al., 2012).

2. Community detection has been used to facilitate recommender systems in online social networks: first by grouping users through their observed edge (or edge weight) interactions, and then by suggesting interests of users to the rest of their group (Sahebi and Cohen, 2011; Xin et al., 2014).

3. To study buyer patterns on eBay, community detection was used to group bidders based on common bidding interests (Reichardt and Bornholdt, 2007b) and to group auctions based on common bidders (Jin et al., 2007). The latter analysis, in particular, was used to predict and recommend equally valuable auction items to bidders (Fortunato, 2010).

Countless other examples of community detection applications can be found in surveys by Porter et al. (2009), Fortunato (2010), and Fortunato and Hric (2016), and the references therein.

Most standard community detection methods involve the optimization of a quality function for network partitions. For some methods, the quality function is a log-likelihood; for other methods,

it is a more general score of the intra-connectedness and inter-disconnectedness of the partition's elements. Various quality functions will be discussed in more detail in Section 1.2. Optimization approaches to community detection have proven extremely effective for decades, in countless applications. However, the results of these methods rarely come with a significance guarantee or statistical interpretation. Basic simulations will show that many commonly-used community detection methods reliably find communities in networks that (arguably) lack community structure.

This thesis introduces and analyzes applications of a *testing-based* framework to community detection on networks with (potentially) multiple data types. The framework re-motivates community detection from a statistical testing perspective, providing an iterative hypothesis-testing algorithm for the discovery of significantly associated node sets. This approach is contrasted with optimization-based community detection, an approach which (in general) does not provide guarantees of statistical significance. As the major components of the work presented here, new testing-based algorithms are derived for various types of networks. Illustrations of these methods' advantages, empirical efficacies, and theoretical results regarding consistency and error control are provided. In Section 1.4, more introductory detail is provided about these contributions. The sections below provide an in-depth review of existing community detection methodology.

1.1 Preliminary Notation

I denote a general network object with n nodes by $\mathcal{G} := ([n], \mathbf{D})$, where $[n] = 1, \ldots, n$ is the node set, and $\mathbf{D}$ is a data object which encodes the observed interactions between nodes. We call $\mathcal{G}$ a "binary" or "un-weighted" network when $\mathbf{D}$ is a matrix $A \in \{0,1\}^{n \times n}$ containing edge indicators. Explicitly, let $u, v \in [n]$ be general node indices, and let $A[u,v]$ denote the u,v-th entry of A, which equals 1 if and only if u and v share an edge. Unless otherwise specified, we assume that $A[u,v]$ is symmetric and that $A[u,u] = 0$ for all $u \in [n]$. A *partition* of a node set is a finite collection $C_1, C_2, \ldots, C_K \subseteq [n]$ such that $\cup_i C_i = [n]$ and $C_i \cap C_j = \phi$ for all $i \neq j$. In other words, each node is assigned to exactly one community. Usually, a partition will be denoted with the concise vector representation $\mathbf{c} \in [K]$, with elements $c(u)$ giving the community assignment of u. We define the *degree* of node $u \in [n]$ by $d(u) := \sum_{v \in [n]} A[u,v]$, and the vector of degrees by $\mathbf{d} := \{d(1), \ldots, d(n)\}$.

The "total" degree is defined $d_T := \sum_{u \in [n]} d(u)$. Note that for undirected binary graphs, d_T is simply twice the number of edges in the graph.

1.2 Fundamental work on community detection

Community detection is a wide and varied and field of research, with plentiful sub- and sub-sub-fields. In what follows, I give a brief overviews of some classical theoretical and methodological directions in the study of communities in networks.

1.2.1 The Stochastic Block Model

Real-world interconnected systems have been studied with tools from network science and graph theory since the early twentieth century. Particularly in social science and biology, it eventually became common to view densely-connected subregions of a graph as emergent macro-structure in the system of scientific interest. Early on, Holland et al. (1983) and Anderson et al. (1992) proposed the Stochastic Block Model (SBM) as one way to model macro-structure in a networked system. In the model, nodes are organized into disjoint "blocks" with diverse intra- and inter- connection probabilities. The blocks represent meaningful real-world node subgroups which potentially interact at different rates within-block than out-block. Snijders and Nowicki (1997) gave both maximum likelihood and Bayesian approaches to estimating both the block structure and the connection probabilities for a 2-block SBM, though they soon generalized their procedure in Nowicki and Snijders (2001). The SBM approach to community detection has been of instrumental importance to the social and biological sciences (Fortunato, 2010; Porter et al., 2009).

Variants of the SBM will be of central focus in parts of this thesis. As such, an explicit definition of the model is now provided. The SBM is a generative stochastic model for an undirected, binary graph $\mathcal{G} := ([n], A)$, with three parameters. The first parameter is the number of communities (or blocks) K. The second is a community partition vector $\mathbf{c}$, where $c(u) \in [K]$ gives the community index of u. The third parameter is a $K \times K$ probability matrix $\mathbf{P}$. In the model, an edge is placed between nodes u and v (independently of all other edges) with probability

$$p_{uv} = \mathbf{P}[c(u), c(v)]. \tag{1.1}$$

Thus, the indicator $\mathbb{1}\{A[u,v]=1\}$ is Bernoulli(p_{uv}), and the resulting network $\mathcal{G}=([n],A)$ has the likelihood

$$L(\mathcal{G}|K,\mathbf{c},\mathbf{P}) := \prod_{1\leqslant u<v\leqslant n} p_{uv}^{A[u,v]}(1-p_{uv})^{1-A[u,v]} \tag{1.2}$$

The model can be seen as a generalization of an Erdős-Rényi network, inducing community structure through $\mathbf{c}$ and the matrix $\mathbf{P}$. The most common type of community structure of interest, in usual applications, is when the diagonals of $\mathbf{P}$ are larger than the off-diagonals. In the model, this will cause nodes to connect more frequently to other nodes in their community, on average, than to nodes outside their community. This is usually called "assortative" community structure. Figure 1.1 shows an example of a network exhibiting strong assortative community structure (Kim et al., 2013). The estimated SBM for this network would have $K=4$ or 5 blocks, large diagonal entires of $\mathbf{P}$, and off-diagonal entries of $\mathbf{P}$ near zero. The SBM is, of course, equally capable of generating *dis*assortive structure, in which nodes connect more frequently outside their community, a pattern which can also be of interest (Aicher et al., 2014). The SBM has since been extended to include overlapping nodes (Whang et al., 2013; Airoldi et al., 2009), directed edges (Latouche et al., 2011), and heterogeneous expected degrees (as will be covered in Section 1.3.2). Aicher et al. (2014) provided a weighted version of the stochastic block model in which edge weights are random variables from an exponential family.

An important sub-field of network analysis regards the *latent space model* (Hoff et al., 2002), which has close ties to the SBM. In the latent space model, nodes are assumed to have unobserved positions in an underlying metric space. The distance to other nodes in the metric space govern their observed interactions in the network. Some fundamental similarities between the latent space model and the stochastic block model were discussed in Rohe et al. (2011). Compared with other models and methods discussed in this work, Bayesian techniques play a much more prominent role in standard applications of latent space models (Sarkar and Moore, 2005).

1.2.2 The configuration and Chung-Lu models

Another random network model that has come to be of great importance to community detection is the configuration model. Loosely speaking, the n-node configuration model is like a random Erdős-Rényi graph, but with the restriction that node degrees be exactly a given (positive) integer

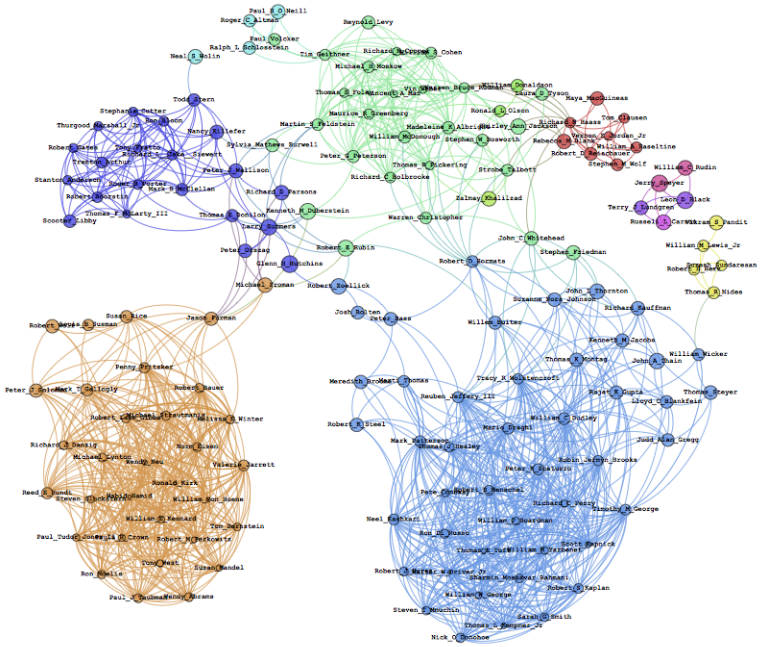

Figure 1.1: Two-hundred high-degree nodes from a network formed by data from the **LittleSis** website, which tracks powerful people and their business and political connections. Edges are placed and weighted according to output from a nonparametric bayesian model with similarities to the SBM (Kim et al., 2013).

sequence $\mathbf{d} = \{d(1), \ldots, d(n)\}$, with $d(u) \leqslant n$ for all $u \in [n]$. Initially, the configuration model was introduced in context of the general study of random graphs and their distributional properties (Bollobás, 1980; Bender, 1974). Molloy and Reed (1995) gave an algorithm for generating the configuration model, which can be written as follows:

1. Form a set L of half-edges, with $d(u)$ half-edges assigned to each node $u \in [n]$.

2. Draw two half-edges uniformly-at-random from L, without replacement; form an edge.

3. Repeat step 2, always without replacement, until all half-edges are exhausted.

Note that this process generates an undirected graph, and thus d_T must be even. This process also allows for "self-loops" ($A[u, u] = 1$) and multiple edges. However, easy modifications of the algorithm can prohibit these features.

A model that is closely related to the configuration model is that of Chung and Lu (2002). Given a degree sequence $\mathbf{d}$, the model is generated by placing an edge between nodes u and v,

independently of all other edges, with probability

$$p_{uv} = \min\left\{\frac{d(u)d(v)}{d_T},\ 1\right\}, \tag{1.3}$$

where d_T is the total degree (see Section 1.1). This "Chung-Lu" or "Expected-Degree Random Graph" model, as it is often called, is more directly related to the Erdős-Rényi model than the configuration model, since it retains edge-independence. In fact, whereas with the configuration model the sequence $\mathbf{d}$ is best seen as a restriction on the Erdős-Rényi, in the Chung-Lu model it is a generalization. When all degrees are equal to some fixed d, the Chung-Lu model reduces to Erdős-Rényi with probability d/n. Nonetheless, fundamental and important similarities remain between the Chung-Lu and configuration models. It is easy to derive that, if $\max_{uv} d(u)d(v) \leqslant d_T$, the expected degrees under the Chung-Lu model are precisely $\mathbf{d}$. Conversely, when $d_T = o(n)$, the probability of an edge between u and v under the configuration model is approximately p_{uv}.

Though neither the configuration nor Chung-Lu models were originally introduced in the context of community detection, they eventually played important (and related) roles in the study of communities. Newman et al. (2002) published an early, often-cited paper proposing the utility of the configuration model for simulating social networks. They noted the inadequacy of Erdős-Rényi networks for this task, as the degree-distribution of real-world networks are always skewed. Notably, Newman writes that the configuration model produces "a graph with exactly the desired degree distribution, but which is in all other respects random". This may be the first acknowledgment (albeit implicit) that the configuration model may be a suitable null model to test for alternative structure in graphs with heterogenous degrees. Indeed, soon after that publication, community detection methods based on the 1st-order structure in the configuration and Chung-Lu model were introduced, as discussed in the next section. These new methods fundamentally changed the field, and the related roles of the configuration and Chung-Lu models in these research directions are now well-known (Olhede and Wolfe, 2012; Durak et al., 2013).

Figure 1.2 demonstrates the efficacy of the configuration model as a null for community detection. On the left, we see an un-labeled, binary network with what appears to be some community structure. On the right, the existing edges have been randomly re-assigned with the Malloy-Reed algorithm, erasing any evidence of community structure. In this example, applying the configura-

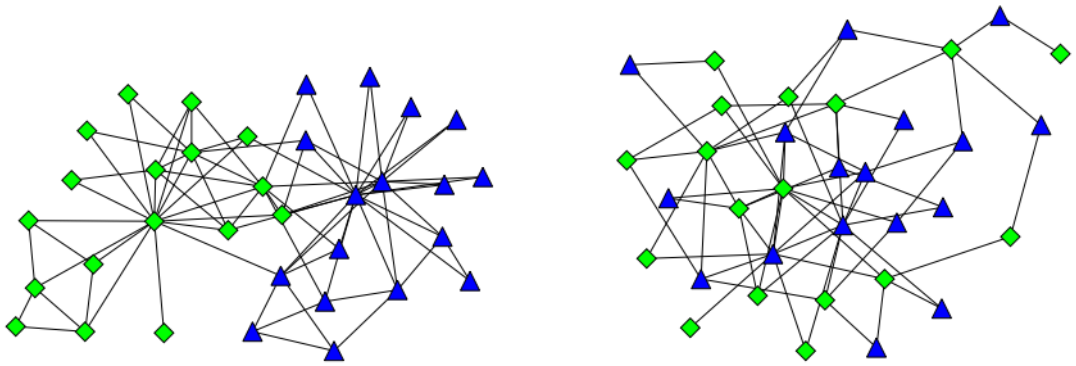

Figure 1.2: The configuration model applied to an empirical random network. Source: Aaron Clauset's CSCI5352 lecture notes, Sante Fe Institute.

tion model algorithm to an empirical network can be thought of as a permutation of the edges, conditional on the observed degrees. Significance of the community structure in the observed network can therefore be assessed either by successive applications of the model algorithm, in the style of a permutation test, or by analysis of test statistics from the network with respect to the distribution of the model.

1.2.3 Modularity and optimization

A novel approach to community detection was introduced via the *modularity* metric by Newman and Girvan (2004). Modularity is a score of a network partition such that, loosely speaking, when it is large, the community structure given by the partition is strong. The modularity score compares the empirical edge densities to the expected edge densities under the configuration model (see Section 1.2.2). Explicitly, the modularity score sums the difference between edge indicators of node pairs in the same community from their corresponding Chung-Lu probability p_{uv} (see Equation 1.3). Given a network partition $\mathbf{c}$, modularity is defined

$$Q(\mathbf{c}) := \frac{1}{d_T} \sum_{u,v \in [n]} \left(A[u,v] - \frac{d(u)d(v)}{d_T} \right) \mathbb{1}(c(u) = c(v)), \tag{1.4}$$

where $\mathbb{1}$ is the indicator function. Note that $Q(\mathbf{c})$ is always in the interval $[-1, 1]$. From the above equation, it is obvious that when the edge densities within the communities outlined by the partition $\mathbf{c}$ are much larger than what is expected under the configuration model, $Q(\mathbf{c})$ will be closer to 1.

Finding the global maximizer of modularity is NP-complete (Brandes et al., 2006). Thus, all commonly-used community detection methods based on modularity employ approximation algorithms. Girvan and Newman gave the first approach to approximately maximizing modularity, which involves first an edge-removal algorithm for finding proposed partitions (Girvan and Newman, 2002), then a method to choose a partition using the modularity score (Newman and Girvan, 2004). Modularity-based methods have been generalized, re-worked, and analyzed in many ways over the years (Newman, 2004b, 2006b; Clauset et al., 2004; Eriksen et al., 2003; Langone et al., 2011), and now are among the most popular tools in network science. The modularity score has also been extended to networks with directed edges and overlapping communities (Nicosia et al., 2009; Chen et al., 2014), and to networks with bipartite community structure (Barber, 2007), as will be discussed more fully in Section 1.3.4. Overall, the modularity approach represents a marked departure from the model-based community detection procedures described in Section 1.2.1.

Importantly, the modularity framework is not the only approach to assessing or finding community partitions in networks. Pre-dating modularity, the *conductance* measure of a node set is the ratio of cross-edges between that set and its complement to the number of internal edges of the set. A thorough study of conductance and how it has been used to detect communities, including novel approaches, is given by Leskovec et al. (2008). Other partition-based community detection methods have roots in information theory or random walks. These methods involve many diverse approaches to community detection and types of algorithms. Three examples are: (i) Infomap (Rosvall and Bergstrom, 2008), an algorithm incorporating information-theoretic measures of community structure; (ii) Walktrap (Pons and Latapy, 2005), an algorithm incorporating random walk probabilities; and (iii) a label-propagation algorithm presented in (Raghavan et al., 2007). These methods perform quickly even for large networks and have been implemented in standard software packages from `R` and `python`, making them commonly used tools in applied network science.

1.2.4 Spectral clustering

Another classical approach to community detection involves projecting the adjacency matrix or Laplacian of the graph onto principal component vectors, and then clustering those vectors with standard methods like k-means or the EM algorithm. Detailed layouts of the various, related spectral clustering methods can be found in von Luxburg (2007). The roots of spectral community

detection are in the graph partitioning work of the mid-to-late twentieth century. One early and particularly influential paper in this field called "Algebraic connectivity of graphs" was written by Miroslav Fiedler (1973). In that paper, Fiedler relates the eigenvalues of an unweighted, undirected network's adjacency matrix to its connectedness. Much subsequent, related work analyzed further relationships between graph spectra and optimal cuts of the graph using linear algebra and theory of random walks (Ding et al., 2001; Pothen et al., 1990). While these results are most directly applied to problems in (for example) distributed computing and circuit designs (Chamberlain et al., 1998; Shirinivas et al., 2010), they laid the groundwork for the spectral approach to clustering and community detection (Boccaletti et al., 2006). Some examples of the application and analysis of spectral approaches to community detection can be found in (for example) Newman (2006a,b); Richardson et al. (2009); Rohe et al. (2011); Newman (2013).

1.3 Recent directions in community detection

In recent years, the field of community detection has expanded to include novel algorithmic approaches, nuanced theoretical analyses, and additional methods for new types of complex networks. Current directions that pertain to the work in this thesis are now discussed.

1.3.1 Community Extraction

The classical approaches to community detection described in Section 1.2 are all based on the evaluation of a community partition. An alternative approach is to conduct set-wise searches, in which communities are identified and evaluated one-by-one. This approach, often called community "extraction", can feature a number of attractive benefits. One particular benefit is that not all nodes need to be assigned a community. Usually, this happens for nodes that do not have strong connectivity to any other subgroup. In this work, I such nodes are called "background". Extraction methods have been put forth in a number of recent publications:

1. For a community $C \subseteq [n]$, Zhao et al. (2011) defined the following measure of connectivity:

$$Q(\mathbf{c}) := |C|^{-1} \left(|C|^{-1} \sum_{u,v \in C} A[u,v] - |C^c|^{-1} \sum_{u \in C, v \in C^c} A[u,v] \right).$$

The above measure compares the average in-degree of C to the average out-degree. Though this measure is fundamentally heuristic and model-free, it captures a reasonable conception of empirical community strength. Zhao et al. (2011) give a node-swapping algorithm to locally maximize this metric over all possible node sets. Though their algorithm can identify background, one of its drawbacks is that overlapping communities are dis-allowed.

2. Lancichinetti et al. (2011) introduced the Order Statistics Local Optimization Method (OSLOM), which locally optimizes a "fitness" function characterizing the statistical significance of a community with respect to the configuration model. Their algorithm is capable of handling networks with directed and weighted edges, can detect hierarchical community structure, and is capable of identifying overlapping communities and background nodes.

3. Wilson et al. (2014) introduced Extraction of Statistically Significant Communities (ESSC), which iteratively refines communities using statistical hypothesis testing. Using the configuration model as a null, ESSC computes tail probabilities of observed edge counts from a target community to all other nodes. ESSC then refines this community by keeping nodes with significant tail probabilities, and discarding others. When applied iteratively, this algorithm adaptively chooses the number of communities, can find overlapping communities without restriction, and naturally identifies background nodes.

The community detection methodologies introduced and analyzed in this thesis are all extraction algorithms. In Section 1.4, brief summaries of these new methods are given, in particular how they relate to or differ from those mentioned above.

1.3.2 The Degree-Corrected Stochastic Block Model

One major drawback of the standard Stochastic Block Model (see Section 1.2.1) is that it does not allow for degree heterogeneity. Indeed, it is easy to derive that, under the standard SBM, the expected degrees of nodes in the same community are identical. As noted in Newman et al. (2002), degrees from real-world networks are often extremely heterogeneous. To account for this, Coja-Oghlan and Lanka (2009) put forth what is now known as the "Degree-Corrected" Stochastic Block Model (DCSBM). The model is a simple but powerful extension of the standard SBM. The DCSBM retains the K, $\mathbf{c}$, and $\mathbf{P}$ parameters of the standard SBM, but also includes an n-vector ϕ.

Each component $\phi(u)$ is a positive weight controlling the node u's propensity to form edges with other nodes. The model is then generated in the Bernoulli-style of the standard SBM, but with probabilities adjusted by the ϕ vector. Defining $\phi_T := \sum_{v \in [n]} \phi(v)$, the appearance of each edge has probability

$$p_{uv} := \frac{\phi(u)\phi(v)}{\phi_T} \mathbf{P}[c(u), c(v)] \tag{1.5}$$

Note that ϕ and $\mathbf{P}$ must be chosen so that $\max_{uv} p_{uv} \leqslant 1$,

The extension of the DCSBM in the manner given by Equation 1.5 is analogous to the Chung-Lu generalization of an Erdős-Rényi network. Moreover, the DCSBM can actually be formulated as a generalization of Chung-Lu that induces community structure with $\mathbf{P}$. One major difference from the Chung-Lu model is that the degree parameter sequence ϕ is not equal to the expected degree sequence, due to perturbation by the probabilities $\mathbf{P}$. However, as explained in Coja-Oghlan and Lanka (2009), given $\mathbf{P}$ and a target degree sequence $\mathbf{d}$, it is possible to set ϕ so that the expected degrees equal $\mathbf{d}$.

Karrer and Newman (2011) give an excellent example of the usefulness of the DCSBM. In Figure 1.3, we see two community partitions of the same network: one given by fitting the SBM, the other by fitting the DCSBM. The images in Figure 1.3 appear in Newman (2012) as reproductions of those from Karrer and Newman (2011). The network data comes from internet scraping of links between political blogs (Adamic and Glance, 2005), and is equipped with an acknowledged ground truth which labels each blog as politically "liberal" or "conservative". The visualization was constructed without knowledge of this ground truth, using only the links between nodes. Nodes are colored by their labeling under the choice of model, and are sized based on their degree. We see that using the Degree-Corrected SBM, the estimated labels correspond to the visualization layout, which means the labeling follows patterns of stronger or weaker edge densities between node subsets. In contrast, using the standard SBM, the estimated labels follow the degree distribution: nodes are split into two groups of larger and smaller degrees. Furthermore, as reported in Karrer and Newman (2011), the estimated labels from the SBM have almost no association with the acknowledged ground truth given in Adamic and Glance (2005), whereas the DCSBM labels closely align with the political division.

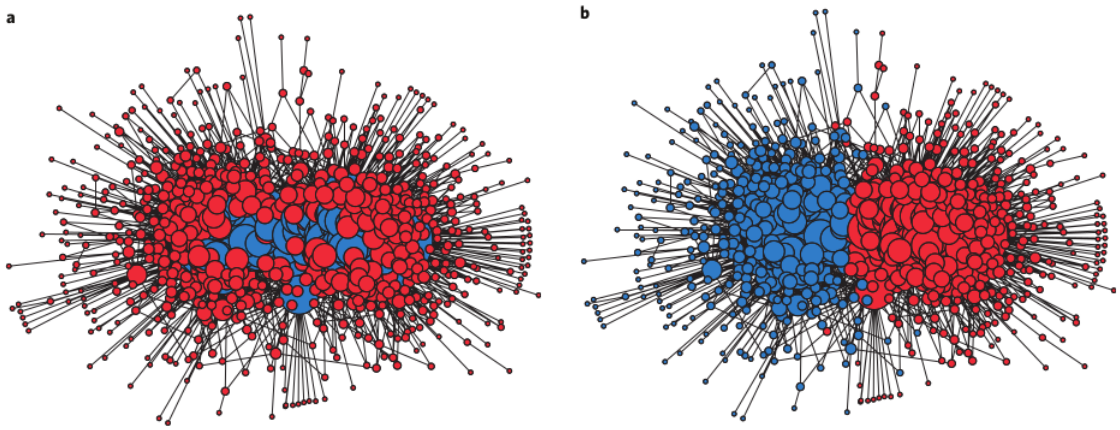

Figure 1.3: Visualization of a political blog network, labels estimated by the standard SBM (a) and the DCSBM (b) (Newman, 2012).

1.3.3 Consistency of community detection methods

An important, general issue about community detection is the consideration of any given method's ability to recover "true" communities in the network. In the community detection literature, this issue is framed as *consistency* under an appropriate generative model. The generative model is almost always taken to be the SBM or some variant thereof. The definition of consistency varies depending on the method under study and the constraints of the theoretical analysis, but it is most often a high probability statement about the clustering error as the number of nodes n tends to infinity. This sub-section contains a thorough review of some prominent consistency analyses of community detection methods.

Early work on consistency of community detection largely focused on simple versions of the SBM, restricting the number of blocks to 2 (Snijders and Nowicki, 1997), or forcing community sizes and connection probabilities to be equivalent (Condon and Karp, 2001). These analyses will not be discussed in detail, since modern-day consistency analyses have largely eliminated these assumptions. For instance, Bickel and Chen (2009) proved the consistency of the modularity and SBM likelihood partition measures under the SBM with an arbitrary number of blocks. Their analysis begins in consideration of the maximizer $\hat{\mathbf{c}}_n$ of a general partition measure (like modularity). The maximizer $\hat{\mathbf{c}}_n$ is assumed to be estimated from a n-node SBM with a fixed number of blocks K that does not depend on n. Denote the probability measure associated with the SBM by $\mathbb{P}_n$.

Bickel and Chen (2009) defined consistency of a partition measure by

$$\mathbb{P}_n(\hat{\mathbf{c}} = \mathbf{c}) \to 1 \text{ as } n \to \infty, \tag{1.6}$$

where the partition vector equivalence is defined up to permutation of the community labels. They show that a general class of partition measures which includes modularity and the SBM likelihood are consistent in the above sense. Their result depends on a few key assumptions:

- The average degree of the SBM grows faster than $\log n$.
- The true model partition maximizes the limiting partition measure.
- The relative community sizes are positive and do not change with n.
- The matrix $\mathbf{P}$ has entries of the same asymptotic order and unique columns.

Along with a few other technical conditions on the partition measure, these assumptions are a significant improvement over those found in preceding consistency analyses of community detection and graph partitioning algorithms. Zhao et al. (2012) generalized the analysis of Bickel and Chen in the following important ways:

1. Instead of taking the standard SBM as the generative model, they assumed the Degree-Corrected SBM (see Section 1.3.2).
2. They defined a notion of in-probability ("weak") consistency, as follows: *The maximizer* $\hat{\mathbf{c}}_n$ *is weakly consistent if for any* $\epsilon > 0$*,*

$$\mathbb{P}_n \left(n^{-1} \sum_{u \in [n]} \mathbb{1}\{\hat{\mathbf{c}}_n(u) \neq \mathbf{c}_n(u)\} < \epsilon \right) \to 1 \text{ as } n \to \infty. \tag{1.7}$$

   Above, the partition equivalence is again defined up to a permutation of labels. They showed that, under assumptions similar to those from Bickel and Chen (2009), maximzers $\hat{c}_n$ are weakly consistent if the average degree tends to infinity. They also proved the strong consistency analog to Bickel and Chen (2009), in the sense of (1.6), when the average degree must grow more quickly than $\log n$.

The work of Bickel and Chen (2009) and Zhao et al. (2012), described above, have been crucial to understanding the properties and performance of partition measures on networks with communities. However, it important to note that their results pertain only to the *global* maximizer of any given partition measure, which in almost all practical situations is computationally prohibitive to produce. This is a non-trivial problem, since local optimizers of partition measures often have diverse community structures (Peel et al., 2016). The consistency of local optimizers has received comparatively little attention, since a local optimizer is algorithm-dependent. Consistency results for common variational algorithms used to fit the SBM have been given by Choi et al. (2012) and Celisse et al. (2012), among others. However, these results generally require stronger assumptions on the sparsity of the network.

Another important class of consistency analyses for community detection focuses on spectral methods. Lei et al. (2015) prove that, under assumptions as weak as, if not weaker than, those in the aforementioned optimization-based analyses, the asymptotic clustering error obtained *in practice* by spectral methods is bounded by a function of the average degree, number of communities, ratio of the maximum and minimum community sizes, and the ratio of the maximum and minimum connection probabilities. This is a general result that contains some of the weakest assumptions available for spectral community detection methods. However, spectral methods are in general more computationally prohibitive than modularity maximization, especially for very large networks. Though the analysis of spectral methods for community detection is a vast field with many open problems, the field will not be discussed in any more depth here, since the consistency analyses in this thesis are more closely related to those for modularity and the SBM likelihood.

1.3.4 Community detection for multilayer and bipartite networks

One important type of network is known as "multilayer" or "multiplex", and features a finite number of distinct edge sets corresponding to the same set of nodes. Notationally, a multilayer network can be written $\mathcal{G} = ([n], [m], A)$, where $[n]$ is a node index set, $[m]$ is a layer index set, and A is an $n \times n \times m$ binary array, with the third dimension indexing the adjacency matrices corresponding to particular layers. The edge sets in these layers can be formed from, for example, elements of a time series, or experimental units from a clinical trial. Multilayer network models have also been applied to modeling and analysis of air transportation routes (Cardillo et al., 2012),

studying individuals with multiple sociometric relations (Fienberg et al., 1980, 1985), and analyzing relationships between social interactions and economic exchange (Ferriani et al., 2013). Kivelä et al. (2014) and Boccaletti et al. (2014) provide two recent reviews of the study of multilayer networks.

The development of community detection methods for multilayer networks is still relatively new. One common approach to multilayer community detection is to project the multilayer network in some fashion onto a single-layer network and then identify communities in the single layer network (Berlingerio et al., 2011; Rocklin and Pinar, 2013). A second common approach to multilayer community detection is to apply a standard detection method to each layer of the observed network separately (Barigozzi et al., 2011; Berlingerio et al., 2013). However, the first approach fails to account for layer-specific community structure and may give an oversimplified or incomplete summary of the community structure of the multilayer network; the second approach does not enable one to leverage or identify common structure between layers. Methods introduced in this thesis will avoid some of these limitations.

In addition to the methods above, there have also been several generalizations of single-layer methods to multilayer networks. For example, Holland et al. (1983) and Paul and Chen (2015) introduce multilayer generalizations of the stochastic block model from Wang and Wong (1987) and Snijders and Nowicki (1997). Peixoto (2015) considers a multilayer generalization of the stochastic block model for weighted networks that models hierarchical community structure as well as the degree distribution of an observed network. Paul and Chen (2016) describe a class of null models for multilayer community detection based on the configuration and expected degree model. Stanley et al. (2016) considered the clustering of layers of multilayer networks based on recurring community structure throughout the network. Mucha et al. (2010) first extended the notion of modularity to multilayer networks, and De Domenico et al. (2015) generalized the map equation, which measures the description length of a random walk on a partition of vertices, to multilayer networks. De Domenico et al. (2013) discuss a generalization of the multilayer method from Mucha et al. (2010) using tensor decompositions.

1.3.4.1 Bi-partite networks

Another type of network is known as a "bipartite". Bipartite networks have two defining properties. First, the node set $[n]$ is bisected into two non-overlapping node sets N_1 and N_2 of sizes

$n_1 := |N_1|$ and $n_2 := |N_2|$. Second, edges pair nodes only when one node is from N_1 and the other is from N_2. In other words, if two nodes are from the same side of the network, no edge can exist to pair them.

Note that the division of the node set in bipartite networks is of a different character than the division of an *edge* set into multiple layers, as in the multi-layer setting discussed in the preceding part of this section. Whereas a multi-layer network has multiple edge sets corresponding to a unified node set, a bipartite network has multiple (two) *node* sets with a shared edge set.

Formally, a bipartite network can be written as $\mathcal{G} = (N_1, N_2, A)$, where N_1 and N_2 are disjoint node sets, and the adjacency matrix A is an $|N_1| \times |N_2|$ matrix containing edge indicators between for edges between nodes in N_1 and nodes in N_2. Communities in bipartite networks are in fact *bi*-communities. A bi-community $(C_1, C_2) \in 2^{N_1} \times 2^{N_2}$ consists of a node set from each side of the network. In applications, it is of interest to find bi-communities (C_1, C_2) such that nodes in C_1 are strongly connected to nodes in C_2, but weakly connected to other nodes in N_2, and vice-versa.

Some community detection methods for bipartite networks have been published and well-cited (e.g. Barber (2007); Du et al. (2008); Liu and Murata (2010)). More recently, Bartlett (2015) extended spectral community detection to bipartite networks. One growing area of application of bipartite community detection is when the node sets are genomic markers, and edges indicate a certain level of interaction strength or statistical correlation (e.g. Platig et al. (2015)).

1.4 Contributions of this thesis

The specific contributions in this thesis follow two broad directions, listed below. Note that notations not defined in Section 1.1 are used loosely, will be defined explicitly in the corresponding chapters.

1. **Adapt testing-based extraction to general networks data.** The testing-based extraction algorithm ESSC, mentioned in Section 1.3.1, was limited to binary, single-layer, undirected networks. In my work, I formalize the testing-based extraction approach in a way generalizable to almost all types of network data. Suppose we are given a network $\mathcal{G} = ([n], \mathbf{D})$ with $\mathbf{D} \in \mathcal{D}$. Two components are needed to perform testing-based community extraction:

    - A test statistic $T : [n] \times 2^{[n]} \times \mathcal{D} \mapsto \mathbb{R}$ of node-to-set association.

- A null model $\mathbb{P}_\theta$ on $\mathcal{D}$ which specifies a notion of "lack of association" for arbitrary nodes and sets u and B.

Chapter 2 is devoted to a rigorous treatment of these concepts. In practice, the testing framework components T and $\mathbb{P}_\theta$ are not always easy to formulate, since $\mathbf{D}$ can consist of data of many different types. Multiple projects in this thesis involve an adaptation of testing-based extraction to new types of network data.

2. **Assess the statistical consistency properties of extraction methods.** The consistency properties of extraction methods have not yet well-investigated. In this thesis, new conceptions of consistency for extraction algorithms are introduced, and consistency properties of proposed methods under various models with planted communities are established.

The following sections introduce specific contributions of this thesis with respect to the directions above.

1.4.1 Community extraction for edge-weighted networks

Recall from Section 1.2 that many classical community detection methods are based, in some way, on a null model. A significant drawback of these methods is that no explicit null model exists for edge-weighted networks. Edge weights are commonplace in network data, and can provide information that improves community detection power and specificity (Newman, 2004a; Boccaletti et al., 2006). While many existing community detection methods have been established for weighted and un-weighted networks alike, due to the absence of an appropriate weighted-network null model, very few of these methods provide statistical significance assessments of weighted-network communities. In contrast, some significance-based community detection methods have recently been introduced in the literature for *un*-weighted networks, due to the popularity and acceptance of the configuration model as a binary-network null. These methods, OSLOM from Lancichinetti et al. (2011) and ESSC from Wilson et al. (2014), were discussed in some detail in Section 1.3.1. While OSLOM can in practice handle edge weights, the method uses an exponential function to calculate nominal tail probabilities for edge weight sums, a testing approach which is not based on an explicit null. As a consequence, communities in weighted networks identified by this approach may in some cases be spurious or unreliable, especially when no "true" communities exist.

The project presented in Chapter 3 has a three-fold purpose: (i) to provide an explicit null model for networks with weighted edges, (ii) to present a community extraction method based on hypothesis tests with the null, and (iii) to provide an analysis of the consistency properties of the method with respect to a weighted stochastic block model. These contributions provide steps toward a rigorous statistical framework with which to study communities in weighted networks. Results of extensive simulations show that the proposed extraction method is more successful than competitors at identifying overlapping and background nodes. The method also extracts communities that align with key features of real data.

1.4.2 Community extraction for multi-layer, binary networks

As discussed in Section 1.3.4, some approaches to multi-layer networks proceed by either aggregating layers into a single-layer weighted or multi-edge network, or by assuming that the same community structure exists across all layers. In general, these practices ignore potential layer-wise heterogeneity. As just one example from social networks, a group of individuals may be well-connected via friendships on Facebook; however, this common group of actors will likely, for example, not work at the same company. In realistic situations such as these, a given vertex community may only be present in a proper subset of the layers. It may be of practical interest to determine the *persistence* properties of such a community across various layer sets, something which layer-aggregation or homogeneous multi-layer modeling cannot accomplish. In general, complex and differential relationships between actors will be reflected in heterogeneous behavior of different layers, behavior which existing multi-layer network community detection methods, for the most part, cannot capture.

Chapter 4 introduces a multilayer community detection method called Multilayer Extraction, which adaptively handles multilayer networks with heterogeneous layers. The primary purpose of the chapter is to introduce a multilayer notion of significance for individual communities via a score function, and to prove a consistency theorem for the score. To show how the score may be used, an algorithm incorporating the score is presented and discussed, though the algorithm is mainly the contribution of other authors on a recently-submitted associated paper.

1.4.3 eQTL analyses and bi-partite correlation networks

In Chapter 5, new methods to study expression Quantitative-Trait Loci (eQTL) networks are presented. An expression Quantitative Trait Locus (eQTL) is a genetic polymorphism (typically a single-nucleotide polymorphism, abbreviated by SNP) that is associated with transcriptional expression levels in a particular tissue. The statistical analysis of eQTLs has become increasingly important in understanding molecular mechanisms by which genetic variation gives rise to complex traits and human disease (cf. Morley et al. (2004); Gilad et al. (2008); Grundberg et al. (2012); Westra et al. (2013)). For example, eQTL studies can be used to plausibly link disease phenotypes analyzed in Genome-Wide Association Studies (GWAS) to gene expression, with recent work focusing on variation across tissues (Gamazon et al., 2015; Ardlie et al., 2015). Although the underlying biology is complex, a fundamental step in many analyses is to compare the genotypes of a large number of SNPs to the expression levels of all known genes, which presents challenges in computation and multiple testing (Wright et al., 2012). Effectively, the quantitative results of these comparisons are edge weights on a complete bi-partite network consisting of SNP loci (henceforth just called "SNPs"), on one side, and genes, on the other.

The first section of Chapter 5 presents a new model for investigating the association strength of a *single* gene-SNP pair. Statistical analyses of eQTLs have often been based on standard linear regression (Shabalin, 2012), with a focus on testing and detection of gene-SNP pair relationships. A key step, commonly considered necessary to avoid false positives, has been to normalize and transform the expression data prior to analysis (Beasley et al., 2009). However, normalization removes the scale of the expression data, and with it, a natural measure of effect size due to genotype. As a consequence, eQTL effect size has often been described in terms of regression partial R^2 between genotype and transformed expression (see for example Stranger et al. (2007)). However, the R^2 statistic can be highly sensitive to transformations of the response, and is difficult to interpret biologically. An appropriate eQTL model should reflect a coherent model of allelic contributions to expression, and be able to capture and describe evidence of dominance that is still rarely examined in detail (Powell et al., 2013). Furthermore, a biologically appropriate effect-size model improves the accuracy of hypothesis tests (as I will show), and provides reliable rankings of eQTLs in terms of effect sizes as opposed to p-values.

In Section 5.1, I propose ACME-eQTL, a new model for the effect size of cis-acting eQTLs, in which the effects of allele count on expression are Additive Contributions on the original expression scale, with Multiplicative Error. In the ACME-eQTL model the log of expression is equal to the *log of a linear systematic term* ("log-of-linear") plus noise and covariate effects: this subtle difference from standard log-linear modeling is of key importance in estimating and interpreting effect sizes. Although the ACME-eQTL model is straightforward, it reflects a marked departure from standard practice in eQTL analyses and has important implications for inferences from effect sizes. The major contributions of this project are efforts to: (i) motivate and introduce the model, (ii) provide a novel fitting algorithm and corresponding software package, and (iii) derive a means to calculate and evaluate appropriate p-values. To support the use of the model, goodness-of-fit tests are performed on real data from the GTEx Project (Lonsdale et al., 2013). Also considered are the model's robustness to skew in the residuals (under the null), and its superior power and estimation accuracy compared with existing models (under the alternative).

In Section 5.2, the second section of Chapter 5, I introduce preliminary work on a community detection method for eQTL networks. Rather than analyzing individual gene-SNP pairs, this method assesses groups of eQTLs in the search for mutually cross-correlated sets of genes and SNPs. This methodological research direction is relatively new, and has been applied to eQTL networks in a few recent publications, for example in Huang et al. (2009); Bao et al. (2010); Platig et al. (2015). The trend in these publications (among others) is to use the following general procedure:

1. Compute all cross-correlations between genes and SNPs.

2. Dichotomize the cross-correlations through some inferential procedure.

3. Perform a binary community detection routine on the resulting bi-partite network.

This approach has two major drawbacks. The first is the computational burden. In modern-day eQTL studies, the genes of interest number in the tens of thousands, and the SNPs of interest number in the hundreds of thousands or millions. All three steps of the above approach are computationally intensive for this scale of data. The second drawback is the information loss incurred by step 2. The statistical significances of eQTLs, as determined by correlation values,

vary widely, so any threshold will be necessarily insufficient to capture the complex relationships in the network. Furthermore, some community detection methods for bi-partite networks are based on null models (e.g. Barber (2007)) that carry assumptions unreasonable for binary networks formed by discretized correlations. In particular, a dichotomized correlation network contains edge-to-edge correlations determined by the underlying partial-correlation structure of the expression data, correlations which are not accounted for in standard models for binary graphs.

The community detection method for bi-partite correlation networks that I introduce, called "Correlation Bi-Community Extraction" (CBCE), overcomes both of the aforementioned issues. CBCE is formulated through a bi-partite network adaptation of the testing-based extraction framework. In this setting, which is described in full in Section 5.2, the goal is to recover eQTL bi-communities, consisting of a gene-subset, SNP-subset pair with large observed cross-correlations. Being an extraction method, CBCE searches for modules one-by-one, which eliminates the need to compute all cross-correlations or a full binary bi-partite network. This greatly reduces the computational burden of the aforementioned approach. Furthermore, the hypothesis testing inherent to CBCE is able to deal with the observed correlations directly, employing a specific null model for correlation sums. In other words, CBCE directly uses all information available in the raw correlation data, and is based upon a principled null model that reflects the correct distribution of the variables at play. In Section 5.2, I detail the CBCE algorithm in full, provide theoretical support for a p-value approximation, and show preliminary results comparing its performance to some existing methods on simulated data.

1.5 Document Organization

The rest of the document is organized as follows. In Chapter 2, I present the testing-based extraction framework in generality. In Chapters 3-5 that follow, I present the full scope of my work on the projects introduced in Sections 1.4.1-1.4.3 (respectively).

CHAPTER 2

Node-Set Testing for Complex Networks

Though community detection methods vary widely, the notion of a "community" usually involves the characteristic that constituent nodes are strongly *internally* associated, but weakly *externally* associated. As discussed in Section 1.2, many classical approaches to detect this type of association in networks rely on a univariate score of a node partition. Dependence on a partition score can be limiting, since such a score must assess the communities defined by the partition simultaneously, with one number. Furthermore, the vast majority of partition-based methods do not come with a natural conception of statistical significance.

In this chapter, a methodological framework called Node-Set Testing (NST) is introduced for community detection on networks with potentially multiple, heterogenous data types. The NST framework involves explicit notions of statistical significance, and is motivated by recent iterative testing-based methods introduced by Lancichinetti et al. (2011) and Wilson et al. (2014). These methods, however, involve significance tests for binary networks only, and do not rigorously establish the theoretical underpinnings of the approach. NST introduces a general yet rigorous definition of a community that does not depend on any partition score. A generalized community extraction algorithm for networks is then built around this definition. A theoretical result is established regarding the error properties of the proposed extraction method.

2.1 Node-Set Testing Framework

Suppose we have a network with n nodes and corresponding data sets $\mathbf{D} := (D_1, D_2, \ldots, D_m)$. A single-layer binary network, for instance, has just one data set $D_1 = A$, the adjacency matrix. A weighted network, however, may have two data sets: $D_1 = A$, the adjacency matrix, and $D_2 = W$, the weight matrix. Note that A and W are distinct and worthy of separate consideration in cases

when weights can be zero despite the existence of an edge. In general, a full network data object can be denoted by the double $\mathcal{G} := ([n], \mathbf{D})$.

Define an association function $a : [n] \times 2^{[n]} \mapsto \mathbb{R}$ as some measure of the *true* relationship between nodes and sets in $\mathcal{G}$. For any $u \in [n]$ and $B \subseteq [n]$, $a(u, B) > 0$ indicates positive association between u and B, $a(u, B) < 0$ indicates negative association, and $a(u, B) = 0$ indicates no association. Using $a(u, B)$, the notion of a *community* can be formally defined, as follows:

Definition 1. *Given $a : [n] \times 2^{[n]} \mapsto \mathbb{R}$, a community is any node set $C \subseteq [n]$ satisfying*

(i) $a(u, C)$ is positive for each $u \in C$, and

(ii) $a(u, C)$ is non-positive for every $u \in C^c$.

The function a is a stand-out feature of the NST approach to community detection. In most community detection methods, the meaning of "association" is tied to a fixed objective function (like modularity or within-sum-of-squares). In contrast, in the NST framework, we have complete freedom in our notion of association, and thus also in our notion of a community. Most often, a will have an interpretation as a "ground-truth" or "population" value of some *empirical* measure of association. Thus, strictly speaking, $a(u, B)$ will depend on some (unknown) distribution $\mathbb{P}$ that is assumed to have generated the data $\mathbf{D}$. Here are examples of association functions from two different types of networks:

1. Let $\mathcal{G} = ([n], A)$ be an undirected binary network with distribution $\mathbb{P}$. Define the expected edge count between u and B under $\mathbb{P}$ by

$$\bar{d}(u, B) := \mathbb{E}\left(\sum_{v \in B} A[u, v]\right). \tag{2.1}$$

   As discussed in Section 1.2.2, the appropriate null for binary networks is the Chung-Lu (or configuration) model, in which edge probabilities are proportional to expected degrees. Therefore, for us to call u and B truly associated, $\bar{d}(u, B)$ should be *greater* than the sum of the corresponding null edge probabilities. Explicitly, let $\bar{d}(u)$ be the expected degree of u under $\mathbb{P}$, and $\bar{d}_T$ the sum of expected degrees (as in Section 1.1). Recall that, if $\mathbb{P}$ is the Chung-Lu model, we have $\mathbb{P}(A[u, v] = 1) = \bar{d}(u)\bar{d}(v)/\bar{d}_T$. Thus, an appropriate association

function is

$$a(u, B) := \bar{d}(u, B) - \sum_{v \in B} \frac{\bar{d}(u)\bar{d}(v)}{\bar{d}_T} := \bar{d}(u, B) - \bar{d}(u)\bar{q}(B), \tag{2.2}$$

with $\bar{q}(B) := \bar{d}_T^{-1} \sum_{v \in B} \bar{d}(v)$, the expected relative edge-density of B under the null. The value $\bar{q}(B)$ is essentially the null probability that a randomly chosen edge connects to a node in B. Thus, an interpretation of (2.2) is that u and B are associated if and only if $\bar{d}(u, B)$ exceeds the null-expected proportion of $d(u)$ incident with B.

2. Let $\mathbf{X} \in \mathbb{R}^{d \times n}$ be a matrix of Euclidean data. Define a correlation network by $\mathcal{G} = ([n], \mathbf{X})$, where edges are identified with the sample Pearson correlations of the d-dimensional $\mathbf{X}$ data. Assume $\mathbf{X}$ has population $n \times n$ correlation matrix Σ, with general element $\rho(u, v)$. Then the natural association function for detecting communities with high average internal correlation is

$$a(u, B) := \sum_{v \in B} \rho(u, v). \tag{2.3}$$

Unlike in the previous example, the sum of the edge population values $\rho(u, v)$ is the association measure of interest, and is equal to zero under the null, without centering. This contrast shows the flexibility of the NST approach across different types of networks.

In the next section, a statistical approach to significance tests for values of $a(u, B)$ is laid out, yielding a general NST community extraction algorithm.

2.2 Node-set association testing

Let $\mathcal{D}$ represent the product space of the data $\mathbf{D} = (D_1, \ldots, D_m)$. To discover communities satisfying Definition 1, we assume the existence of a summary statistic $T : [n] \times 2^{[n]} \times \mathcal{D} \mapsto \mathbb{R}$ with the characteristic that large enough values of $T(u, B, \mathbf{D})$ suggest positive values of $a(u, B)$. We also assume the existence of an explicit null model $\mathbb{P}_\theta$ which gives the distribution of $\mathbf{D} \in \mathcal{D}$ when $a(u, B) = 0$. Here, θ is a parameter (or set of parameters) for the null model. Assuming θ is known, we can test the hypotheses

$$H_0 : a(u, B) = 0 \quad \text{vs.} \quad H_A : a(u, B) > 0 \tag{2.4}$$

with the one-sided p-value

$$p(u, B, \mathbf{D}) := \mathbb{P}_\theta\big(T(u, B, \widetilde{\mathbf{D}}) > T(u, B, \mathbf{D})\big), \tag{2.5}$$

where $\widetilde{\mathbf{D}}$ is a random realization of the network data under the null. However, in practice θ is rarely known. Usually, θ must be estimated or set to plug-in values computed from the data. This is analogous to the treatment of the unknown parameter p in a standard z-test of a binomial random variable. The canonical confidence interval for p depends on p itself. Thus, the maximum likelihood estimate $\hat{p}$ is used in the formula for the interval, even though $\hat{p}$ is also the test statistic of interest. Data-dependent measures of connection strength are also not without precedent in community detection methodology. In the modularity score, for instance, empirical degrees are stand-ins for the (unknown) expected degrees of the Chung-Lu/configuration model.

Illustrating choices of T and $\mathbb{P}_\theta$ in practical situations, the examples laid out in Section 2.1 are continued:

1. To complete the NST framework for binary undirected networks, first recall that in this setting, the network data is simply $\mathbf{D} = A$, the adjacency matrix. We continue to write $\mathbf{D}$ to conform to generic notation. In this setting, we use the test statistic

$$T(u, B, \mathbf{D}) := \sum_{v \in B} A[u, v], \tag{2.6}$$

and the Chung-Lu null model with parameter $\theta = (\bar{d}(1), \ldots, \bar{d}(n))$. In practice, θ is set to the empirical degrees $(d(1), \ldots, d(n))$. Let $\mathbb{E}$ denote expectation under $\mathbb{P}$, the true generating model of $\mathbf{D}$. Let $\mathbb{E}_\theta$ denote expectation under $\mathbb{P}_\theta$. Then

$$a(u, B) := \bar{d}(u, B) - \bar{d}(u)\bar{q}(B) = \mathbb{E}\Big[T(u, B, \mathbf{D})\Big] - \mathbb{E}_\theta\Big[T(u, B, \widetilde{\mathbf{D}})\Big]$$

Hence, if $a(u, B)$ is positive, the p-value in (2.5) will be stochastically larger than uniform.

2. Finishing the NST framework for correlation networks, we recall that $\mathbf{D} = \mathbf{X}$, the Euclidean data for the nodes $[d]$, and define the test statistic

$$T(u, B, \mathbf{D}) := \sum_{v \in B} r(u, v), \tag{2.7}$$

where $r(u, v)$ is the Pearson correlation between node u and v. Regarding the null model, recall that $\mathbf{X}$ is assumed to have true population correlations Σ with general entry $\rho(u, v)$. Under the null corresponding to the hypotheses (2.4), we assume that $a(u, B) := \sum_{v \in B} \rho(u, v) = 0$. If in fact $a(u, B)$ is positive, however, the p-value in (2.5) will be stochastically larger than uniform. Note that the distribution of $T(u, B, \widetilde{\mathbf{D}})$ depends on Σ. Therefore, we calculate the p-value in (2.5) with the parameter $\theta = \hat{\Sigma}$, the estimated correlation matrix of $\mathbf{X}$.

Derivations of closed-form expressions for the p-values in the examples above are part of their respective applications of NST, and outside the scope of this framework. Wilson et al. (2014) provided an asymptotic result for (2.6), which facilitated an approximate p-value in that setting. Bodwin et al. (2015) gave a central limit theorem for a test-statistic similar to (2.7) for the mining of differential correlation. Similar theoretical results are major components of my work on the NST methods, introduced in the subsequent chapters of this thesis.

2.3 The Stable Community Search (SCS) algorithm

For arbitrary networks $\mathcal{G} = ([n], \mathbf{D})$, we now have the tools to determine the significance of the empirical association of any node $u \in [n]$ and any node set $B \subseteq [n]$, via the p-value $p(u, B, \mathbf{D})$. The usage of this p-value function is within a community extraction algorithm based on multiple-testing. Given a candidate community B, we "update" B by taking in (or keeping) significantly connected nodes, as judged by $p(u, B, \mathbf{D})$, and by leaving out (or expelling) others. Explicitly, the update procedure can be written as the map $U_\alpha : 2^{[n]} \times \mathcal{D} \mapsto 2^{[n]}$, defined:

Core update U_α

Given a network $\mathcal{G} = ([n], \mathbf{D})$ and input set $B \subseteq [n]$:

1. Calculate p-values $\mathbf{p} := \{p(u, B, \mathbf{D}) : u \in [n]\}$.

2. Obtain threshold $\tau(\mathbf{p})$ from a multiple-testing procedure.

3. Return $B' := \{u : p(u, B, \mathbf{D}) \leqslant \tau(\mathbf{p})\}$.

Remark: Many different multiple testing rules yielding τ are available, the most stringent being the well-known Bonferroni correction. The choice of the multiple-testing rule should be situation-specific, and will be addressed as needed in later chapters.

The update U_α is an exploratory tool for moving an input set B closer to a true community. Consider that, if the initial set B has a majority group of nodes from a community C (as in Definition 1), the association $a(u, B)$ will be positive for $u \in C$, and non-positive otherwise. If T or some transformation thereof is a good estimator of a, the statistics $\{T(u, B, \mathbf{D})\}_{u\in[n]}$ will be large (or positive) for $u \in C$, and small (or negative) otherwise. Hence, U_α applied to B will return many nodes in C, and few nodes in C^c. Indeed, ideally, we should expect $U_\alpha(C, \mathbf{D})$ to return C, given strong enough signal in the data.

The preceding reasoning motivates an algorithm that searches for "stable communities" C satisfying $U_\alpha(C, \mathbf{D}) = C$. By definition, all interior nodes of a stable community C are significantly connected to C, and exterior nodes are not. Therefore, a stable community can be thought of as an "empirical" true community. We define a stable community search procedure, which iteratively applies U_α until convergence:

Stable Community Search (SCS) algorithm

Given a network $\mathcal{G} = ([n], \mathbf{D})$ and initial set $B \subseteq [n]$:

1. Set $t = 1$, $B_t = B$, and $B_0 := \phi$.

2. If $B_t = B_{t'}$ for any $t' < t$, terminate and return B_t.

3. Set $B_{t+1} \leftarrow U_\alpha(B_t, \mathcal{G})$ and $t \leftarrow t + 1$. Go to step 2.

Since the number of possible subsets B_t is finite, SCS is guaranteed to terminate. If $t' = t - 1$, then $U_\alpha(B_t, \mathcal{G}) = B_t$, and B_t is a stable community. If $t' < t$, the algorithm has reached a stable *sequence* of communities. In full detail, SCS will terminate in one of two states:

1. With a stable community C, satisfying $U_\alpha(C, \mathcal{G}) = C$.

2. With a stable sequence of sets $B_1, \ldots, B_J$ satisfying

$$U_\alpha(B_1, \mathcal{G}) = U_\alpha(B_2, \mathcal{G}) = \ldots = U_\alpha(B_J, \mathcal{G}) = U_\alpha(B_1, \mathcal{G}).$$

Stable sequences are somewhat problematic, as the each set in the sequence is, of course, not stable itself. However, stable sequences are quite rare in practice. Furthermore, the stable sequences that do arise are usually short, with highly overlapping node sets. Therefore, the constituent sets of a stable sequence can still be of practical interest, for a given application. In Appendix A.1, an algorithm is provided to resolve stable sequences which addresses these considerations.

2.4 Background and Type-I Error

Arguably, many nodes in real-world networks do not truly belong to any community. In this thesis, these nodes are termed "background", as in Wilson et al. (2014). Within the NST framework, the notion of a background node can be formally defined as follows:

Definition 2. *Let $\mathcal{G}$ be a random network with distribution $\mathbb{P}$. Let $a(u, B)$ be the population association between u and B corresponding to $\mathbb{P}$. Then $u \in [n]$ is a* ***background*** *node if and only if $a(u, B) = 0$ for all $B \subseteq [n]$.*

Remark. Note that $a(u, B) < 0$ would imply that u is *negatively* associated with B, and is therefore not an example of null behavior. Node sets that are mutually *negatively* associated are often said exhibit "disassortive" structure, which, though important in some contexts (Aicher et al., 2014), is not as often of scientific interest. Though the approach introduced in this chapter focuses on assortive structure, it lays the groundwork for extraction of disassortive communities, as well. Extraction of disassortive communities could be accomplished by using *left*-tail rather than right-tail p-values.

A real-world example of a background node in a network otherwise laden with communities is a happenstance friend in a social network. Suppose you travel to a new city, and happen to meet and spend some time with a local, or maybe fellow traveler. Subsequently, you decide to "friend" this person on an online social platform. It is unlikely that your new friend will have significant connectivity to your existing friend groups, like your college or high school friends, or your work friends. Nonetheless, upon analysis of your social network, the classical partition-based community detection methods discussed in Sections 1.2.1-1.2.4 will force this node into some community.

In contrast, the SCS algorithm is naturally suited to handle background nodes. Recall that, by definition, for any background node $u \in [n]$, we have $a(u, B) = 0$ for all $B \subseteq [n]$. Thus, for any $B \subseteq [n]$, the statistic $T(u, B, \mathbf{D})$ will follow the null model, hence $p(u, B, \mathbf{D}) \sim U[0, 1]$. If U_α employs an appropriate multiple testing rule, the proportion of background nodes in $U_\alpha(B, \mathcal{G})$ should therefore be bounded in either expectation or probability, depending on the rule. The classical multiple-testing rule is the well-known Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995), defined as follows:

1. Given a set of p-values $\mathbf{p} := \{p_u\}_{u \in [n]}$ and a target FDR $\alpha \in (0, 1)$.
2. Calculate the adjusted p-values $p^*_u := n\, p_u \,/\, j(u)$, where $j(u)$ is the rank of p_u in $\mathbf{p}$.
3. Compute threshold $\tau(\mathbf{p}) := \max\{p_u : p^*_u \leqslant \alpha\}$.

Benjamini and Hochberg (1995) show that if the p-values are mutually independent, this procedure ensures that the expected proportion of false discoveries, or in our context, the expected proportion of background nodes in $U_\alpha(B, \mathcal{G})$, is no greater than α.

2.4.1 Global Error Control

While use of the multiple-testing rule has direct implications about false discoveries in SCS output sets, its implications for the global error of SCS are not immediately clear. In particular, we define a *null* random network as follows:

Definition 3. *Let $\mathcal{G}$ be a random network with distribution $\mathbb{P}$. Let $a(u, B)$ be the population association between u and B corresponding to $\mathbb{P}$. Then $\mathcal{G}$ is a **null** network if and only if $a(u, B) = 0$ for all $u \in [n]$ and $B \subseteq [n]$.*

Note that, by Definition 2, $\mathcal{G}$ consists completely of background nodes. We now consider the probability that SCS recovers *any* stable community in a null network. Ideally, any application of the SCS algorithm on such a network will converge to the empty set. However, this is not guaranteed, due to random fluctuations in the data. Explicitly, let $\mathcal{G} = ([n], \mathbf{D})$ be a random network with distribution $\mathbb{P}$. Define the set of stable communities in $\mathcal{G}$ as

$$\mathcal{C}(\mathbf{D}, \alpha) := \{B \subseteq [n] : U_\alpha(B, \mathbf{D}) = B\} \tag{2.8}$$

We define global Type I error at level α as $\mathbb{P}(\mathcal{C}(|\mathbf{D}, \alpha)| > 0)$. One key assumption is needed on $\mathbb{P}$ to bound the Type-I error at α.

Assumption 1. *For $B \subseteq [n]$, denote the set of p-values used by the update U_α by*

$$\mathbf{p}(B) := \{p(1, B, \mathbf{D}), \ldots, p(n, B, \mathbf{D})\}.$$

Assume that under $\mathbb{P}$, for all $B \subseteq [n]$, the p-values $\mathbf{p}(B)$ are independent and uniformly distributed.

The following theorem establishes that, under Assumption 1, the Type-I error is bounded by α if U_α uses the Benjamini-Hochberg rule. The proof is given in Section A.2.

Theorem 4. (OST global error control)

Fix $\alpha \in [0, 1]$ and $n > 1$. Let $\mathcal{G} = ([n], \mathbf{D})$ be a random network with distribution $\mathbb{P}_n$. Assume $\mathbb{P}_n$ satisfies Assumption 1. Then if U_α uses the Benjamini-Hochberg multiple testing procedure, $\mathbb{P}\big(\, |\mathcal{C}(\mathbf{D}, \alpha)| > 0 \,\big) \leqslant \alpha$.

2.4.2 Discussion

The impact of Theorem 4 is powerful. First, note that the theorem makes no reference to the type of network data. It depends only on the existence of a global null model that provides uniform and independent p-values for the node-set test statistic. Second, the theorem depends only on the update $U_\alpha(\,\cdot\,, \mathcal{G})$, not the SCS algorithm overall. Thus, the SCS algorithm should be viewed simply as *one* way to search for fixed points. Importantly, the choice of initialization method for the SCS algorithm, or ways to resolve cycles (discussed at the end of Section 2.3) do not have an effect

on global Type-I error. Theorem 4 regards the probability of the *existence* of fixed points, which always upper-bounds our probability of finding them.

Of course, Theorem 4 has limitations. First, Assumption 1 is almost never satisfied exactly, even when the network is completely null as in Definition 3. Small dependencies between the tests arise due to inherent dependencies in network data. That said, in some simple cases it is easy to see that these dependencies vanish uniformly with the size of the network. Consider a *directed* binary network $\mathcal{G} := ([n], A)$, noting that here, A is not symmetric. Without loss of generality we assume the rows of A contain the *in*-edges indicators of the network. The natural node-to-set association statistic in this setting is then the *in*-degree of u to B, defined

$$\vec{D}(u, B, A) := \sum_{v \in B} A[u, v].$$

Consider that the statistics $\{\vec{D}(1, B, A), \ldots, \vec{D}(n, B, A)\}$ are, in this setting, mutually independent. Therefore, if the p-values $\mathbf{p}(B) := \{p(1, B, A), \ldots, p(n, B, A)\}$, were exact, they would also be mutually independent. However, as with the binary-network example laid out in the previous sections, a reasonable null model in this setting depends on the expected degrees of the assumed generative model of $\mathcal{G}$, which are unknown and must be estimated from the data. In particular, each p-value $p(u, B, A)$ is a function of (and only of) $\vec{D}(u, B, A)$ and the observed in degrees $\{\vec{D}(1), \ldots, \vec{D}(n)\}$ where $\vec{D}(u) := \vec{D}(u, [n], A)$ for all $u \in [n]$. Thus, for any finite n, the p-values are dependent through the observed degrees.

However, as n approaches infinity, these dependencies vanish uniformly, since (in many standard cases) the observed degrees will approach their limiting values. Thus, loosely speaking, the p-values will be *asymptotically* exact and mutually independent. For instance, suppose that the unknown generative model of $\mathcal{G}$ is a directed Erdős-Rényi network with edge probability 0.5. Under this model, $\vec{D}(u)$ is a Binomial$(n, 1/2)$ random variable, and from Bernstein's Inequality and a union bound it follows that

$$\max_{u \in [n]} \left\{ |\vec{D}(u) - n/2| \right\} \xrightarrow{p} 0 \quad \text{as} \quad n \to \infty.$$

In practice, test-wise dependency issues for finite n can potentially be resolved by using more stringent FDR control procedures, like that proposed by Benjamini and Yekutieli (2001). Variants of Theorem 4 that involve these procedures are immediate areas for future work.

Another issue that limits the scope of Theorem 4 is that evaluation of tail probabilities for reasonable node-set test statistics T often involve asymptotic distributional approximations. This means that for finite n and any node set B, p-values will not be precisely uniform under the null. The non-uniformity issue cannot be solved in generality, since every application of SCS involves a different test statistic and, therefore, a different method (approximate or otherwise) of calculating p-values. Thus, Theorem 4 does not perfectly guarantee Type-I error under a global null in any application that involves a distributional approximation. However, *empirical* Type-I error control for the methods presented in this paper has been verified in a variety of simulation settings, as will be shown in subsequent chapters.

CHAPTER 3

Continuous Configuration Model Extraction

In this chapter, the Node-Set Testing framework introduced in Chapter 2 is applied to the problem of community detection on weighted networks. The centerpiece of this application is the introduction of a weighted network null model that allows for arbitrary degrees and (separately) arbitrary weighted degrees. Such a model is currently absent from the literature. The null, called the "continuous configuration model", allows for rigorous statistical tests of graph statistics for weighted networks. In particular, we define an NST test statistic for weighted networks, and apply the continuous configuration model to yield a testing-based extraction method called CCME. The continuous configuration model also serves as a useful tool for simulation. A benchmarking framework introduced in the following sections involves simulated networks with both communities and background nodes simulated under the null. Such networks with both communities and background are crucial for validating the performance of community detection methods on realistic data.

The rest of this chapter is organized as follows. Chapter-specific notation is introduced in Section 3.1. In Section 3.2, the continuous configuration model is motivated and stated. In Section 3.3, the NST test statistic is defined, and theoretical results are given establishing its limiting distribution and asymptotic consistency properties. The overall implementation and application of the core NST algorithm within CCME is described in Section 3.4. Evaluations of CCME's empirical efficacy on simulations and real data are presented in Sections 3.5 and 3.6 (respectively). A discussion is offered in Section 3.7.

3.1 Notation and terminology

This section gives notation unique to this chapter, which will differ slightly from that presented in Section 1.1. We denote an undirected weighted network on n nodes by a triple $\mathcal{G} := ([n], E, w)$, where $[n] := \{1, \ldots, n\}$ is the node set, E is the edge set, consisting of all unordered node-pairs

$\{u, v\}$ for which there is an edge between u and v, and $w : [n] \times [n] \mapsto [0, \infty)$ is a symmetric function that assigns a non-negative weight to each pair of nodes with $w(u, v) = 0$ if $\{u, v\} \notin E$. The degree of a node u is defined by $d(u) := \sum_{v \in [n]} \mathbb{1}(\{u, v\} \in E)$, and we denote the vector of node degrees by $\mathbf{d} = (d(1), \ldots, d(n))$. In an analogous fashion, we define the *strength* of a node by $s(u) := \sum_{v \in [n]} w(u, v)$, and the strength vector of the network by $\mathbf{s} = (s(1), \ldots, s(n))$. The total degree and strength of $\mathcal{G}$ are given by $d_T := \sum_{v \in [n]} d(v)$ and $s_T := \sum_{v \in [n]} s(v)$, respectively.

3.2 The continuous configuration model

To understand the continuous configuration model, it helps to consider the intuition behind, and use of, the binary configuration model for unweighted networks. The configuration model was discussed in detail in Section 1.2.2; a brief reminder of that discussion is given here. The binary configuration model for an n-node network is based on a given degree vector $\mathbf{d}$ corresponding to the nodes. Studied originally in Bollobás (1980) and Bender (1974), the model is equivalent to a process in which each node u receives $d(u)$ half-edges, which are paired uniformly-at-random without replacement until no half-edges remain (Molloy and Reed, 1995). In other words, the model guarantees a graph with degrees $\mathbf{d}$ but otherwise uniformly distributed edges. Thus, given an observed network with degrees $\mathbf{d}$, a typical draw from the configuration model under $\mathbf{d}$ represents that network without any community structure. This suggests that communities in an observed binary network should be defined by node sets having intra-connectivity significantly beyond what is expected under the model. Indeed, that is exactly how this model is employed in community detection, beginning with the introduction of the modularity metric (see Section 1.2.3).

The degrees $\mathbf{d}$ of the configuration model can be thought of as the nodes' relative propensities to form ties. Chung and Lu made this notion explicit by defining a Bernoulli-based model for a n-node unweighted network with a given expected degree sequence (Chung and Lu, 2002). Under this model, the probability of nodes u and v sharing an edge is exactly $d(u)d(v)/d_T$. As null model for community detection, the Chung-Lu and configuration are often equated (Durak et al., 2013). Indeed, for sparse graphs it can be shown that the probability of an edge between u and v under the configuration model is approximately the Chung-Lu probability. The *continuous* configuration model, introduced in this section, extends the spirit of the configuration and Chung-Lu models

by taking both observed degrees $\mathbf{d}$ and strengths $\mathbf{s}$ as node propensities for (respectively) edge connection and edge weight. We define the model explicitly in the following sub-section. Some new notation will be needed. Given a vector $\mathbf{x}$ of dimension n, we define for any indices $u, v \in [n]$ the ratio

$$r_{uv}(\mathbf{x}) := \frac{x(u)x(v)}{\sum_u x(u)} \tag{3.1}$$

Define $\tilde{r}_{uv} := \min(1, r_{uv}(\mathbf{x}))$. Note that when $\mathbf{x}$ is a degree sequence $\mathbf{d}$, $r_{uv}(\mathbf{d})$ is the Chung-Lu probability of an edge between nodes u and v. Finally, for a vector $\mathbf{y}$ of dimension n, define $f_{uv}(\mathbf{x}, \mathbf{y}) := r_{uv}(\mathbf{y})/\tilde{r}_{uv}(\mathbf{x})$.

3.2.1 Model statement

The continuous configuration model on n nodes has the parameter set $\theta := (\mathbf{d}, \mathbf{s}, \kappa)$, where $\mathbf{d} \in [n]^n$ is a degree vector, $\mathbf{s} \in [0, \infty)^n$ is a strength vector, and $\kappa > 0$ is a variance parameter. Let F be a distribution on the non-negative real line with mean one and variance κ. The model specifies a random weighted graph $\mathcal{G} := ([n], E, W)$ on n nodes as follows:

1. $\mathbb{P}(\{u, v\} \in E) = \tilde{r}_{uv}(\mathbf{d})$ independently for all node pairs $u, v \in [n]$

2. For each edge $\{u, v\} \in E$, generate an independent random variable $\xi(u, v)$ according to F, and assign edge weights by:

$$W(u,v) = \begin{cases} f_{uv}(\mathbf{d}, \mathbf{s})\xi(u,v) & \{u,v\} \in E \\ 0 & \{u,v\} \notin E \end{cases}$$

The edge generation defined by step 1 is equivalent to the Chung-Lu model: edge indicators are Bernoulli, with probabilities adjusted by the propensities $\mathbf{d}$. The weight generation in step 2 mirrors this process. Edge weights follow the distribution F, with means adjusted by the propensities $\mathbf{s}$, through $f(\mathbf{d}, \mathbf{s})$. It is easily derived from the model that

$$P(\{u, v\} \in E) = \min\left\{1, \frac{d(u)d(v)}{d_T}\right\} \quad \text{and} \quad \mathbb{E}\left(W(u,v)\right) = \frac{s(u)s(v)}{s_T}, \tag{3.2}$$

equations which extend the binary-network notion of null behavior to edge weights. Furthermore, if $r(u,v) \leqslant 1$ for all $u, v \in [n]$, the equations in 3.2 imply that

$$\mathbb{E}\left(D(u)\right) = d(u) \quad \text{and} \quad \mathbb{E}\left(S(u)\right) = s(u) \;\; \text{for all} \;\; u \in [n]. \tag{3.3}$$

where $D(u)$ and $S(u)$ are (random) degree and strength of u under the model. Thus, the continuous configuration model can be thought of as a random weighted graph with given expected degrees and given expected strengths.

3.2.2 Null specification of the model

As described in Section 2.2, for the continuous configuration model to be used in an NST algorithm, we must specify its parameter set θ based on the data. Given an observed network $\mathcal{G}$, we straightforwardly use the observed degrees and strengths $\mathbf{d}$ and $\mathbf{s}$ as the first two parameters of the model. The third parameter of the continuous configuration model, κ, is also computed from the $\mathcal{G}$, and meant to capture its observed average edge-weight variance. We use the following method-of-moments estimator to specify κ:

$$\hat{\kappa}(\mathbf{d}, \mathbf{s}) := \sum_{\{u,v\} \in E} \left(w(u,v) - f_{uv}(\mathbf{d}, \mathbf{s})\right)^2 / \sum_{\{u,v\} \in E} f_{uv}(\mathbf{d}, \mathbf{s})^2 \tag{3.4}$$

This estimator is derived as follows. Define the edge indicator $e_{uv} := \mathbb{1}(\{u,v\} \in E)$, and note that under the continuous configuration model with $\mathbf{d}$ and $\mathbf{s}$,

$$\mathrm{Var}\left(W(u,v) \,\middle|\, e_{uv}\right) \;=\; f_{uv}(\mathbf{d}, \mathbf{s})^2 \,\mathrm{Var}\left(\xi_{uv}\right) \;=\; f_{uv}(\mathbf{d}, \mathbf{s})^2 \,\kappa. \tag{3.5}$$

Therefore

$$\begin{aligned} \mathbb{E}\left\{ \sum_{\{u,v\} \in E} \left(W(u,v) - f_{uv}(\mathbf{d}, \mathbf{s})\right)^2 \,\middle|\, E \right\} &= \sum_{\{u,v\} \in E} \mathrm{Var}\left(W(u,v) \,\middle|\, e_{uv}\right) \\ &= \kappa \sum_{\{u,v\} \in E} f_{uv}(\mathbf{d}, \mathbf{s})^2, \end{aligned}$$

Dividing through by $\sum_{\{u,v\} \in E} f_{uv}(\mathbf{d}, \mathbf{s})$ motivates equation 3.4.

Strictly speaking, the distribution F is also a parameter of the model. However, for the purposes of the NST approach to weighted networks, we do not require a null specification of F. As we discuss in the next section, p-values from the model will be based on a central limit theorem which requires only a third-moment assumption on F. While estimating F could improve the model's efficacy as a null, in general this would require potentially costly computational procedures, and additional theoretical assumptions that might be difficult to support or verify in practice. The specification of F will be most useful for applications of the model that involve simulations or likelihood-based analyses.

3.3 Test statistic and theoretical results

Following the NST framework introduced in Chapter 2, a summary statistic and an associated significance test under the continuous configuration model is now established. These components are to be used in an SCS algorithm for weighted networks. The final sub-sections consist of theoretical analyses of the statistic both under the null and under a model with planted communities.

3.3.1 A test statistic for node-set association in weighted networks

As a natural statistic of connectivity between a node $u \in [n]$ and a set $B \subseteq [n]$ in a weighted network $\mathcal{G} = ([n], E, W)$, we define the sum of edge weights as

$$S(u, B, \mathcal{G}) := \sum_{v \in B} W(u, v). \tag{3.6}$$

We now begin to apply the NST framework laid out in Section 2.1. To do so, we assume that any observed weighted network $\mathcal{G}$ is generated by an unknown random weighted network model with distribution $\mathbb{P}$. Our NST framework association quantity $a(u, B)$ will then be the true expected value of $S(u, B, \mathcal{G})$ under $\mathbb{P}$, minus its expected value under the continuous configuration null model. This is analogous to the adaptation of NST to binary networks, illustrated in Section 2.2. To define $a(u, B)$ explicitly, we first give the expected value (and variance) of $S(u, B, \mathcal{G})$ under the null:

Proposition 5. *Let $\mathcal{G} = ([n], E, W)$ be a random network generated by the continuous configuration model with parameters $\theta = (\mathbf{s}, \mathbf{d}, \kappa)$. For any $(u, B) \in [n] \times 2^{[n]}$, let $\mu(u, B|\theta)$ and $\sigma(u, B|\theta)$ be,*

respectively, the mean and standard deviation of $S(u, B, \mathcal{G})$ under $\mathcal{G}$. Then

$$\mu(u, B|\theta) \equiv \mu(u, B|\mathbf{s}) = \sum_{v \in B} r_{uv}(\mathbf{s}) \tag{3.7}$$

and

$$\sigma(u, B|\theta)^2 = \sum_{v \in B} r_{uv}(\mathbf{s}) f_{uv}(\mathbf{d}, \mathbf{s}) \left(1 - \tilde{r}_{uv}(\mathbf{d}) + \kappa\right) \tag{3.8}$$

The proof, given in Appendix B.1, follows from easy calculations with the model's generating procedure (see Section 3.2.1). Denote the expected value under $\mathbb{P}$ by $\mathbb{E}$, let $\mathbf{S}$ be the (random) strengths from $\mathcal{G}$ under $\mathbb{P}$, and define $\bar{\mathbf{s}} := \mathbb{E}(\mathbf{S})$. Similarly to examples in Section 2.2, we then define our NST association quantity as follows:

$$a(u, B) := \mathbb{E}\left\{S(u, B, \mathcal{G})\right\} - \mu(u, B|\bar{\mathbf{s}}). \tag{3.9}$$

Of course, $\bar{\mathbf{s}}$ is not observed. Analogously to examples in Section 2.2, we base our inference about $S(u, B, \mathcal{G})$ on the continuous configuration model with parameter θ containing plug-in values computed from the data. In particular, let $\theta = (\mathbf{D}, \mathbf{S}, \hat{\kappa}(\mathbf{D}, \mathbf{S}))$, where $\mathbf{D}$ and $\mathbf{S}$ are the degrees and strengths from $\mathcal{G}$, and $\hat{\kappa}$ is the variance estimator defined in Section 3.2.2. Following the NST framework (see Equation 2.5), we assess the (one-sided) statistical significance of $S(u, B, \mathcal{G})$ with the p-value

$$p(u, B, \mathcal{G}) := \mathbb{P}_\theta(S(u, B, \widetilde{\mathcal{G}}) > S(u, B, \mathcal{G})), \tag{3.10}$$

where $\widetilde{\mathcal{G}}$ is a random weighted network distributed according to $\mathbb{P}_\theta$. Note that, above, $\mathcal{G}$ is not random with respect to $\mathbb{P}_\theta$; the operator $\mathbb{P}_\theta$ acts as a distribution function on the observed value of $S(u, B, \mathcal{G})$.

Unfortunately, a closed-form expression for the p-value in (3.10) is analytical only for the most trivial specifications of θ. Therefore, the p-values used for the update $U_\alpha(\,\cdot\,, \mathcal{G})$ cannot be calculated exactly, in most cases. To overcome this issue, we seek test statistic for $S(u, B, \mathcal{G})$ with an asymptotic normal distribution. This analysis is the focus of the next section.

3.3.2 Asymptotic Normality of $S(u, B, \mathcal{G})$

A central limit theorem is now established for the summary statistic introduced in Section 3.3.1, yielding a closed-form approximation for the p-value (3.10). In the setting of the theorem, for any $n > 1$, a random network $\mathcal{G}_n$ is generated by a continuous configuration model with parameter $\theta_n := (\mathbf{d}_n, \mathbf{s}_n, \kappa_n)$ and common weight distribution F. The following regularity conditions are required on the sequence $\{\theta_n\}_{n>1}$. Let λ_n denote the average entry of $\mathbf{d}_n$, (which is the average expected degree of $\mathcal{G}_n$). For each $r \geqslant 0$ let $L_{n,r} := n^{-1} \sum_{u\in[n]} (d_n(u)/\lambda_n)^r$ be the normalized r^{th}-moment of $\mathbf{d}_n$. Note that $L_{n,1} = 1$. The regularity conditions are then as follows:

Assumption 2. *There exists $\beta > 0$ such that, with $e_n(u|\beta) := s_n(u)/d_n(u)^{1+\beta}$,*

$$0 < \liminf_{n\to\infty} \min_{u\in[n]} e_n(u|\beta) \quad \textit{and} \quad \limsup_{n\to\infty} \max_{u\in[n]} e_n(u|\beta) < \infty.$$

Assumption 3. *Let β be as in Assumption 2. There exists $\varepsilon > 0$ such that, for both $r = 4\beta + 2$ and $r = 4\beta + 2 + \varepsilon$,*

$$0 < \liminf_{n\to\infty} L_{n,r} \quad \textit{and} \quad \limsup_{n\to\infty} L_{n,r} < \infty$$

Assumption 4. $\limsup_{n\to\infty} \sup_{u,v\in[n]} r_{uv}(\mathbf{d}_n) < \infty.$

Assumption 5. *The sequence $\{\kappa_n\}_{n\geqslant 1}$ is bounded away from zero and infinity, and F has finite third moment.*

Assumption 2 reflects the common relationship between strengths and degrees in real-world weighted networks (Barrat et al. (2004); Clauset et al. (2009)). Assumptions 3-4 are needed to control the extremal behavior of the degree distribution. They exclude, for instance, cases with a few nodes having $d_n(u) \asymp n$ and the remaining nodes having $d_n(u) = O(1)$. We note that the Assumption 3 becomes more stringent as β increases, since as β increases the strength-degree power law becomes more severe.

Theorem 6. *For each $n > 1$, let $\mathcal{G}_n$ be generated by the continuous configuration model with parameter θ_n and weight distribution F. Suppose $\{\theta_n\}_{n\geqslant 1}$ and F satisfy Assumptions 2-5. Fix a node sequence $\{u_n\}_{n\geqslant 1}$ with $u_n \in [n]$ and a positive integer sequence $\{b_n\}_{n\geqslant 1}$ with $b_n \leqslant n$. Suppose*

$d_n(u_n)b_n/n \to \infty$ as $n \to \infty$. Let $B_n \subseteq [n]$ be a node set chosen independently of $\mathcal{G}_n$ according to the uniform distribution on all sets of size b_n. Then

$$\frac{S(u_n, B_n, \mathcal{G}_n) - \mu_n(u_n, B_n|\theta_n)}{\sigma_n(u_n, B_n|\theta_n)} \Rightarrow \mathcal{N}(0,1) \quad as \quad n \to \infty \tag{3.11}$$

The proof is given in Section B.6 of the supplemental. Essentially, Theorem 6 says that $S(u, B, \mathcal{G})$ is asymptotically Normal provided that B is "typical" and that $d(u)$ and B are sufficiently large. The theorem justifies the following approximation of the p-value in (3.10):

$$p(u, B) \approx 1 - \Phi\left(\frac{S(u, B, \mathcal{G}) - \mu(u, B|\theta)}{\sigma(u, B|\theta)}\right) \tag{3.12}$$

Above, θ is specified from the observed network $\mathcal{G}$, as described in Section 3.2.2. With this p-value, the SCS algorithm laid out in Section 2.3 can now be applied to weighted networks. In practice, we find that using this approximation allows the algorithm to filter between background nodes and nodes in real communities (see Section 3.5).

3.3.3 Consistency of SCS

In this section, we evaluate the ability of the SCS algorithm based on the continuous configuration model to identify true communities in a planted-community model. Explicitly, we consider a sequence of networks $\{\mathcal{G}_n\}_{n>1}$ where each network in the sequence is generated by a weighted stochastic block model (WSBM). The WSBM we employ is similar to that presented in Aicher et al. (2014), but is generalized to include node-specific weight parameters. In other words, it is "strength-corrected" as well as degree-corrected, in a manner analogous to the original degree-corrected SBM (Coja-Oghlan and Lanka, 2009). The proofs of Theorem 8 and Theorem 9 are given in Appendix B.7.

3.3.3.1 The weighted stochastic block model

For fixed $K > 0$, we define a K-block WSBM on $n > 1$ nodes as follows. Let $\mathbf{c}_n$ be a community partition vector with $c_n(u) \in [K]$ giving the community index of u. Denote community i by $C_{i,n} := \{u : c_n(u) = i\}$. Define $\pi_{i,n} := n^{-1}|C_{i,n}|$ with $\boldsymbol{\pi}_n$ the associated vector. Let $\mathbf{P}$ and $\mathbf{M}$

be fixed $K \times K$ matrices with non-negative entries encoding intra- and inter-community baseline edge probabilities and edge weight expectations, respectively. Let ϕ_n and ψ_n be arbitrary n-vectors with positive entries, which are parameters giving nodes individual propensities to form edges and assign weight (independent of $\mathbf{P}$ and $\mathbf{M}$). To ensure proper edge probabilities, we assume that $\max(\phi_n)^2 \max(\mathbf{P}) \leqslant 1$. For identifiability, we assume the vectors ϕ_n and ψ_n sum to n. Finally, let F be a distribution on the positive real line with mean 1 and variance $\sigma^2 \geqslant 0$. The WSBM can then be specified as follows:

1. Place edge $\{u, v\}$ with probability $\mathbb{P}_n(\{u, v\} \in E) = r_{uv}(\phi_n)\mathbf{P}[c_n(u), c_n(v)]$, independently across node pairs.

2. For each edge $\{u, v\} \in E$, generate an independent random variable $\xi(u, v)$ according to F. Determine edge weight $W(u, v)$ by:

$$W(u,v) = \begin{cases} 0 & \text{if } \{u,v\} \notin E \\ f_{uv}(\psi_n, \phi_n)\mathbf{M}[c_n(u), c_n(v)]\xi(u,v) & \text{if } \{u,v\} \in E \end{cases}$$

The many parameters involved with this model allow for node heterogeneity and community structure. When $\mathbf{P}$ and $\mathbf{M}$ are proportional to a $K \times K$ matrix of ones, the WSBM reduces to the continuous configuration model with parameters $\mathbf{d} \propto \phi$, $\mathbf{s} \propto \psi$, and $\kappa = \sigma^2$. Community structure is introduced in the network by allowing the diagonal entries of $\mathbf{P}$ and $\mathbf{M}$ to be arbitrarily larger than the off-diagonals.

3.3.3.2 Consistency theorem

Fix $K > 0$. For our consistency analysis of SCS, we consider of a sequence of random networks $\{\mathcal{G}_n\}_{n>1}$, where $\mathcal{G}_n$ is generated by a K-community WSBM. In this setting, we incorporate an additional parameter ρ_n, and let $\mathbf{P}_n := \rho_n \mathbf{P}$ replace $\mathbf{P}$ for each $n > 1$. This lets us distinguish the role of the asymptotic order of the average expected degree, defined $\lambda_n := n\rho_n$, from the profile of the edge densities within and between communities ($\mathbf{P}$). Importantly, our results require only that $\lambda_n / \log n \to \infty$, reflecting the sparsity of real-world networks. Throughout this section, we denote the vector of (random) strengths from $\mathcal{G}_n$ by $\mathbf{S}_n$.

We now define an explicit notion of consistency in terms of the SCS algorithm. Recall from Chapter 2 that for fixed FDR $\alpha \in (0,1)$, a set $C \subseteq [n]$ from a network $\mathcal{G}_n$ is a stable community if $U_\alpha(C, \mathcal{G}_n) = C$.

Definition 7. *We say that SCS is consistent for a sequence of WSBM random networks $\{\mathcal{G}_n\}_{n>1}$ if for any FDR level $\alpha \in (0,1)$, the probability that the true communities $C_{1,n}, \ldots, C_{K,n}$ are stable communities approaches 1 as $n \to \infty$.*

To assess the conditions that allow a target set C to be a stable community, we seek more general conditions under which the update $U_\alpha(\,\cdot\,, \mathcal{G})$ outputs C given any initial set B. If $U_\alpha(B, \mathcal{G}_n) = C$, all nodes $u \in C$ must have significant connectivity to B, as judged by the p-value approximation defined in 3.12. It is clear from that p-value expression that, for the update to return C, the test statistic $S(u, B, \mathcal{G}_n)$ must be significantly larger than $\mu(u, B|\mathbf{S}_n)$, its expected value under the continuous configuration model. Therefore, our first result hinges on asymptotic analysis of that deviation, which we denote by

$$A(u, B, \mathcal{G}_n) := S(u, B, \mathcal{G}_n) - \mu_n(u, B|\mathbf{S}_n). \tag{3.13}$$

Note that the "population" value of $A(u, B, \mathcal{G}_n)$, with all random variables replaced by their expected values, is precisely our NST association quantity (see Equation 3.9) between u and B under $\mathcal{G}_n$. Explicitly, let $\mathbb{E}_n$ denote expectation under $\mathcal{G}_n$, and define $\bar{\mathbf{s}}_n := \mathbb{E}\mathbf{S}_n$. The true association between any $u \in [n]$ and $B \subseteq [n]$ under $\mathcal{G}_n$ is then $a_n(u, B) := \mathbb{E}_n S(u, B, \mathcal{G}_n) - \mu(u, B|\bar{\mathbf{s}}_n)$. Hence, if $a_n(u, B) > 0$, u should be included in the set update. The purpose of Theorem 8 is to make this notion rigorous. The theorem is stated in terms of a normalization of a_n, defined

$$\tilde{a}_n(u, B) := \lambda_n^{-1} a_n(u, B), \tag{3.14}$$

where λ_n is the order of the average expected degree. Given a sequence of initial sets $\{B_n\}_{n>1}$ and target sets $\{C_n\}_{n>1}$, Theorem 8 establishes that $U_\alpha(B_n, \mathcal{G}_n) = C_n$ with probability approaching 1 if $a_n(u, B)$ is bounded away from zero, and is positive if and only if $u \in C_n$. The theorem requires the following two assumptions:

Assumption 6. *There exist constants $m_+ > m_- > 0$ such that, for all $n > 1$, the entries of ϕ_n, ψ_n, $\mathbf{P}$, $\mathbf{M}$, and $\boldsymbol{\pi}_n$ are all bounded in the interval $[m_-, m_+]$.*

Assumption 7. *F is independent of n and has support $(0, \eta)$ with $\nu < \infty$.*

Assumption 6 is standard in consistency analyses involving block models (e.g. Zhao et al. (2012), Bickel and Chen (2009)). Note that it does not imply constant edge density, as the sparsity parameter ρ_n is allowed to vanish. Assumption 7 allows the use of Bernstein's inequality throughout the proof, but may be relaxed if there are constraints on the moments of F allowing the use of a similar inequality.

Theorem 8. *Fix $K > 1$. For each $n > 1$, let $\mathcal{G}_n$ be a n-node random network generated by a K-community WSBM with parameters satisfying Assumptions 6 - 7. Suppose $\lambda_n / \log n \to \infty$. Let $\{B_n\}_{n>1}$, $\{C_n\}_{n>1}$ be sequences of node sets satisfying the following: there exist constants $q \in (0, 1]$ and $\Delta > 0$ such that for all n sufficiently large, $|B_n|, |C_n| \geqslant qn$, and*

$$\tilde{a}_n(u, B_n) \geqslant \Delta, \;\; u \in C_n, \quad \text{and} \quad \tilde{a}_n(u, B_n) \leqslant -\Delta, \;\; u \notin C_n. \tag{3.15}$$

Then if the update U_α uses the p-value approximation given in Equation (3.12),

$$\mathbb{P}_n\big(U_\alpha(B_n, \mathcal{G}_n) = C_n\big) \to 1 \text{ as } n \to \infty.$$

To prove the consistency of SCS, we show that condition 3.15, when it involves the community sequence, is guaranteed by a concise condition on the model parameters. Let $\tilde{\pi}_{i,n} := \sum_{v \in C_{i,n}} \psi_n(v)$, and $\tilde{\boldsymbol{\pi}}_n$ the vector of $\tilde{\pi}_{i,n}$'s. The consistency theorem requires the following additional assumption, an analog to which can be found in Zhao et al. (2012) for consistency of modularity under the degree-corrected SBM:

Assumption 8. *$\tilde{\boldsymbol{\pi}}_n \equiv \tilde{\boldsymbol{\pi}}$ does not depend on n.*

Assumption 8 can be ignored with the minor complication that (3.16) must then hold for sufficiently large n. Define $\mathbf{H} := \mathbf{P} \cdot \mathbf{M}$, the entry-wise product. Note that when ϕ and ψ are proportion to 1-vectors, $\mathbb{E}[W(u, v)] = \mathbf{H}[c(u), c(v)]$ for all $u, v \in [n]$. Thus, the interpretation of $\mathbf{H}$

is as the baseline inter/intra-community weight expectations after integrating out edge presence. Defining $\tilde{\mathbf{\Pi}} := \tilde{\boldsymbol{\pi}}\tilde{\boldsymbol{\pi}}^t$, we state the consistency theorem:

Theorem 9. *Fix $K > 0$. Let $\{\mathcal{G}_n\}_{n>1}$ be a sequence of networks generated by a K-community WSBM satisfying Assumptions 6-8. Suppose that the matrix*

$$\mathcal{M} := \mathbf{H} - \frac{\mathbf{H}\tilde{\mathbf{\Pi}}\mathbf{H}}{\tilde{\boldsymbol{\pi}}^t\mathbf{H}\tilde{\boldsymbol{\pi}}} \tag{3.16}$$

has positive diagonal entries and negative off-diagonal entries. If $\lambda_n/\log n \to \infty$, SCS is consistent for $\{\mathcal{G}_n\}_{n>1}$.

Understanding of condition 3.16 begins with the consideration of the case $K = 2$, when it reduces to the requirement that $\mathbf{H}[1,1]\mathbf{H}[2,2] > \mathbf{H}[1,2]^2$. Interestingly, this condition implies that when (for example) we have

$$\mathbf{H} = \begin{bmatrix} 0.5 & 0.25 \\ 0.25 & 0.25 \end{bmatrix},$$

U_α is consistent for both communities, even though community 1 nodes have the same baseline weight expectation to community 2 nodes as they do to other community 1 nodes. To see why, consider a WSBM with the $\mathbf{H}$ matrix above and ψ proportional to the 1-vector, inducing homogenous within-community expected strengths. Under this model, the expected strengths from community 2 are less than those from community 1. Therefore, under the corresponding continuous configuration model, we expect nodes from community 2 to have *higher* baseline weight expectation to nodes from community 1.

In some sense, then, the matrix $\mathcal{M}$ reveals whether or not appropriate signal exists in the model, with respect to the continuous configuration null. Notice that this signal need not be present in both $\mathbf{P}$ and $\mathbf{M}$. For instance, the condition would be satisfied if $\mathbf{H}$ is a scalar multiple of $\mathbf{M}$, that is, if $\mathbf{P}$ is proportional to the $\mathbf{1}$-matrix. This entails that SCS is consistent even when the edge structure of $\mathcal{G}_n$ is Erdős-Renyi, as long as the edge weight signal is assortative. Of course, the opposite also holds, namely that SCS is consistent even when assortative community signal is only present in $\mathbf{P}$.

3.3.3.3 Connection to weighted modularity and related work

The conditions of Theorems 8-9 have a deep relationship to weighted modularity (WM), which is the modularity metric with degrees replaced by strengths (Newman, 2004a). For fixed $n > 1$, let $\mathbf{c}$ be any partition of $[n]$. Define $K := \max\{\mathbf{c}\}$ and $C_u := \{v : c(v) = c(u)\}$. Then the (random) WM of $\mathbf{c}$ on $\mathcal{G}_n$ can be written

$$\begin{aligned} Q_n^w(\mathbf{c}) &:= \frac{1}{S_{n,T}} \sum_{uv \in [n]} \{W(u,v) - r_{uv}(\mathbf{S}_n)\} \, \mathbb{1}\{c(u) = c(v)\} \\ &= \frac{1}{S_{n,T}} \sum_{i \in [K]} \sum_{c(u)=c(v)} W(u,v) - r_{uv}(\mathbf{S}_n) = \frac{1}{S_{n,T}} \sum_{u \in [n]} \sum_{v \in C_u} W(u,v) - r_{uv}(\mathbf{S}_n) \\ &= \frac{1}{S_{n,T}} \sum_{u \in [n]} S_n(u, C_u) - \mu_n(u, C_u | \mathbf{S}_n) = \frac{1}{S_{n,T}} \sum_{u \in [n]} A(u, C_u, \mathcal{G}_n) \end{aligned}$$

Thus, the contribution of u to WM with its assignment C_u is precisely the random association from u to C_u. Writing the population WM as $\bar{q}_n^w(\mathcal{C}) := n^{-1} \sum_u \tilde{a}_n(u, C_u)$, it can be shown that condition (3.16) implies q_n^w is maximized by $\mathcal{C}_n$, the true community partition.

The consistency analysis of the (binary) modularity metric under the degree-corrected SBM, provided by Zhao et al. (2012), similarly hinges on maximization of population modularity. It is unsurprising, then, that the parameter condition for their result can be (analogously) expressed as a fixed $K \times K$ matrix having positive diagonals and negative off-diagonals. In fact, if the WSBM parameter $\mathbf{M}$ is proportional to a matrix of 1s, and the parameter ψ is a scalar multiple of ϕ, condition 3.16 in Theorem 9 is equivalent to the parameter assumptions on modularity consistency in Zhao et al. (2012). Furthermore, their analysis also requires that $\lambda_n / \log n \to \infty$. However, the definition of consistency and proof approach for the theorems here are entirely novel.

3.4 The Continuous Configuration Model Extraction algorithm

In the previous section, we established an asymptotic result showing that ground-truth communities are, with high probability, fixed points of the SCS algorithm using the test statistic $S(u, B, \mathcal{G})$ and the p-value approximation in (3.12). This result demonstrates the in-principle sensibility of the search algorithm. In practice, we must rely on local, heuristic algorithms for initialization and termination, as with other exploratory methods. For instance, k-means is often used to initialize the

EM algorithm, and modularity is often locally maximized through agglomerative pairing (Clauset et al., 2004). We incorporate SCS in a general community detection method for weighted networks entitled Continuous Configuration Model Extraction (CCME), written in loose detail as follows:

The CCME Community Detection Method for Weighted Networks

1. Given an observed weighted network $\mathcal{G}$, obtain initial node sets $\mathcal{B}_0 \subseteq 2^{[n]}$.

2. Apply SCS to each node set in $\mathcal{B}_0$, resulting in stable communities $\mathcal{C}$.

3. Remove sets from $\mathcal{C}$ that are empty or redundant.

We describe steps 1 and 3 in more detail below (step 2 was covered thoroughly in Section 2.3). Importantly, the method has no connection to any graph-partition criteria. It proceeds solely by the SCS algorithm, which assesses communities independently. This allows CCME to adaptively return overlapping communities and background nodes.

3.4.1 Step 1: Initialization

Just as estimation procedures for mixture-models can be initialized with heuristic techniques like k-means, it is possible to initialize CCME with partition-based community detection algorithms. However, we have observed this approach to perform somewhat poorly in practice. Instead, we initialize with a novel, more thorough algorithm based on the continuous configuration model. For a fixed node $u \in [n]$, and any node $v \in [n]$, we define

$$z_u(v) := \frac{w(u,v) - f_{uv}(\mathbf{s},\mathbf{d})}{\sqrt{\theta} f_{uv}(\mathbf{s},\mathbf{d})} \ \vee \ 0$$

The measure $z_u(v)$ acts like a truncated z-statistic, quantifying the extremity of the weight $w(u,v)$. The initial node set corresponding to u, denoted by $B_0(u)$, is formed by sampling $d(u)$ nodes with replacement from $[n]$ with probability proportional to $z_u(v)$. The intuition behind this procedure is that if u is part of a highly-connected node set C, then $z_u(v)$ for nodes $v \in C$ will be larger (on average) than for other nodes.

3.4.2 Step 3: Filtering of $\mathcal{C}$

The CCME community detection method returns a final collection of communities $\mathcal{C}$, containing the results of the SCS algorithm for each initial set in $\mathcal{B}_0$. By default, we remove any empty or duplicate sets from $\mathcal{C}$. In some applications, pairs of sets in $\mathcal{C}$ will have high Jaccard similarity. In Appendix B.2, we detail a method of pruning these near-duplicates from $\mathcal{C}$. Additionally, in Appendix B.2, we describe routines to suppress the application of SCS to initial sets that are "weakly" intra-connected, or with high overlap to already-extracted communities. These routines greatly reduce the runtime of CCME, and, on some simulated networks, improve accuracy.

3.5 Simulations

This section contains a performance analysis of CCME and existing methods on a benchmarking simulation framework. Simulated networks are generated from the Weighted Stochastic Block Model (see Section 3.3.3.1), with slight modifications to include overlapping communities and background nodes, when necessary. The performance measures, competing methods, simulation settings, and results are described in the subsections below.

3.5.1 Performance measures and competing methods

To assess the performance of a community detection method, three measures are used:

1. **Overlapping Normalized Mutual Information (oNMI):** Introduced by Lancichinetti et al. (2009), oNMI is an information-based measure between 0 and 1 that approaches 1 as two covers of the same node set become similar and equals 1 when they are the same. From a method's results, we calculate oNMI with respect to the true communities *only* for the nodes the method placed into communities.

2. **Community nodes in background (%C.I.B.):** The percentage of true community nodes incorrectly assigned to background.

3. **Background nodes in communities (%B.I.C.):** The percentage of true background nodes (if present) incorrectly placed into communities.

In addition to CCME, two other weighted-network methods capable of identifying overlapping nodes are assessed. One of these is OSLOM (Lancichinetti et al., 2011), as described in Section 1.3.1. The other is SLPAw, a weighted-network version of a recent overlapping label propagation algorithm Xie et al. (2011). Also included are three commonly used partition-based methods implemented the `R` package `igraph` (Csárdi and Nepusz, 2006): Fast-Greedy, which performs local modularity optimization via a hierarchical agglomeration (Clauset et al., 2004); Walktrap, an agglomerative algorithm that locally maximizes a score based on random walk theory (Pons and Latapy, 2006); Infomap, an information-flow mapping algorithm that uses random walk transition probabilities (Rosvall and Bergstrom, 2008).

Remark 1. Being extraction methods, only CCME and OSLOM directly specify background nodes. For all other methods, we manually specify background nodes as all nodes in singleton communities. However, these methods almost never returned singleton communities, even under weak or non-existent signal.

Remark 2. A more thorough simulation study would include other modularity maximization algorithms, especially those not implemented in `R`. However, as CCME is coded in `R`, the easiest initial benchmarking step is to compare it with other community detection algorithms with implementations in the same language. Furthermore, we have not yet implemented weighted stochastic block model fitting algorithms (e.g. Aicher et al. (2014)) into the benchmarking study, as initial attempts to run these algorithms proved computationally intensive.

3.5.2 Simulation settings and results

In this section an overview of the simulation procedure for the benchmarking framework is given. We relegate precise details to Appendix B.3. We first describe "default" parameter settings of the WSBM; in the simulation settings below, individual parameters are toggled around their default values, to reveal the dependence of the methods to those parameters. At each unique parameter setting, 20 random networks were simulated. The points in each plot from Figure 3.1 show the average performance measure of the methods over the 20 repetitions.

The default WSBM setting has the number of nodes at $n = 5,000$. The community memberships were set by obtaining community sizes from a power law, then assigning nodes uniformly at random. This process produced approximately 3 to 7 communities per network. Full details are

provided in Appendix B.4.1. Recall the parameters $\mathbf{P}$ and $\mathbf{M}$, which induce baseline intra- and inter-community edge and weight signal. In the default setting, these matrices have off-diagonals equal to 1 and diagonals equal to constants $s_e = 3$ and $s_w = 3$ (respectively). In some simulation settings, overlapping and background nodes are added (as described later in this section), but the default setting includes neither overlap nor background.

Common parameter settings. For all simulated networks (regardless of the setting), the node-wise edge parameters ϕ were drawn from a power law to induce degree heterogeneity. The parameter ϕ is scaled so that the expected average degree of each network was equal to $\sqrt{n}$, which induces sparsity in the network. The parameter ψ is set by the formula $\psi = \phi^{1.5}$ to ensure a non-trivial relationship between strengths and degrees. See Appendix B.4.3 for full detail.

3.5.2.1 Networks with varying signal levels

The first setting tested the methods' dependence on s_e and s_w. These values were moved along an even grid on the range $[1, 3]$. Plots A-1 and B-1 in Figure 3.1 show the performance measure results when s_w is fixed at 3, plots A-2 and B-2 show results when $s_e = 3$, and plots A-3 and B-3 show results when s_e and s_w are moved along $[1, 3]$ together. Many methods had large oNMI scores in this simulation setting. We transformed the oNMI scores using the function

$$\text{t-oNMI}_a(x) := (\tfrac{1}{1-x+a} - \tfrac{1}{1+a})/(\tfrac{1}{a} - \tfrac{1}{1+a})$$

with $a = 0.05$. This is a monotonic, one-to-one transformation from $[0, 1]$ to itself, which stretches the region close to 1, allowing a clearer comparison between the methods' performances. CCME consistently out-performed all competing methods, especially when either the edge or weight signal was completely absent.

The plots in row B show that when either s_e or s_w were near 1, OSLOM and CCME assigned many background nodes. This is consistent with these methods' unique abilities to leave nodes unassigned when they are not significantly connected to communities. That said, %C.I.B. can be seen as a measure of sensitivity, since ideally no nodes would be assigned to background when any signal is present. In this regard, CCME outperformed OSLOM, sometimes handily.

3.5.2.2 Networks with overlapping communities

The second setting involved networks with overlapping nodes. To add overlapping nodes to the default network, two parameters were introduced: o_n, the number of overlapping nodes, and o_m, the number of memberships for each overlapping node. The particular overlapping nodes and community memberships were chosen uniformly-at-random. This closely follows a simulation approach taken by Lancichinetti et al. (2011). Plots C-1 and C-2 show performance results from the setting with o_n moving away from 0 and $o_m = 2$. Plot C-3 shows results from the setting with $o_n = 500$ and $o_m \in \{1, \ldots, 4\}$. We find that CCME consistently outperforms all methods in terms of accuracy (oNMI), and outperforms OSLOM in terms of sensitivity (%C.I.B.).

3.5.2.3 Networks with overlapping communities and background nodes

The final simulation setting involved networks with both overlap and background nodes. The number of background nodes was fixed at 1,000, and number of community nodes varied from $n = 500$ to $n = 5,000$. For each network, $o_n = n/4$ nodes were randomly chosen to overlap $o_m = 2$ communities (also chosen at random). Background nodes were created by first simulating the n-node community sub-network, and then generating the 1,000-node background sub-network according to the continuous configuration model, using empirical degrees and strengths from the community sub-network. The complete details of this procedure are given in the Appendix B.5.

The results of this simulation setting are shown in row D from Figure 3.1. From plot D-1, we see that OSLOM and CCME had the highest oNMI scores, favoring OSLOM when the number of community nodes decreased. Because this simulation setting involved background nodes, the %B.I.C. metric is relevant, and can be taken as a measure of specificity: ideally, nodes from the background sub-network should be excluded from communities. From plot D-2, we see that methods incapable of assigning background had %B.I.C. equal to 1. We found that CCME correctly ignored background nodes as the network size increased, whereas OSLOM became increasingly *anti*-conservative for larger networks. Furthermore, CCME again had lower %C.I.B. than OSLOM.

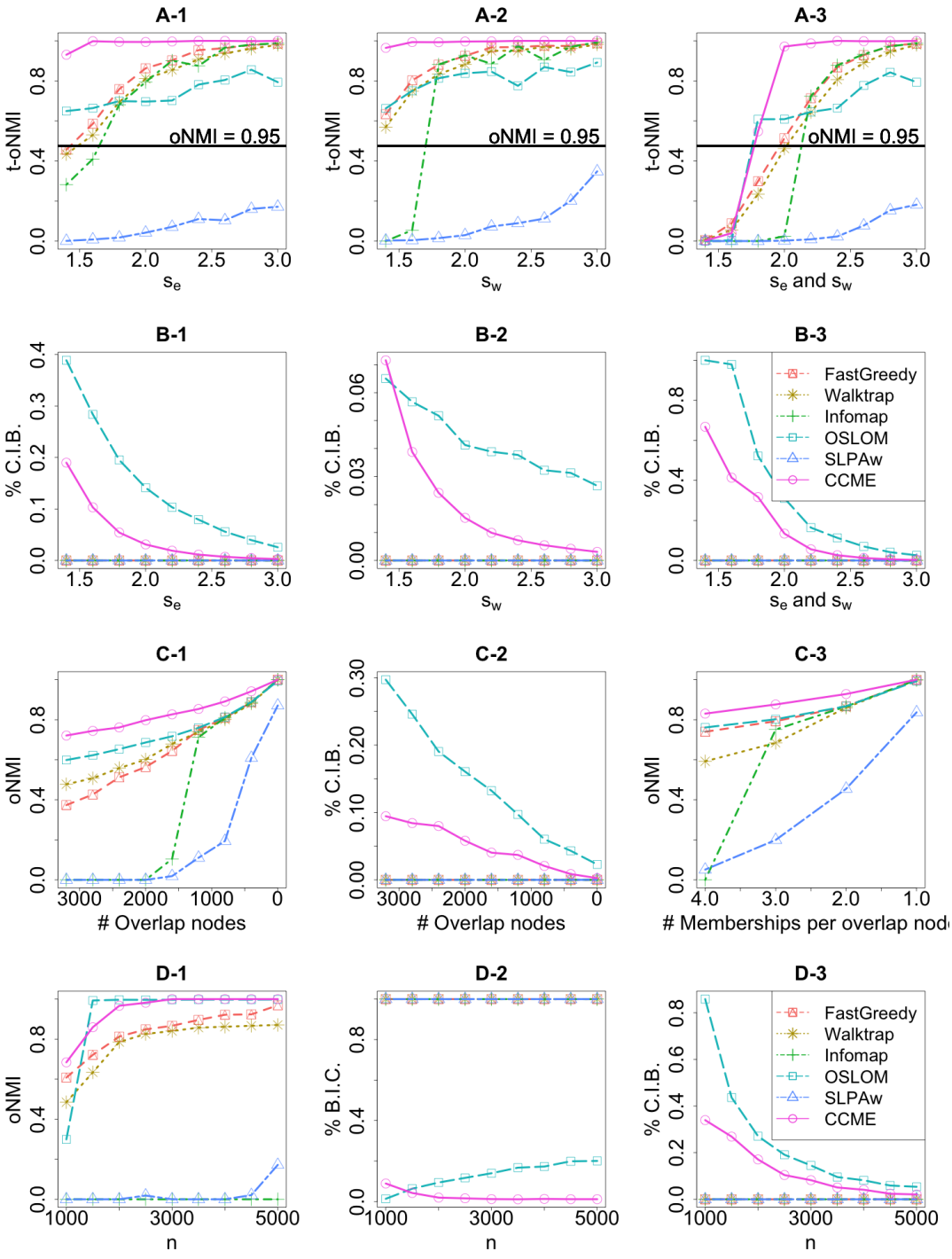

Figure 3.1: Simulation results described in Sections 3.5.2.1-3.5.2.3. Legends correspond to all plots.

3.6 Applications

In this section, results from CCME, OSLOM, and SLPAw (the methods capable of returning overlapping communities) on two real data sets are discussed.

3.6.1 U.S. airport network data

The first application involves commercial airline flight data, obtained from the Bureau of Transportation Statistics (www.transtats.bts.gov). For each month from January to July of 2015, we created a weighted network with U.S. airports as nodes, edges connecting airports that exchanged flights, and edges weighted by aggregate passenger count. We also constructed a year-aggregated network, formed simply by taking the union of the month-wise edge sets, and adding the month-wise weights.

With the monthly data, OSLOM and CCME tended to find communities consistent with geography, whereas SLPAw placed most of the network into one community. With the year-aggregated data, OSLOM also agglomerated most airports, whereas CCME continued to respect the geography. Since the aggregated data is much more edge-dense, this suggests the performance of OSLOM and SLPA may suffer on weighted graphs with high or homogeneous edge-density, whereas CCME is able to detect proper community structure from the weights alone. This aligns with the simulation results described in Section 3.5.2.1. In Figure 3.2, we display the methods' results when applied to the June and year-aggregated data sets from 2015. We provide results from other months in Figures B.3-B.5 of the Appendix.

3.6.2 ENRON email network

An email corpus from the company ENRON was made available in 2009. In the networks literature, the un-weighted network formed by linking communicating email addresses is well-studied; see www.cs.cmu.edu/~./enron for references and Leskovec et al. (2010b) for the data. For the purposes of this paper, we derived a *weighted* network from the original corpus, using message count between addresses as edge weights. Though the corpus was formed from email folders of 150 ENRON executives, we made the network from addresses found in *any* message. This full network has 80,702 nodes, comprised of a majority of non-ENRON addresses, and likely many spam or irrele-

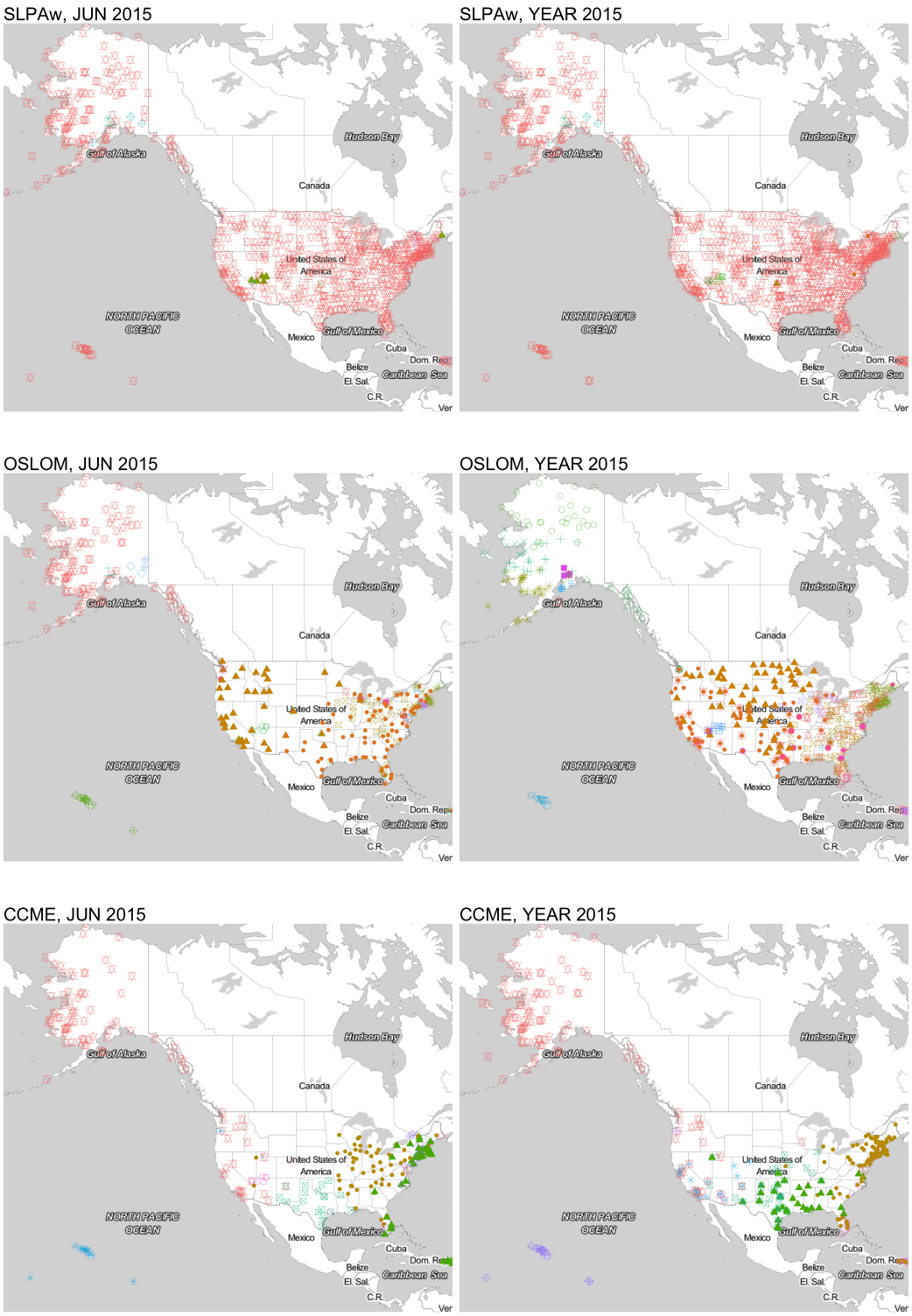

Figure 3.2: SLPAw, OSLOM, and CCME results from June 2015 and 2015-year-aggregated U.S. airport networks. Maps created with `ggmap` (Kahle and Wickham, 2013)

vant senders. Thus the network has many potential "true" background nodes. We applied CCME, OSLOM, and SLPAw to the network to see which methods best focused on company-specific areas of the data.

Tables 3.1 and 3.2 give basic summaries of the results, which show noticeable differences between the outputs of the methods. CCME placed far fewer into nodes into communities, but detected larger communities with more overlapping nodes. Notably, CCME had the highest percentage of ENRON addresses among nodes it placed into communities (see Table 3.3). These results suggest that CCME was more sensitive to critical relationships in the network.

Table 3.1: Metrics from methods' results on ENRON network: number of communities, minimum community size, median community size, maximum community size, count of nodes in any community

	Num.Comms	Min.size	Med.size	Max.size	Num.Nodes
CCME	185	2	687	5416	14552
OSLOM	405	2	19	770	17635
SLPAw	2138	2	4	4793	79316

Table 3.2: Metrics from methods' results on ENRON network: number of overlapping nodes, minimum # of memberships, median # of mem'ships, max. # of mem'ships

	Num.OL.Nodes	Min.mships	Med.mships	Max.mships
CCME	8104	2	9	78
OSLOM	462	2	2	8
SLPAw	3860	2	2	4

Table 3.3: Top domains associated with community nodes from each method, by proportion

CCME.Domains	Prop.
enron.com	0.784
aol.com	0.008
cpuc.ca.gov	0.006
pge.com	0.004
socalgas.com	0.003
dynegy.com	0.003

OSLOM.Domains	Prop.
enron.com	0.529
aol.com	0.029
haas.berkeley.edu	0.016
hotmail.com	0.015
yahoo.com	0.009
jmbm.com	0.005

SLPAw.Domains	Prop.
enron.com	0.423
aol.com	0.039
hotmail.com	0.023
yahoo.com	0.016
haas.berkeley.edu	0.007
msn.com	0.006

3.7 Discussion

This chapter introduced the continuous configuration model, the first null model for community detection on weighted networks. The continuous configuration model allows for the construction of a flexible and powerful community extraction method called CCME, via the NST framework. It was shown that a standardized statistic for CCME tests is asymptotically normal, a result which enables an analytic approximation to p-values used in the method. Another theorem established

asymptotic consistency under a weighted stochastic block model for the stable-community search procedure inherent to CCME.

On simulated networks the proposed method CCME is competitive with the best existing community detection methods. CCME was the dominant method for simulated networks with large numbers of overlapping nodes. Furthermore, on networks with background nodes, CCME was the only method to correctly label true background nodes while maintaining high detection power and accuracy for nodes belonging to communities. On real data, CCME gave results that accorded with known (external) features of the data set.

The continuous configuration model may have applications outside the setting of this paper. One may investigate the distributional properties of many different graph-based statistics under the model, as a means of assessing statistical significance in practice. For instance, the appropriate theoretical analysis, for which Theorem 6 may be a precedent, could yield an approach to the assessment of statistical significance of weighted modularity. Another benefit of an explicit null for weighted networks is the potential for simulation. Using the continuous configuration model, and parts of the framework presented in this paper, one can generate weighted networks having true background nodes with arbitrary expected degree and strength distributions.

Software. The `R` code for the CCME method is available in the github repo 'jpalowitch/CCME'. The code for reproducing the analyses in Sections 3.5 and 3.6 is available at the github repo 'jpalowitch/CCME_analyses'.

CHAPTER 4

Multi-layer Community Extraction

This chapter is devoted to the multilayer network community extraction method introduced briefly in Section 1.4.2. Importantly, the method applies to *un*-weighted multilayer networks only. We note this because the last chapter was devoted to a study of weighted single-layer networks. Henceforth in this chapter, all discussion of networks will pertain to the un-weighted, multi-layer setting. The first section presents a multilayer network null model and a significance score for multi-layer vertex sets. In Section 4.2, theoretical results are introduced regarding the asymptotic consistency properties of the proposed score. Section 4.3 provides a detailed description of the Multilayer Extraction procedure that incorporates the score.

4.1 Significance-based scoring of a vertex-layer group

Seeking a vertex partition that optimizes, or approximately optimizes, an appropriate score function is a standard approach to single layer community detection (e.g. (Newman, 2006b; Wang and Wong, 1987; Chung, 1997); see Section 1.2.3 for a complete discussion). Rather than scoring a partition of the available network, Multilayer Extraction makes use of a significance based score that is applicable to individual vertex-layer sets. In the following sections, we describe the multilayer null model, and then the proposed score. First, we define some notation that will be used exclusively for this chapter and Appendix C. Let $\mathbf{G}(m,n) = ([n],[m],(E_1,\ldots,E_m))$ be an observed (m,n)-multilayer network, with $[n] = 1,\ldots,n$ the node set, $[m] = 1,\ldots,m$ the layer set, and E_ℓ the edge set of layer $\ell \in [m]$. For each layer $\ell \in [m]$ and pair of vertices $u, v \in [n]$, let

$$x_\ell(u,v) = \mathbb{1}(\{u,v\} \in E_\ell)$$

indicate the presence or absence of an edge between u and v in layer ℓ of $\mathbf{G}(m,n)$. The *degree* of a vertex $u \in [n]$ in layer ℓ, denoted by $d_\ell(u)$, is the number of edges incident on u in G_ℓ. Formally,

$$d_\ell(u) \;=\; \sum_{v \in [n]} x_\ell(u,v).$$

The *degree sequence* of layer ℓ is the vector $\boldsymbol{d}_\ell = (d_\ell(1), \ldots, d_\ell(n))$ of degrees in that layer; the full degree set of $\mathbf{G}(m,n)$ is the list $\boldsymbol{d} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m)$ containing the degree sequence of each layer in the network. Define as $d_{T,\ell} := \sum_{u \in [n]} d_\ell(u)$ the total degree in layer ℓ. In this section, a partition vector $\mathbf{c}$ retains its meaning from Chapters 1-3, but we denote its elements by c_u instead of $c(u)$.

4.1.1 The Null Model

Our significance-based score for vertex-layer sets in multilayer networks is based on comparing observed graph statistics with a null distribution, as in the NST framework (Chapter 2). The null model we use for this algorithm is best described as a multi-layer configuration model, where each layer of a random network is generated (independently) according to the single-layer configuration model. The single-layer configuration model was described in detail in Section 1.2.2. We give a brief reminder of the model, using the notation from this setting. Let $\mathcal{G}(m,n)$ denote the family of all (m,n)-multilayer networks. Given the degree sequence $\boldsymbol{d}$ of the observed network $\mathbf{G}(m,n)$, we define a multilayer configuration model and an associated probability measure $\mathbb{P}_{\boldsymbol{d}}$ on $\mathcal{G}(m,n)$, as follows. In layer G_1, each node is given $d_1(u)$ half-edges. Pairs of these half-edges are then chosen uniformly at random, to form edges until all half-edges are exhausted (disallowing self-loops and multiple edges). This process is done for every subsequent layer $G_2, \ldots, G_m$ independently, using the corresponding degree sequence from each layer.

Under the above null model, each layer is generated according to the Molloy and Reed (1995) algorithm for the single-layer configuration model Bollobás (1980); Bender (1974). The probability of an edge between nodes u and v in layer ℓ depends only on the degree sequence $\boldsymbol{d}_\ell$ of the observed graph G_ℓ. The distribution $\mathbb{P}_{\boldsymbol{d}}$ has two complementary properties that make it useful for identifying communities in an observed multilayer network: (i) it preserves the degree structure of the observed network; and (ii) subject to this restriction, edges are assigned at random. As discussed in 1.2.2,

these characteristics have caused configuration model to have long been taken as the appropriate null model against which to judge the quality of a proposed community partition.

The configuration model is the null model which motivates the modularity score of a partition in a network (Newman, 2004b, 2006b). Recall the modularity score, discussed in Section 1.2.3:

$$Q(\mathbf{c}) := \frac{1}{d_T} \sum_{u,v \in [n]} \left(x(u,v) - \frac{d(u)d(v)}{d_T} \right) \mathbb{1}(c(u) = c(v)) \tag{4.1}$$

A brief reminder of the motivation for this score is as follows. Above, the ratio $d(u)d(v)/d_T$ is the approximate expected number of edges between u and v under the configuration model. If the partition $\mathbf{c}$ represents communities with a large observed intra-edge count relative to what is expected under the configuration model, it receives a high modularity score. The identification of the communities that (approximately) maximize the modularity of a partition is among the most common techniques for community detection in networks.

4.1.2 Multilayer Extraction Score

Rather than scoring a partition, the Multilayer Extraction method scores individual vertex-layer sets. We define a multilayer node score that is based on the single-layer modularity score (4.1) and amenable to iterative maximization. We first define a local *set* modularity for a collection of vertices $B \subseteq [n]$ in the layer $\ell \in [m]$:

$$Q_\ell(B) := \frac{1}{n\binom{|B|}{2}^{1/2}} \sum_{u,v \in B : u<v} \left(x_l(u,v) - \frac{d_\ell(u)d_\ell(v)}{d_{T,\ell}} \right) \tag{4.2}$$

The scaling term in the equation above is related to the total number of vertices in the network and the total number of possible edges between the vertices in B. This score is one version of the various set-modularities considered in Fasino and Tudisco (2016), and is reminiscent of the *local* modularity score introduced in Clauset et al. (2004).

The Multilayer Extraction procedure seeks communities that are *assortative* across layers, in the sense that $Q_\ell(B)$ is large and positive for each $\ell \in L$. In light of this, we define the *multilayer set score* as

$$H(B,L) := \frac{1}{|L|} \left(\sum_{\ell \in L} Q_\ell(B)_+ \right)^2, \tag{4.3}$$

where Q_+ denotes the positive part of Q. Generally speaking, the score acts as a yardstick with which one can measure the connection strength of a vertex-layer set. Large values of the score signify densely connected communities.

We note that the multilayer score $H(B, L)$ is reminiscent of a chi-squared test-statistic computed from $|L|$ samples. That is, under appropriate regularity assumptions on $Q_\ell(B)$, the score in (4.3) will be approximately chi-squared with one degree of freedom.

4.2 Consistency Analysis

In this section I evaluate the asymptotic consistency of the multi-layer local modularity score. Existing work on consistency of community detection algorithms was given comprehensive treatment in Section 1.3.3. In Section 4.2.1, I introduce a multi-layer version of the Stochastic Block Model (SBM; see Section 1.2.1), which is the model under which the consistency of the score will be established. In Section 4.2.2, I describe and state the consistency results. Section 4.2.3 contains the proofs of the results.

4.2.1 The Multilayer Stochastic Block Model

We assess the consistency of Multilayer Extraction under the multilayer stochastic block model (MLSBM) with two vertex communities, defined as a probability distribution $\mathbb{P}_{m,n} = \mathbb{P}_{m,n}(\cdot|\boldsymbol{P}, \pi_1, \pi_2)$ on the family of undirected multilayer networks with m layers, n vertices and 2 communities. The distribution is fully characterized by (i) the containment probability π_1 for community 1, and (ii) a sequence of symmetric 2×2 matrices $\boldsymbol{P} = \{P_1, \ldots, P_m\}$ where $P_\ell = \{P_\ell(i, j)\}$ with entries $P_\ell(i, j) \in (0, 1)$. Under the distribution $\mathbb{P}_{m,n}$, a random multilayer network $\widehat{\boldsymbol{G}}(m, n)$ is generated using two simple steps:

1. A subset of $\lceil \pi_1 n \rceil$ vertices are placed in community 1, and remaining vertices are placed in community 2. Each vertex u in community j is assigned a community label $c_u = j$, forming a partition vector $\mathbf{c} \in \{1, 2\}^n$.

2. An edge is placed between nodes $u, v \in [n]$ in layer $\ell \in [m]$ with probability $P_\ell(c_u, c_v)$, independently from pair to pair and across layers, and no self-loops are allowed.

For a fixed n and m, the community labels $\mathbf{c}_n = (c_1, \ldots, c_n)$ are chosen once and only once, and the community labels are the same across each layer of $\widehat{\boldsymbol{G}}(m, n)$. On the other hand, the inner and intra community connection probabilities (and hence the assortativity) can be different from layer to layer, introducing heterogeneity among the layers. Note that when $m = 1$, the MLSBM reduces to the (single-layer) stochastic block model from Wang and Wong (1987).

4.2.2 Consistency of the Score

We evaluate the consistency of the Multilayer Extraction score under the MLSBM described above. Our first result addresses the vertex set maximizer of the score given a fixed layer set $L \subseteq [m]$. Our second result (Theorem 12 in Section 4.2.2.1) leverages the former to analyze the global maximizer of the score across layers and vertex sets. Explicitly, consider a multilayer network $\widehat{\boldsymbol{G}}(m, n)$ with distribution under the multilayer stochastic block model $\mathbb{P}_{m,n} = \mathbb{P}_{m,n}(\boldsymbol{P}, \pi_1, \pi_2)$. For a fixed vertex set $B \subseteq [n]$ and layer set $L \subseteq [m]$, define the random score by

$$\widehat{H}(B, L) := \frac{1}{|L|} \left(\sum_{\ell \in L} \widehat{Q}_\ell(B)_+ \right)^2,$$

where $\widehat{Q}_\ell(B)$ is the set-modularity of B in layer ℓ under $\mathbb{P}_{m,n}$. Our main results address the behavior of $\widehat{H}(B, L)$ under various assuptions on the parameters of the MLSBM.

Toward the first result, for a fixed layer set $L \subseteq [m]$, let $\widehat{B}_{opt}(n)$ denote the node set that maximizes $\widehat{H}(B, L)$ (if more than one set does, any may be chosen arbitrarily). To define the notion of a "misclassified" node, for any two sets $B_1, B_2 \subseteq [n]$ let $d_h(B_1, B_2)$ denote the Hamming distance (rigorously defined as the cardinality of the symmetric difference between B_1 and B_2). We then define the number of misclassified nodes by a set B by

$$\texttt{Error}(B) := d_h(B, C_1) \wedge d_h(B, C_2).$$

Note that this definition accounts for arbitrary labeling of the two communities. As the nodes and community assignments are registered across layers, neither d_h nor `Error` depend on the choice of L. Before stating the main theorem, we define a few quantities that will be used throughout its statement and proof:

Definition 10. *Let "*det*" denote matrix determinant. For a fixed layer set $L \subseteq [m]$, define*

$$\delta_\ell := \det P_\ell \quad \delta(L) := \min_{\ell \in [L]} \delta_\ell \quad \pi := (\pi_1, \pi_2)^t \quad \kappa_\ell := \pi^T P_\ell \pi \quad \kappa(L) := \min_{\ell \in [L]} \kappa_\ell \tag{4.4}$$

The value δ_ℓ determines, by its positivity, whether or not the community structure in layer ℓ will be detectable in the limit. Hence, a requirement of the theorem below will be that $\delta(L) > 0$. κ_ℓ is the (normalized) overall average edge density of layer ℓ. Note that in this asymptotic setting, we assume that π_1 does not change with n. We now state the fixed-layer-set consistency result:

Theorem 11. *Fix m and let $\{\widehat{\boldsymbol{G}}(m,n)\}_{n>1}$ be a sequence of multilayer stochastic 2 block models where $\widehat{\boldsymbol{G}}(m,n)$ is a random graph with m layers and n nodes generated under $\mathbb{P}_{m,n}(\cdot|\boldsymbol{P}, \pi_1, \pi_2)$. Assume $\pi_1 \leqslant \pi_2$. Fix a layer set $L \subseteq [m]$. If $\delta(L) > 0$ then there exist constants $A, \eta > 0$ depending on π_1 and $\delta(L)$ such that for all fixed $\varepsilon \in (0, \eta)$,*

$$\mathbb{P}_{m,n}\left(\texttt{Error}\left(\widehat{B}_{opt}(n)\right) < An^\varepsilon \log n\right) \geqslant 1 - \exp\left\{-\frac{\kappa(L)^2 \varepsilon}{32} n^\varepsilon (\log n)^{2-\varepsilon} + \log 4|L|\right\} \tag{4.5}$$

for large enough n.

Note that an immediate corollary of Theorem 11 is that for any $\varepsilon \in (0,1)$,

$$\mathbb{P}_{m,n}\left(\texttt{Error}\left(\widehat{B}_{opt}(n)\right) < n^\varepsilon \log n\right) \to 1 \text{ as } n \to \infty.$$

Therefore, the constants A and η play a role only in bounding the convergence rate of the probability.

The proof of Theorem 11 is given in Section 4.2.3.1. We note that the assumption that $\pi_1 \leqslant \pi_2$ is made without loss of generality, since the community labels are arbitrary. When $m = 1$, Theorem 11 implies asymptotic $n \to \infty$ consistency in the (single-layer) stochastic block model. In this case, the condition that $\delta_\ell = P_\ell(1,1)P_\ell(2,2) - P_\ell(1,2)^2 > 0$ is a natural requirement on the inner community edge density of a block model. This condition appears in a variety of consistency analyses, including the evaluation of modularity (Zhao et al., 2012). When $m > 1$, Theorem 11 implies the vertex set that maximizes $H(B, L)$ will have asymptotically vanishing error with high probability, given that L is a fixed layer set with *all* layers satisfying $\delta_\ell > 0$.

4.2.2.1 Consistency of the joint optimizer

Theorem 11 does not address the *joint* optimizer of the score across all vertex-layer pairs. First, we point out that for a fixed $B \subseteq [n]$, the limiting behavior of the score $\widehat{H}(B, L)$ depends on $L \subseteq [m]$ through the layer-wise determinants $\{\delta_\ell : \ell \in [n]\}$ and the scaling constant $\frac{1}{|L|}$ inherent to $H(B, L)$, as defined in equation (4.3). In what follows, we consider the asymptotic optimality of several alternate scaling constants with respect to all layer-set pairs, and argue that $1/|L|$ is the natural choice of scaling. To this end, let $\gamma : \mathbb{N}^+ \mapsto \mathbb{R}^+$ be an arbitrary non-decreasing function of $|L|$. Define a general version of the multilayer extraction score as

$$H_\gamma(B, L) := \frac{1}{\gamma(|L|)} \left(\sum_{\ell \in L} Q_\ell(B)_+ \right)^2 \tag{4.6}$$

and let $\widehat{H}_\gamma(B, L)$ be the corresponding random version of this score under an MLSBM. We first provide an illustrative example. Consider a MLSBM with $m > 1$ layers having the following structure: the first layer has positive determinant, and all $m-1$ remaining layers have determinant equal to 0. Note that $\delta_1 > 0$ implies that the first layer has ground-truth assortative community structure, and that $\delta_\ell = 0$ for all $\ell > 1$ implies that the remaining layers are (independent) Erdos-Renyi random graphs. In this case, the desired global optimizer of $H_\gamma(B, L)$ is community 1 (or 2) and the first layer. However, setting $\gamma(|L|) \equiv 1$ (effectively ignoring the scaling of H) will ensure that, in fact, the *entire* layer set is optimal, since $Q_\ell(B)_+ \geqslant 0$ by definition. It follows that setting $\gamma(|L|)$ to increase (strictly) in $|L|$, which introduces a penalty on the size of the layer set, is desirable.

For a fixed scaling function γ, define the global joint optimizer of $\widehat{H}(B, L)$ by

$$\left(\widehat{B}_{opt}^{(n)}, \widehat{L}_{opt}^{(n)} \right) := \underset{2^{[n]} \times 2^{[m]}}{\arg\max}\, \widehat{H}_\gamma(B, L) \tag{4.7}$$

Note that $\left(\widehat{B}_{opt}^{(n)}, \widehat{L}_{opt}^{(n)} \right)$ is random, and may contain multiple elements of $2^{[m]} \times 2^{[n]}$. The next theorem addresses the behavior of $\left(\widehat{B}_{opt}^{(n)}, \widehat{L}_{opt}^{(n)} \right)$ under the MLSBM for various choices of $\gamma(|L|)$, and shows that setting $\gamma(|L|) = |L|$ is desirable for consistency.

Theorem 12. *Fix m and let $\{\widehat{\boldsymbol{G}}(m,n)\}_{n>1}$ be a sequence of multilayer stochastic 2 block models where $\widehat{\boldsymbol{G}}(m,n)$ is a random graph with m layers and n nodes generated under $\mathbb{P}_{m,n}(\cdot|\boldsymbol{P},\pi_1,\pi_2)$. Assume $\pi_1 \leqslant \pi_2$. Fix $0 = \delta^{(0)} < \delta^{(1)} < 1$. Suppose the layer set $[m]$ is split according to $[m] = \cup_{i=0,1} L_i$, where $\delta_\ell = \delta^{(i)}$ for all $\ell \in L_i$. Then for any $\varepsilon > 0$, the following hold:*

(a) Suppose $\gamma(|L|) \equiv 1$. Let $\widehat{L}^+ := \{\ell : \widehat{Q}_\ell(\widehat{B}^{(n)}_{opt}) > 0\}$. Then for all $n > 1$, $\widehat{L}^{(n)}_{opt} = \widehat{L}^+$, and

$$\mathbb{P}_{m,n}\left(\texttt{Error}\left(\widehat{B}^{(n)}_{opt}\right) < n^\varepsilon \log n\right) \to 1 \quad as \quad n \to \infty$$

(b) Suppose $\gamma(|L|) = |L|$. Then

$$\mathbb{P}_{m,n}\left(\widehat{L}^{(n)}_{opt} = L_1,\ \texttt{Error}\left(\widehat{B}^{(n)}_{opt}\right) < n^\varepsilon \log n\right) \to 1 \quad as \quad n \to \infty$$

(c) Suppose $\gamma(|L|) = |L|^2$. Then

$$\mathbb{P}_{m,n}\left(\widehat{L}^{(n)}_{opt} \subseteq 2^{L_1},\ \texttt{Error}\left(\widehat{B}^{(n)}_{opt}\right) < n^\varepsilon \log n\right) \to 1 \quad as \quad n \to \infty$$

The proof of Theorem 12 is given in Section 4.2.3.5. Part (a) implies that setting $\gamma(|L|) \equiv 1$ ensures that the optimal layer set will be, simply, all layers with positive modularity, thereby making this an undesirable choice for the function γ. Part (c) says that if $\gamma(|L|) = |L|^2$, the layer set with the highest *average* layer-wise modularity will be optimal (with high probability as $n \to \infty$), which means that all subsets of L_1 are asymptotically equivalent with respect to $\widehat{H}(B,L)$ (with high probability). By part (b), if $\gamma(|L|) = |L|$, then L_1 is the unique asymptotic maximizer of the population score (with high probability). Therefore, $\gamma(|L|) = |L|$ is the most desirable choice of scaling.

4.2.3 Proofs

In this section we prove the theoretical results given in Section 4.2. The majority of the section is devoted to a detailed proof of Theorem 11 and supporting lemmas. This is followed by the proof of Theorem 12, of which we give only a sketch, as many of the results and techniques contributing to the proof of Theorem 11 can be re-used.

4.2.3.1 Proof of Theorem 11, and Supporting Lemmas

We prove Theorem 11 via a number of supporting lemmas. We begin with some notation:

Definition 13. *For a fixed vertex set $B \subseteq [n]$ define*

$$\rho_n(B) = \frac{|B \cap C_{1,n}|}{|B|}, \quad s_n(B) = \frac{|B|}{n}, \quad v_n(B) := (\rho_n(B), 1 - \rho_n(B)) \tag{4.8}$$

We will at times suppress dependence on n and B in the above expressions.

Definition 14. *Define the **population** normalized modularity of a set B in layer ℓ by*

$$\mathcal{Q}_\ell(B) := \frac{s_n(B)}{\sqrt{2}} \left(v_n(B)^t P_\ell v_n(B) - \frac{(v_n(B)^t P_\ell \pi)^2}{\kappa_\ell} \right) \tag{4.9}$$

*Define the **population** score function $\mathcal{H}(\cdot, L) : 2^{[n]} \mapsto \mathbb{R}$ by*

$$\mathcal{H}(B, L) = |L|^{-1} \left(\sum_{\ell \in [L]} \mathcal{Q}_\ell(B) \right)^2 \tag{4.10}$$

Throughout the results in this section, we assume that $L \subseteq [m]$ is a fixed layer set (as in the statement of Theorem 11). We will therefore, at times, suppress the dependence on L from $\delta(L)$ and $\kappa(L)$ (from Definition 10).

4.2.3.2 Sketch of the Proof of Theorem 11

The proof of Theorem 11 is involved and broken into many lemmas. In this section, we give a rough sketch of the argument, as follows. The lemmas in this section establish that:

1. $C_{1,n}$ maximizes the population score $H_*(\cdot, L)$ (Lemmas 15 and 16).
2. For large enough sets $B \subseteq [n]$, the random score $\widehat{H}(B, L)$ is bounded in probability around the population score $H_*(B, L)$ (Lemmas 18 and 21).
3. **Inductive Step**: For fixed $k > 1$, assume that $d_h(\widehat{B}_{opt}(n), C_{1,n})/n = O_p(b_{n,k})$, where larger k makes $b_{n,k}$ of smaller order. Then, based on concentration properties of the score, in fact $d_h(\widehat{B}_{opt}(n), C_{1,n})/n = O_p(b_{n,k+1})$ (Lemma 22).

4. There exists a constant η such that for any $\varepsilon \in (0, \eta)$, $d_h(\widehat{B}_{opt}(n), C_{1,n})/n = O_p(n^{\varepsilon} \log n)$ (Theorem 11). This result is shown using the Inductive Step.

4.2.3.3 Supporting lemmas for the Proof of Theorem 11

Lemma 15. *Define* $\phi(L) := (|L|^{-1} \sum_{\ell} \frac{\det P_{\ell}}{2\kappa_{\ell}})^2$. *Then:*

1. *For any* $B \subseteq [n]$, $q_{\ell}(B) = \frac{s_n(B)}{\sqrt{2}}(\pi_1 - \rho_n(B))^2 \cdot \frac{\det P_{\ell}}{2\kappa_{\ell}}$, *and therefore*

$$\mathcal{H}(B, L) = |L|\phi(L)\frac{s_n(B)^2}{2}(\pi_1 - \rho_n(B))^4$$

2. $\delta(L)^2 \leqslant \phi(L) \leqslant \frac{1}{\pi_1^2 \delta(L)^2}$ *and therefore* $\mathcal{H}(C_{1,n}, L) \geqslant |L|\frac{\pi_1^2}{2}(1 - \pi_1^4)\delta(L)^2$

Lemma 16. *Fix any* $n > 1$. *Define*

$$\mathcal{R}(t) := \begin{cases} \{B \subseteq [n] : |s(B) - \pi_1| \vee [1 - \rho(B)] \leqslant t\}, & \pi_1 < \pi_2 \\ \{B \subseteq [n] : |s(B) - \pi_1| \vee \rho(B) \leqslant [1 - \rho(B)] \leqslant t\}, & \pi_1 = \pi_2 \end{cases}$$

Then there exists a constant $a > 0$ *depending just on* π_1 *such that for sufficiently small* t, $B \notin \mathcal{R}(t)$ *implies* $H_*(B, L) < H_*(C_{1,n}, L) - a|L|\phi(L)t$.

The proofs of Lemmas 15-16 are given in Appendix C. We now give a general concentration inequality for $\widehat{H}(B, L)$, which shows that for sufficiently large sets $B \subseteq [n]$, $\widehat{H}(B, L)$ is close to the population score $H_*(B, L)$ with high probability. This result is used in the proof of Lemma 21, and its proof is given in Appendix C. We first give the following definition:

Definition 17. *For fixed* $\varepsilon > 0$ *and* $n > 1$, *define* $\mathcal{B}_n(\varepsilon) := \{B \subseteq [n] : |B| \geqslant n\varepsilon\}$.

Lemma 18. *Fix* $\varepsilon \in (0, 1)$. *Let* κ *be as in Definition 10. For each* $n > 1$ *suppose a collection of node sets* $\mathcal{B}_n$ *is contained in* $\mathcal{B}_n(\varepsilon)$. *Then for large enough* n,

$$\mathbb{P}_n\left(\sup_{\mathcal{B}_n}\left(\left|\widehat{H}(B, L) - \mathcal{H}(B, L)\right|\right) > \frac{4|L|t}{n^2} + \frac{52|L|}{\kappa n}\right) \leqslant 4|L||\mathcal{B}_n| \exp\left(-\kappa^2 \frac{\varepsilon t^2}{16n^2}\right)$$

for all $t > 0$.

We now define new notation that will serve the remaining lemmas:

Definition 19. *Let $\gamma_n := \log n/n$, and for any integer $k > 0$, define $b_{n,k} := \gamma_n^{1-\frac{1}{2^k}}$.*

Definition 20. *For any $r \in [0,1]$ and $C \subseteq [n]$, define the r-neighborhood of C by $N(C,r) := \{B \subseteq [n] : d_h(B,C)/n \leqslant r\}$. For all $n > 1$, any constant $A > 0$, and fixed $k > 1$, define*

$$N_{n,k}(A) := \begin{cases} N\left(C_1, A \cdot b_{n,k-1}\right) \cup N\left(C_2, A \cdot b_{n,k-1}\right), & k > 1 \\ \mathcal{B}_n(A), & k = 1 \end{cases}$$

Lemma 21, stated below, is a concentration inequality for the random variable $\widehat{H}(B,L)$ on particular neighborhoods of C_1:

Lemma 21. *Fix $\varepsilon \in (0, \pi_1)$ and any constant $A > 0$. For $k > 1$ satisfying $1/2^{k-1} < \varepsilon$, we have for sufficiently large n that*

$$\sup_{B \in N_{n,k}(A)} \left| \widehat{H}(B,L) - \mathcal{H}(B,L) \right| \leqslant 5|L| b_{n,k} \tag{4.11}$$

with probability greater than $1 - 2\exp\{-\frac{\kappa^2 \varepsilon}{32} n \gamma_n^{1-\varepsilon} \log(n) + \log 4|L|\}$. The conclusion holds with $k = 1$ if $A = \varepsilon$.

The proof of Lemma 21 is given in Appendix C. We now state and prove the key lemma used to drive the induction step in the proof of Theorem 11:

Lemma 22. *Fix $\varepsilon \in (0, \pi_1)$ and an integer $k > 1$ satisfying $1/2^{k-1} < \varepsilon$. Suppose there exist constants $A, b > 0$ such that for large enough n,*

$$\mathbb{P}_n\left(\widehat{B}_{opt}(n) \in N_{n,k}(A)\right) \geqslant 1 - b\exp\left\{-\frac{\kappa^2 \varepsilon}{32} n \gamma_n^{1-\varepsilon} \log n + \log 4|L|\right\} := 1 - b\beta_n(\varepsilon)$$

Then there exists a constant $A' > 0$ depending only on π_1 and δ such that for large enough n, $\mathbb{P}_n\left(\widehat{B}_{opt}(n) \in N_{n,k+1}(A')\right) \geqslant 1 - (4+b)\beta_n(\varepsilon)$. The conclusion holds for $k = 1$ if $A = \varepsilon$.

Proof. Assume $\pi_1 < \pi_2$; the following argument may be easily adapted to the case where $\pi_1 = \pi_2$, which we explain at the end. Recall $b_{n,k}$ from Definition 19. For $c > 0$, define

$$\mathcal{R}_{n,k}(c) := \{B \subset [n] : |s(B) - \pi_1| \vee [1 - \rho(B)] \leqslant c \cdot b_{n,k}\},$$

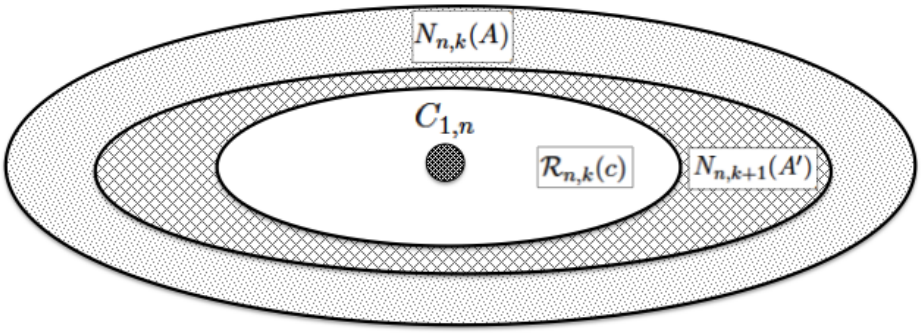

Figure 4.1: Illustration of relationship between collections of node sets.

Note that sets $B \in \mathcal{R}_{n,k}(c)$ have bounded Hamming distance from $C_{1,n}$, as shown by the following derivation. Writing $s = s(B)$ and $\rho = \rho(B)$, for all $B \in \mathcal{R}_{n,k}(c)$ we have

$$n^{-1}|d_h(B, C_{1,n})| = n^{-1}\big(|B \setminus C_{1,n}| + |C_{1,n} \setminus B|\big) = n^{-1}\big(|B| - |B \cap C_{1,n}| + |C_{1,n}| - |B \cap C_{1,n}|\big)$$

$$= s + \pi_1 - 2\rho s \leqslant s + (s + c \cdot b_{n,k}) - 2\,(1 - c \cdot b_{n,k})\, s$$

$$= c \cdot b_{n,k} + 2sc \cdot b_{n,k} \leqslant 3c \cdot b_{n,k} \tag{4.12}$$

Therefore, $\mathcal{R}_{n,k}(c) \subseteq N(C_{1,n}, A' \cdot b_{n,k}) \subset N_{n,k+1}(A')$ with $A' = 3c$.

We have assumed $\widehat{B}_{opt}(n) \in N_{n,k}(A)$ with high probability; our aim is to show $\widehat{B}_{opt}(n) \in N_{n,k+1}(A')$. Since $\mathcal{R}_{n,k}(c) \subseteq N_{n,k+1}(A')$, it is sufficient to show that $\widehat{B}_{opt}(n) \notin N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c$ with high probability. This is illustrated by figure 4.1: since $\widehat{B}_{opt}(n)$ is inside the outer oval (with high probability), it is sufficient to show that it cannot be outside the inner oval. To this end, it is enough to show that, with high probability, $\widehat{H}(B, L) < \widehat{H}(C_{1,n}, L)$ for all sets B in $N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c$. Note that by Lemma 21,

$$\sup_{B \in N_{n,k}(A)} \widehat{H}(B, L) < \mathcal{H}(B, L) + 5|L| b_{n,k} \tag{4.13}$$

for large enough n, with probability at least $1 - 2\beta_n(\varepsilon)$. Next, since $cb_{n,k} \to 0$ as $n \to \infty$, by Lemma 16 there exists a constant $a > 0$ depending just on π_1 such that for large enough n, $B \in \mathcal{R}_{n,k}(c)^c$ implies $\mathcal{H}(B, L) < \mathcal{H}(C_{1,n}) - a|L|\phi(L)cb_{n,k}$. Applying Lemma 21 again, we also have

$\mathcal{H}(C_{1,n}, L) < \widehat{H}(C_{1,n}) + 5|L|b_{n,k}$ with probability at least $1 - 2\beta_n(\varepsilon)$. Furthermore, $\phi(L) \geqslant \delta^2$ by Lemma 15. Applying these inequalities to (4.13), we get

$$\sup_{B \in N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c} \widehat{H}(B, L) < \widehat{H}(C_{1,n}, L) - a|L|\delta^2 c b_{n,k} + 10|L|b_{n,k} \tag{4.14}$$

with probability at least $1 - 4\beta_n(\varepsilon)$. With c large enough, (4.14) implies that $\widehat{H}(B, L) < \widehat{H}(C_{1,n}, L)$ for all $B \in N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c$. This proves the result in the $\pi_1 < \pi_2$ case.

If $\pi_1 = \pi_2$, the argument is almost identical. We instead define $\mathcal{R}_{n,k}(c)$ as

$$\mathcal{R}_{n,k}(c) := \{B \subseteq [n] : |s(B) - \pi_1| \vee \rho(B) \vee [1 - \rho(B)] \leqslant c \cdot b_{n,k}\}.$$

A derivation analogous to that giving inequality (4.12) yields

$$n^{-1} \left(d_h(B, C_{1,n}) \vee d_h(B, C_{2,n})\right) \leqslant 3c \cdot b_{n,k}$$

which directly implies that $\mathcal{R}_{n,k}(c) \subseteq N_{n,k+1}(A')$ with $A' = 3c$. The rest of the argument goes through unaltered. ■

4.2.3.4 Proof of Theorem 11

Recall $Q_\ell(B)$ from Definition 4.2 and let $\widehat{Q}_\ell(B)$ be its random version under the MLSBM, as in Section 4.2.2. For any $B \subseteq [n]$, we have the inequality

$$\left[\widehat{Q}_\ell(B)\right]_+ \leqslant \frac{Y_\ell(B)}{n\binom{|B|}{2}^{1/2}} \leqslant \frac{\binom{|B|}{2}}{n\binom{|B|}{2}^{1/2}} \leqslant \frac{|B|}{n} \tag{4.15}$$

This yields the following inequality for $\widehat{H}(B, L)$:

$$\widehat{H}(B, L) = |L|^{-1} \left[\left(\sum_{\ell \in [L]} Q_\ell(B)\right)_+\right]^2 \leqslant |L|^{-1} \left[\sum_{\ell \in [L]} Q_\ell(B)_+\right]^2 \leqslant |L|^{-1} n^{-2} |B|^2 \tag{4.16}$$

Recall that $\mathcal{B}_n(\varepsilon) := \{B \in 2^{[n]} : |B| \geqslant \varepsilon n\}$. Inequality (4.16) implies $\widehat{H}(B, L) \leqslant |L|\varepsilon^2$ for all $B \in \mathcal{B}_n(\varepsilon)^c$. By part 2 of Lemma 15, $\phi(L) \geqslant \delta^2$. Therefore, defining $\tau := \frac{\pi_1^2}{2}(1-\pi_1)^4\delta^2/2$,

$$|L|\tau < |L|\phi(L)\frac{\pi_1^2}{2}(1-\pi_1)^4 = H_*(B, L)$$

Therefore, for all $B \in \mathcal{B}_n(\varepsilon)^c$, we have $\widehat{H}(B, L) \leqslant |L|\varepsilon^2 < H_*(C_{1,n}, L) - |L|(\tau - \varepsilon^2)$. By Lemma 21, for large enough n we therefore have

$$\sup_{\mathcal{B}_n(\varepsilon)^c} \widehat{H}(B, L) < \widehat{H}(C_{1,n}, L) - |L|(\tau - \varepsilon^2) + 5|L|\gamma_n^{1-\varepsilon} \tag{4.17}$$

with probability greater than $1 - 2\beta_n(\varepsilon)$, where $\beta_n(\varepsilon) := \exp\{-\frac{\kappa^2\varepsilon}{32}n\gamma_n^{1-\varepsilon}\log n + \log 4|L|\}$. For any $\varepsilon < \sqrt{\tau}$, inequality (4.17) implies $\widehat{H}(B, L) < \widehat{H}(C_{1,n}, L)$ for all $B \in \mathcal{B}_n(\varepsilon)$, and therefore $\widehat{B}_{opt}(n) \in \mathcal{B}_n(\varepsilon)$, with probability at least $1 - 2\beta_n(\varepsilon)$. Note that $\varepsilon < \sqrt{\tau} < \pi_1$, and $N_{n,k}(\varepsilon) = \mathcal{B}_n(\varepsilon)$ by Definition 20. Therefore, the conditions for Lemma 22 with $k = 1$ (and $A = \varepsilon$) are satisfied. For any fixed $\varepsilon \in (0, \eta)$ with $\eta := \sqrt{\tau}$, we may now apply Lemma 22 recursively until $1/2^k \leqslant \varepsilon$. This establishes that for sufficiently large n,

$$\mathbb{P}_n\left(\widehat{B}_{opt}(n) \in N_{n,k}(A)\right) \geqslant 1 - (2 + 4k)\beta_n(\varepsilon) \tag{4.18}$$

By definition, $\widehat{B}_{opt}(n) \in N_{n,k}(A)$ implies that

$$\texttt{Error}(\widehat{B}_{opt}(n)) := \min_{C=C_1,C_2} d_h(\widehat{B}_{opt}(n), C) \leqslant A \cdot n \cdot b_{n,k}. \tag{4.19}$$

Note that

$$n \cdot b_{n,k} = n\gamma_n^{1-\frac{1}{2^k}} = n \cdot n^{\frac{1}{2^k}-1}(\log n)^{1-\frac{1}{2^k}} < n^{\varepsilon}\log n$$

since $1/2^k \leqslant \varepsilon$. Combined with inequality (4.18), this completes the proof. ■

4.2.3.5 Proof of Theorem 12

To prove part (a), we first note that Theorem 11 implies that on the layer set L_1, for any $\varepsilon > 0$, $\texttt{Error}(\widehat{B}_{opt}^{(n)}) = O_p(n^{\varepsilon}\log n)$. Lemma 15 can be used to show that $\mathcal{H}(B, L) = 0$ for any $L \subseteq L_0$ and

any $B \subseteq [n]$. Using Lemma 18 and taking a union bound over L_0, it is then straightforward to show (using techniques from the proof of Theorem 11) that on the full layer set $[m]$, for any $\varepsilon > 0$, $\texttt{Error}(\widehat{B}_{opt}) = O_p(n^{\varepsilon} \log n)$. Considering now $\widehat{L}^{(n)}_{opt}$, observe that if $\widehat{Q}_{\ell}(B) \leqslant 0$, then $\widehat{H}(B, L) = \widehat{H}(B, L \setminus \{\ell\})$. This immediately implies that $\widehat{L}^{(n)}_{opt} = \widehat{L}^{+}$.

To prove part (b), we note that it is straightforward to show (using Lemma 15) that $\mathcal{H}(B, L_1) \geqslant \mathcal{H}(B, L)$ for any $L \subset [m]$, with equality if and only if $L = L_1$. Using Lemma 18 and a union bound over $[m]$ will show that $\widehat{L}^{(n)}_{opt} = L_1$ with high probability. Applying Theorem 11 completes the part. Part (c) is shown similarly, with the application of Lemma 15 showing that for any $L \subseteq L_1$ and $L' \subseteq [m]$, $\mathcal{H}(B, L) \geqslant \mathcal{H}(B, L')$, with equality if and only if $L' \subseteq L_1$. ■

4.3 The Multilayer Extraction Procedure

This section introduces a community detection method based on the multilayer set-score called Multilayer Extraction. Importantly, while the method has algorithmic similarity to the SCS procedure for the NST framework (Chapter 2), it does not incorporate explicit hypothesis testing. The method is still testing-based by virture of the set-score's basis in a null-model. Instead of being used for hypothesis testing, however, the set-score will be optimized in a greedy fashion, as described below. Regardless, the method does quite well against competing methods for multilayer un-weighted networks, as shown in the corresponding publication Wilson et al. (2016). As the simulation and application sections for Multilayer Extraction were completed by other authors on this publication, they are omitted from this thesis.

Multilayer Extraction is built around three operations: initialization, extraction, and refinement. In the initialization stage, a family of seed vertex sets is specified. Next an iterative extraction procedure (**Extraction**) is applied to each of the seed sets. **Extraction** alternately updates the layers and vertices in a vertex-layer community in a greedy fashion, improving the score at each iteration, until no further improvement to the score is possible. The family of extracted vertex-layer communities is then reduced using the **Refinement** procedure, which ensures that the final collection of communities contains the extracted community with largest score, and that the pairwise overlap between any pair of communities is at most β, where $\beta \in [0, 1]$ is a user-defined parameter.

The importance and relevance of this parameter is discussed in Section 4.3.3.1. We describe the Multilayer Extraction algorithm in more detail below.

4.3.1 Initialization

For each vertex $u \in [n]$ and layer $\ell \in [m]$ let $N(u, \ell) = \{v \in [n] : \{u, v\} \in E_\ell\}$ be the set of vertices connected to u in G_ℓ. We will refer to $N(u, \ell)$ as the neighborhood of u in layer ℓ. Let $\mathcal{B}_0 = \{N(u, \ell), u \in [n], \ell \in [m]\}$ be the family of all vertex neighborhoods in the observed multilayer network $\mathbf{G}(m, n)$. Multilayer Extraction uses the vertex sets in $\mathcal{B}_0$ as seed sets for identifying communities. Our choice of seed sets is motivated by Gleich and Seshadhri (2012), who showed that vertex neighborhoods are optimal seed sets for local detection methods seeking communities with low conductance.

4.3.2 Extraction

Given an initial vertex set, the **Extraction** procedure seeks a vertex-layer community with large score. The algorithm iteratively conducts a *Layer Set Search* followed by a *Vertex Set Search*, and repeats these steps until a vertex-layer set, whose score is a local maximum, is reached. In each step of the procedure, the score of the candidate community strictly increases, and the procedure is stopped once no improvements to the score are possible. These steps are described next.

Layer Set Search: For a fixed vertex set $B \subseteq [n]$, **Extraction** searches for the layer set L that maximizes $H(B, \cdot)$ using a rank ordering of the layers that depends only on B. In particular let $Q_\ell(B)$ be the local set modularity of layer ℓ from (4.2). Let L_o be the layer set identified in the previous iteration of the algorithm. We will now update the layer set $L_0 \rightsquigarrow L$. This consists of the following three steps:

(i) Order the layers so that $Q_{\ell_1}(B) \geqslant \cdots \geqslant Q_{\ell_m}(B)$.

(ii) Identify the smallest integer k such that $H(B, \{\ell_1, \ldots, \ell_k\}) \geqslant H(B, \{\ell_1, \ldots, \ell_k, \ell_{k+1}\})$. Write $L_p := \{\ell_1, \ldots, \ell_k\}$ for the proposed change in the layer set.

(iii) *If* $H(B, L_p) > H(B, L_o)$ set $L = L_p$. *Else* set $L = L_o$

In the first iteration of the algorithm (where we take $L_o = \emptyset$), we set $L = L_p$ in step (iii) of the search. The selected layer set L_p is a local maximum for the score $H(B, \cdot)$.

Vertex Set Search: Suppose now that we are given a vertex-layer set (B, L). **Extraction** updates B, one vertex at a time, in a greedy fashion, with updates depending on the layer set L and the current vertex set. In detail, for each $u \in [n]$ let

$$\delta_u(B, L) = \begin{cases} H(B/\{u\}, L) - H(B, L) & \text{if } \ u \in B \\ H(B \cup \{u\}, L) - H(B, L) & \text{if } \ u \notin B. \end{cases} \tag{4.20}$$

Vertex Set Search iteratively updates B using the following steps:

(i) Calculate $\delta_u(B, L)$ for all $u \in [n]$. If $\delta_u(B, L) \leqslant 0$ for all $u \in [n]$, then stop. Otherwise, identify $u^* = \arg\max_{u \in [n]} \delta_u(B, L)$.

(ii) If $u^* \in B$, then remove u^* from B. Otherwise, add u^* to B.

At each iteration of **Extraction**, the score of the updated vertex-layer set strictly increases, and the eventual convergence of this procedure to a local maximum is guaranteed as the possible search space is finite. The resulting local maxima is returned as an extracted community.

4.3.3 Refinement

Beginning with the n vertex neighborhoods in each layer of the network, the **Extraction** procedure identifies a collection $\mathcal{C}_T = \{(B_t, L_t)\}_{t \in T}$ of at most $m*n$ vertex-layer communities. Given an overlap parameter $\beta \in [0, 1]$, the family $\mathcal{C}_T$ is refined in a greedy fashion, via the **Refinement** procedure, to produce a subfamily $\mathcal{C}_S$, $S \subseteq T$, of high-scoring vertex-layer sets having the property that the overlap between any pair of sets is at most β.

To quantify overlap, we specify a generalized Jaccard match score to measure overlap between two communities. We measure the overlap between two candidate communities (B_q, L_q) and (B_r, L_r) using a generalized Jaccard match score

$$J(q, r) \ = \ \frac{1}{2} \frac{|B_q \cap C_r|}{|B_q \cup C_r|} + \frac{1}{2} \frac{|L_q \cap L_r|}{|L_q \cup L_r|} \tag{4.21}$$

It is easy to see that $J(q, r)$ is between 0 and 1. Moreover, $J(q, r) = 1$ if and only if $(B_q, L_q) = (B_r, L_r)$ and $J(q, r) = 0$ if and only if (B_q, L_q) and (B_r, L_r) are disjoint. Larger values of $J(\cdot, \cdot)$ indicate more overlap between communities.

In the first step of the procedure, **Refinement** identifies and retains the community (B_s, L_s) in $\mathcal{C}_T$ with the largest score and sets $S = \{s\}$. In the next step, the procedure identifies the community (B_s, L_s) with largest score that satisfies $J(s, s') \leqslant \beta$ for all $s' \in S$. The index s is then added to S. **Refinement** continues expanding S in this way until no further additions to S are possible, namely when for each $s \in T$, there exists an $s' \in S$ such that $J(s, s') > \beta$. The refined collection $C_S = \{B_s, L_s\}_{s \in S}$ is returned.

4.3.3.1 Choice of β

Many existing community detection algorithms have one or more tunable parameters that control the number and size of the communities they identify (von Luxburg, 2007; Leskovec et al., 2008; Mucha et al., 2010; Lancichinetti et al., 2011; Wilson et al., 2014). The family of communities output by Multilayer Extraction depends on the overlap parameter $\beta \in [0, 1]$. In practice, the value of β plays an important role in the structure of the vertex-layer communities. For instance, setting $\beta = 0$ will provide vertex-layer communities that are fully disjoint (no overlap between vertices or layers). On the other hand, when $\beta = 1$ the procedure outputs the full set of extracted communities, many of which may be redundant. In exploratory applications, we recommend investigating the identified communities at multiple values of β, as the structure of communities at different resolutions may provide useful insights about the network itself (see for instance Leskovec et al. (2008) or Mucha et al. (2010)).

Empirically, we observe that the number of communities identified by the Multilayer Extraction procedure is non-decreasing with β, and there is typically a long interval of β values over which the number and identity of communities remains constant. In practice we specify a default value of β by analyzing the number of communities across a grid of β between 0 and 1 in increments of size 0.01. For fixed i, let $\beta_i = (i - 1) * 0.01$ and let $k_i = k(\beta_i)$ denote the number of communities identified at β_i. The default value β' is the smallest β value in the longest stable window, namely

$$\beta' \;=\; \text{smallest } \beta_i \text{ such that } k(\beta_i) = \text{mode}(k_1, \ldots, k_{101})$$

4.4 Discussion

This chapter introduced a significance-based node-set score for multilayer binary networks. It was showed that, under appropriate conditions, the score of true communities in a 2-block SBM is bounded asymptotically close to the global maximizer of the score, with high probability. Then, an implementation of the score in a community detection method for multi-layer binary networks, called Multilayer Extraction, was detailed. Applications of Multilayer Extraction to simulations and real data can be found in Wilson et al. (2016).

CHAPTER 5

Bipartite Correlation Networks

This chapter introduces new methods for Expression-Quantitative Loci (eQTL) analysis. Some background on eQTL analysis was presented in Section 1.4.3. In Section 5.1, a new model for eQTL effect size is introduced. In Section 5.2, preliminary and on-going work is presented regarding a detection method for communities in bipartite eQTL networks.

5.1 The ACME-eQTL model for eQTL effect size

The following sub-sections comprise a thorough motivation, validation, and exploration of a new non-linear regression model called ACME-eQTL for the dependence between a gene and a genomic loci (SNP). Section 5.1.1-5.1.2 introduces some existing work and notation. In Section 5.1.3, the ACME-eQTL model is motivated and stated in detail, followed by statistical tests on real data to show its ability to characterize cis-eQTL associations. In Section 5.1.4, results of analyses are presented showing the robustness of ACME-eQTL model p-values to potential violations of model assumptions in real data. Section 5.1.5 contains a thorough power analysis, comparing ACME-eQTL to existing eQTL models. In Section 5.1.6, the ACME-eQTL model is applied to cis-eQTLs identified using recent data from the GTEx Project, and ACME-eQTL effect sizes are related to other biological features of the data. A summary of these analyses and results is given in Section 5.1.7.

5.1.1 Existing approaches to gene expression modeling

Gene expression data are rarely analyzed on the original scale, due to heteroskedasticity and heavy-tailed errors (Rantalainen et al., 2015). Logarithmic transformation of expression is a standard pre-processing step for many microarray platforms (e.g. Morley et al. (2004); Irizarry et al. (2003)) and often plays an important role in downstream analyses such as differential expression

(e.g. Network et al. (2013); Li et al. (2014)). RNA-Seq data are inherently count-based, and statistical analyses of such data often make use of binomial, negative-binomial, or Poisson generalized linear models (McCarthy et al., 2012; Zhou et al., 2011; Zwiener et al., 2014), all of which use logarithmic or near-logarithmic canonical link functions. However, count-based modeling is rarely used in eQTL analysis, possibly due to computational requirements, and the fact that several stages of read-count normalization that are usually applied to gene expression data. Furthermore, count-based modeling may not be necessary in studies with large sample sizes (Zhou et al., 2011). Instead, eQTL analyses often involve direct application of linear regression to log-transformed expression with additive allelic effects (e.g. Myers et al. (2007)).

Another common transformation of expression is inverse quantile-normalization (e.g. Dixon et al. (2007)), which is used to ensure normality of residuals under the null in the linear regression setting (Beasley et al., 2009; Szymczak et al., 2013). Given a vector y of length n, the quantile-normalization (QN) transformation is the function $Q(y_i) = \Phi^{-1}((rank(y_i)/(n+1))$, mapping each value to a normal quantile corresponding to its rank. Henceforth, eQTL analysis involving linear regression of quantile-normalized gene expression will be referred to as "QN-linear".

The QN-linear model yields p-values that are approximately uniform under the null of no association between expression and genotype. However, the quantile-normalization mapping inherent to this approach erases all connection between the linear model parameters and the original gene expression values. Hence, the estimated coefficients derived from QN-linear regression do not reflect the scale of the original data, and contain almost no information about the true allelic effect. As a result, QN-linear model effect-size estimates from eQTLs with clearly diverse signal-to-noise ratios can yield nearly identical p-values (see Appendix D.5).

5.1.2 Framework and notation

We assume that genotype, gene expression read counts, and covariate data are available for each of n samples. The genotype at a SNP is the number of minor alleles 0, 1, or 2, rounded if using imputed data. Genotype is contained in an $S \times n$ matrix where $S > 0$ is the number of SNP markers; we denote a (length n) row vector of the genotype matrix by s. The expression of a gene is measured by the number of mapped reads relative to the overall library size (see Appendix D.1 for more details). Read count data for expression is contained in a $T \times n$ matrix where $T > 0$ is

the number of genes or transcripts; we denote a (length n) row vector of the expression matrix by c. Finally, measurements of p covariates (like sex or batch) for each sample are stored in a $p \times n$ matrix.

Throughout this chapter, analyses of real data are performed only on "cis" gene-SNP pairs, for which the SNPs are within 1 megabase upstream or downstream of the transcription start or stop sites. The ACME-eQTL model may also easily be applied for trans-analyses, but further consideration of dominance terms may be warranted (see the discussion in Section 5.1.3.1).

5.1.3 The ACME-eQTL model and diagnostics

In this section we introduce the ACME-eQTL model in the context of several alternative approaches to estimating eQTL effect size. We exclude approaches that involves QN-transformed gene expression because, as discussed in Section 5.1.1, our aim is to arrive at a model that retains a direct relationship to the scale of the expression data. We also exclude linear regression on raw gene expression since (as discussed in Section 5.1.1) the error distribution from such a model is known to be heteroskedastic and non-normal. In Appendix B, we show that the non-normality observed in real-data residuals after linear regression with raw expression causes severe Type-I error rates.

In the absence of rank-based normalization, log transformed expression is a natural starting point for regression modeling, and can been shown to yield approximately normal residuals. To illustrate this, we display the results of tests of normality and heteroskedasticity of residuals after fitting QN-linear, standard linear, and various log-linear models (see Appendix D.4) to data from the GTEx pilot project. We see that models using log-transformed expression perform much better than those based on raw expression. Comparison to the QN-linear model results indicate that the log-transformation is still somewhat subject to noise and outliers. However, we consider the resulting violations of normality and homoskedasticity to be acceptably modest, when balanced against the ability to assess effect size with a biologically accurate model. In Section 5.1.4 we provide evidence that p-values for the ACME-eQTL model (which takes log-expression as its response) are approximately uniform under the null.

5.1.3.1 Log ANCOVA and log-linear models

Here we detail the modeling of eQTL association for a single gene-SNP pair (though a full-genome analysis setting involves separate estimations for millions of pairs). Let $y_i := \log(1 + c_i)$ denote the log-transformed normalized gene read count from sample $1 \leqslant i \leqslant n$, where the addition of 1 avoids taking the logarithm of zero. The value c_i is the result of taking the original raw count for the gene in sample i, library-normalizing and then scaling up to the magnitude of the original mean count (see Appendix D.1). Let s_i denote the minor allele count for the SNP in sample i ($s_i \in \{0, 1, 2\}$). Let $\boldsymbol{Z}_i$ denote the $p \times 1$ vector of covariates for sample i, and let γ be an unknown $p \times 1$ covariate coefficient vector. Finally, let $\varepsilon_1, \ldots, \varepsilon_n$ be independent $N(0, \sigma^2)$ errors with positive variance σ^2. Note that the quantities σ and γ may differ across gene-SNP pairs. It is common in eQTL studies to assume that the covariate effect $\mathbf{Z}_i^T\gamma$ contributes to expression on the same scale as the noise (e.g. Shabalin (2012)). We follow this practice for all models considered in this section (including linear regressions with both raw and quantile-normalized gene expression). Additional support for this convention can be seen from the fact that the covariates were computed from normalized data to reduce the influence of outliers (Ardlie et al., 2015), so it is natural for them to be residualized on the log-scale.

One approach to modeling allelic effects is to assume that each genotype is associated with a distinct level of average log-expression. Such a model effectively includes a dominance term for the homozygous genotype for the reference allele, and yields what we refer to as the "log-ANCOVA" model:

$$y_i = \alpha_0 \mathbb{1}_0(s_i) + \alpha_1 \mathbb{1}_1(s_i) + \alpha_2 \mathbb{1}_2(s_i) + \boldsymbol{Z}_i^T\gamma + \varepsilon_i. \tag{5.1}$$

Here the parameters α_j are unknown log-expression means corresponding to the genotypes, and $\mathbb{1}_k(s_i) = 1$ if $s_i = k$, and zero otherwise.

A simpler model is the additive "log-linear" regression model

$$y_i = \theta_0 + \theta_1 s_i + \boldsymbol{Z}_i^T\gamma + \varepsilon_i, \tag{5.2}$$

where θ_0 is baseline log-expression, and θ_1 is the contribution to log-expression of each reference allele. This model includes one fewer degree of freedom than log-ANCOVA, due to the loss of the

dominance term for the secondary reference allele. Here, the mean relationship between alleles and log-expression depends only on allele *count*. Linear regression of transformed expression has been heavily used in eQTL analysis (Ardlie et al., 2015), perhaps partly because evidence of eQTL dominance effects are scant, even in trans-analyses (Wright et al., 2014). However, the log-linear model assumes additivity of allelic effects on the log scale. Despite the widespread use of log-linear regression in the eQTL setting, no one (to our knowledge) has posed a biological motivation for log-scale allelic effect additivity. In fact, this feature runs counter to the simplest model for allele-specific expression, in which allelic effects are additive on the *raw* expression scale, an assumption inherent to the proposed ACME-eQTL model, detailed below.

5.1.3.2 Model statement

The current understanding of cis-eQTL variation in humans is that it is largely allele-specific (Castel et al., 2015), i.e., the transcription of a gene in a particular chromosome is influenced primarily by one or more SNP alleles on the same chromosome. Thus, in the absence of feedback mechanisms, the effect of each SNP allele should be additive on the *original* scale of measured expression. Nevertheless, working with expression on the log-scale is still desirable, based on aforementioned analyses of residual distributions. With these considerations in mind, we propose a log-based non-linear regression model (ACME-eQTL) of the following form:

$$y_i = \log(\beta_0 + \beta_1 s_i) + \boldsymbol{Z}_i^T \gamma + \varepsilon_i. \tag{5.3}$$

Here β_0 is the baseline mean of *raw* expression, and β_1 the additive contribution of each allele. Exponentiating each side of equation 5.3, we may write the model on the expression scale as:

$$c_i + 1 = \left(\beta_0 + \beta_1 s_i\right) \cdot \exp\left(\boldsymbol{Z}_i^T \gamma + \varepsilon_i\right). \tag{5.4}$$

It is clear from this equation that the effect of genotype is linear, as desired, while the effects of noise and covariates are multiplicative. We fit the ACME-eQTL model to data via maximum likelihood, using a Gauss-Newton algorithm, which is derived in Appendix D.6.

The effect size. The coefficients β_1 and β_0 from the ACME-eQTL model operate on the original expression scale, so they lend themselves naturally to a "fold-change" interpretation that is often employed by biologists. In particular, the ratio β_1/β_0 represents the fraction of mean increase due to a single referent allele compared to the baseline genotype 0. We note that Equation (5.3) may be written in the form

$$y_i = \log(\beta_0) + \log\left(1 + \frac{\beta_1}{\beta_0} s_i\right) + \boldsymbol{Z}_i^T \gamma + \varepsilon_i, \tag{5.5}$$

which separates the role of β_0 in determining baseline expression and the role of β_1/β_0 in determining the effect of genotype. Equation 5.5 also plays a role in the fitting algorithm. In what follows we use "effect size" to refer to the ratio β_1/β_0. In Appendix D.7 we obtain a formula for the standard error of β_1/β_0, using a reduced Hessian matrix derived from the model. We note that other notions of effect size may be of interest to biologists. For example, an alternative effect size model, studied simultaneously and independently, incorporates allele additivity through an explicit focus on a fold-change parameter (Mohammadi et al., 2016).

Fitting algorithm and software. Though the ACME-eQTL model can in principle be fit by brute-force likelihood maximization, we found stock implementations of this approach to be, in general, slow and unreliable in practice. We have therefore crafted a custom fitting algorithm for ACME-eQTL and a corresponding software implementation in the `R` statistical computing language. A derivation of our algorithm is provided in Appendix D.6. Our software package is available on the CRAN repository under the name `ACMEeqtl`, installable with the latest `R` release. In Section 5.1.5.1, we provide comparisons of computation time between our software, maximum likelihood, and existing approaches to eQTL analysis.

5.1.3.3 Model fit diagnostics

In the previous section, we pointed out a fundamental mis-match between the log-linear and ACME-eQTL approaches to eQTL effect-size estimation. The log-linear model assumes allelic effect additivity on the log-expression scale, whereas ACME-eQTL assumes additivity on the raw-expression scale. The empirical evaluation of these assumptions can be carried out with goodness-of-fit tests. In this section, we perform these tests using subsamples of real data from the GTEx

Project. The subsampling (described fully in Appendix D.2) was designed to highlight differences between the models by over-representing gene-SNP pairs with strong eQTL evidence, as the log-linear and ACME-eQTL models are indistinguishable under the null model ($\beta_1 = 0$).

Goodness-of-fit test statistic. The log-linear and ACME-eQTL models are each nested within the log-ANCOVA model. For instance, the log-ANCOVA model reduces to the ACME-eQTL model via the parameterization

$$\begin{aligned}
\alpha_0(\beta) &:= \log(\beta_0), \\
\alpha_1(\beta) &:= \log(\beta_0) + \log\left(1 + \frac{\beta_1}{\beta_0}\right), \\
\alpha_2(\beta) &:= \log(\beta_0) + \log\left(1 + 2\frac{\beta_1}{\beta_0}\right).
\end{aligned}$$

Thus, if either model is sufficient to explain variation in gene expression, any further improvements in the log-ANCOVA fit should be small and consistent with the extra degree of freedom in that model. Conversely, a model with poor fit will be (significantly) surpassed by log-ANCOVA.

In testing sets of coefficients in nonlinear regression models with normal errors, F-tests are widely used (Smyth, 2002), and generally better handle the degree of freedom issues posed by numerous covariates than do likelihood ratio tests. Define SSE_3 as the sum of squared residuals from the fit of log-ANCOVA, and SSE_2 as the sum of squared residuals from the fit of any nested model with 2 degrees of freedom (like LL and ACME-eQTL). Then the goodness-of-fit test statistic in our setting is $F = \frac{SSE_2 - SSE_3}{SSE_3/(n-p-3)}$ which is approximately F-distributed with 1 and $n - p - 3$ degrees of freedom (recall that n denotes sample size, while p denotes the number of covariates in the model). A p-value for the goodness-of-fit test is then obtained from the upper-tail of $F_{1,n-p-3}$.

Results. For both the log-linear and ACME-eQTL models, we applied the goodness-of-fit F test to every gene-SNP pair in a sub-sample of pairs from Thyroid tissue data (with $n = 105$ tissue samples), using GTEx pilot data (Ardlie et al., 2015). To judge the distribution of the F-test p-values from each model, we plotted Q-Q plots on the $-\log_{10}$ scale (see Figure 5.1). On each plot, we also supplied the genomic inflation factor λ (hereafter "inflation factor"). The inflation factor is defined by $\lambda := \text{median}_i\{\chi_i^2\}/0.455$, where χ_i^2 is the 1 d.f. chi-squared test statistic corresponding to the p-value for the i-th test, following the original reasoning for genomic control

(Devlin and Roeder, 1999). Figure 5.1 shows that the distribution of F-statistic p-values from the log-linear model grow increasingly non-uniform as eQTLs become more significant, suggesting that the log-linear model is mis-specified. In contrast, F-statistic p-values for the ACME-eQTL model are approximately uniform for eQTLs of all strengths. In other words, the fit of the ACME-eQTL model is largely indistinguishable from that of log-ANCOVA, whereas the fit of the log-linear model is largely insufficient to explain non-null eQTLs. Overall, these results provide strong empirical support for the raw-expression allelic additivity assumption of the ACME-eQTL model, and against the log-expression additivity of the log-linear model. We conclude that, among models nested within log-ANCOVA (which are all models based solely on allelic effect), ACME-eQTL best conforms to the underlying biology of eQTLs.

5.1.4 Model p-values and Type I error

In this section we address violations of residual normality, which can affect Type I error. The large number of tests performed in eQTL analyses presents a special challenge for false positive control. For cis-analysis, the number of tests is typically on the order of 10^7 (Lonsdale et al., 2013), while for trans-analysis, the number of tests can exceed 10^{10}. Using a Bonferroni bound to control family-wise error at 0.05 requires p-values to be accurate at values of 10^{-9} for cis-analyses and at values of roughly 10^{-12} for trans-analyses. In order to perform a test of no effect, i.e.,

$$H_0 : \frac{\beta_1}{\beta_0} = 0 \quad \text{vs.} \quad H_a : \frac{\beta_1}{\beta_0} \neq 0$$

for a given gene-SNP pair, we fit the ACME-eQTL model and the reduced mean-model with $\beta_1 = 0$, and then derive a p-value by comparing the resulting F-statistic with the $F(1, n-p-2)$ distribution. Non-normality in errors can potentially result in non-uniform F-statistic p-values under the null. To assess this, we examined the performance of ACME-eQTL on simulated null data with realistic errors. In addition, we examined the effect of skew in errors for the extremal p-values resulting from large numbers of tests.

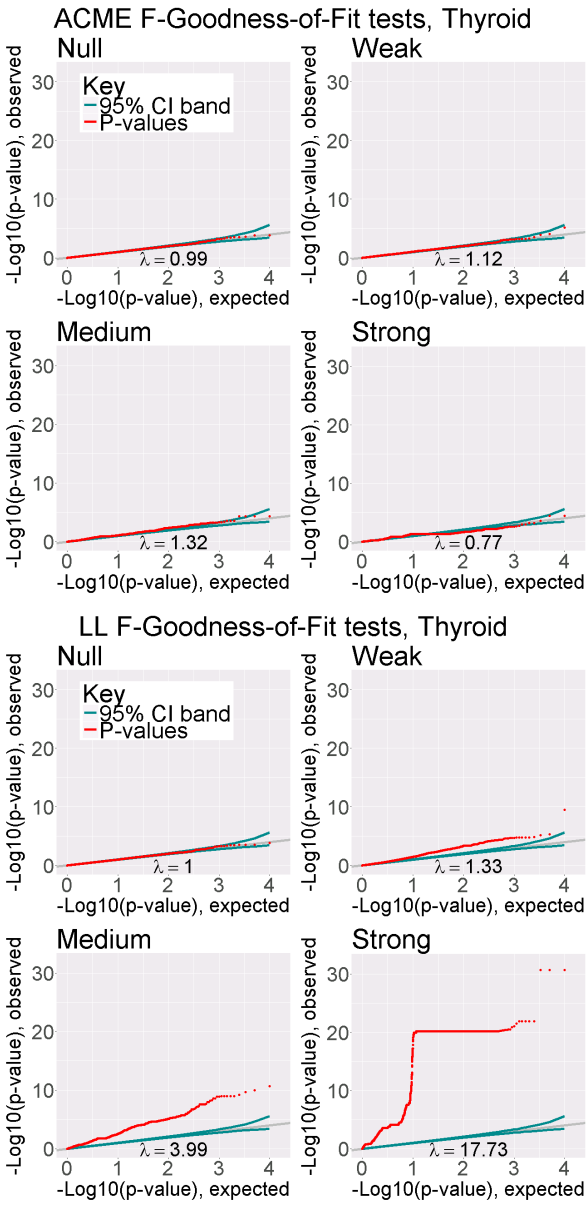

Figure 5.1: Q-Q plots of likelihood ratio test p-values for ACME-eQTL and log-linear models, in each sector of GTEx Thyroid sample data, $n = 105$. The grey line is where we would expect the p-values (represented by the red dots) to fall if they were perfectly uniform, and the green line represents the 95% window of error around this expectation. λ is the estimated genomic inflation factor.

5.1.4.1 Empirical performance of the F test

We began our investigation of the empirical performance of the F test by fitting the ACME-eQTL model to null data simulated with realistic residuals. The residuals for each simulated gene-SNP pair were obtained by re-sampling estimated residuals from ACME-eQTL fits to real GTEx data (full details to be found in Appendix D.3). Both the ACME-eQTL and log-linear

models were fit to 1 million null eQTLs generated in this manner, and p-values were obtained from each method using the F-test. The results are shown in Figure 5.2. The F-test p-values appear nearly uniform for both models, as the inflation factors were exactly 1.00 (corresponding to no inflation). We emphasize that these conclusions address the behavior of the ACME-eQTL fit under a realistic null − the earlier analyses established that ACME-eQTL offers superior fit for real data when evidence of the alternative is strong.

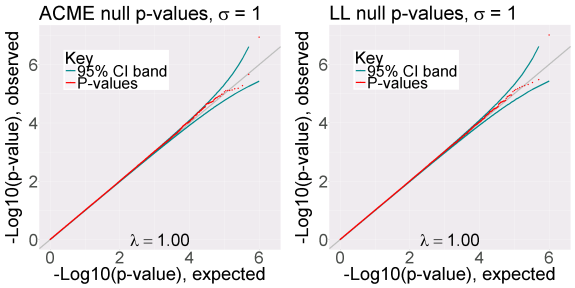

Figure 5.2: p-value distributions from null simulated data with realistic errors and real covariate/genotype data. λ values are inflation factors.

5.1.5 Power, estimation accuracy, and computation speed

In this section we present simulation results which display the detection power, estimation accuracy, and computation speed of the ACME-eQTL model versus existing alternatives. Recall the representation of the ACME-eQTL model from equations 5.5 and D.2, involving the parameter $\eta := \beta_1/\beta_0$. We simulated 100 repetitions of the model at each value of η from an even grid along the range $(-0.5, 10)$. The sample-size was set to $n = 105$, as components of the simulations were taken from real data (as in Section 5.1.4.1). Explicitly, at each repetition, the other components of the model were set as follows:

1. Allele counts from a randomly sampled real-data allele count vector corresponding to GTEx samples of Thyroid tissue.

2. Real-data covariate matrix corresponding to Thyroid samples, constant across all repetitions and values of η.

3. Noise vector ($\varepsilon_{n\times 1}$) and covariate effect ($\gamma_{p\times 1}$) generated as normals with mean 0 and covariances $\sigma^2_\varepsilon I_n$ and $\sigma^2_\gamma I_p$ (respectively), independent within and across repetitions.

We replicated the above simulation framework for various choices of σ_γ, with σ_ε fixed at 1. With simulated data in hand, we applied linear (RAW), quantile-normalized linear (QN), log-linear (LL), log-ANCOVA (ANCOVA), and ACME-eQTL models to each instance of the simulation. For each value of η, we computed the average and standard deviation over the repetitions of the following metrics (per model).

1. F-test p-value for hypotheses $H_0 : \eta = 0$ vs. $H_1 : \eta \neq 0$.
2. Estimated raw expression value when reference allele count equals 1.
3. Estimated raw expression value when reference allele count equals 2.

Note that estimation with the QN model cannot give metrics (2) or (3), as the model is based on a rank-normalized version of expression. Furthermore, the parameters of the LL model are not directly comparable to those of ACME-eQTL or RAW (as they are additive on the log-expression scale), which motivates our choice to evaluate estimated expression rather than estimated β_1.

In Figure 5.3, we display results from the above simulation framework with $\sigma_\varepsilon = \sigma_\gamma = 1$. Our results show that the ACME-eQTL model achieves the most power and estimation accuracy among the alternative methods. Hence, if the ACME-eQTL model is the best representation of underlying eQTL biology in terms of allele count (as the analyses in Section 5.1.3.3 suggest), use of the existing methods reduces accuracy and sensitivity.

5.1.5.1 Computation times

To assess computation speed, we timed various methods on every simulation instance involved with Figure 5.3. There were 10,000 unique values of η, and therefore 1 million simulation instances. On each simulation instance, we recorded the computation time of the following fitting procedures:

1. LL model with least-squares estimation.
2. ACME-eQTL model with maximum likelihood using the BFGS method implemented in the `optim` function from `R`.
3. ACME-eQTL model with the custom fitting algorithm derived in Section D.6.

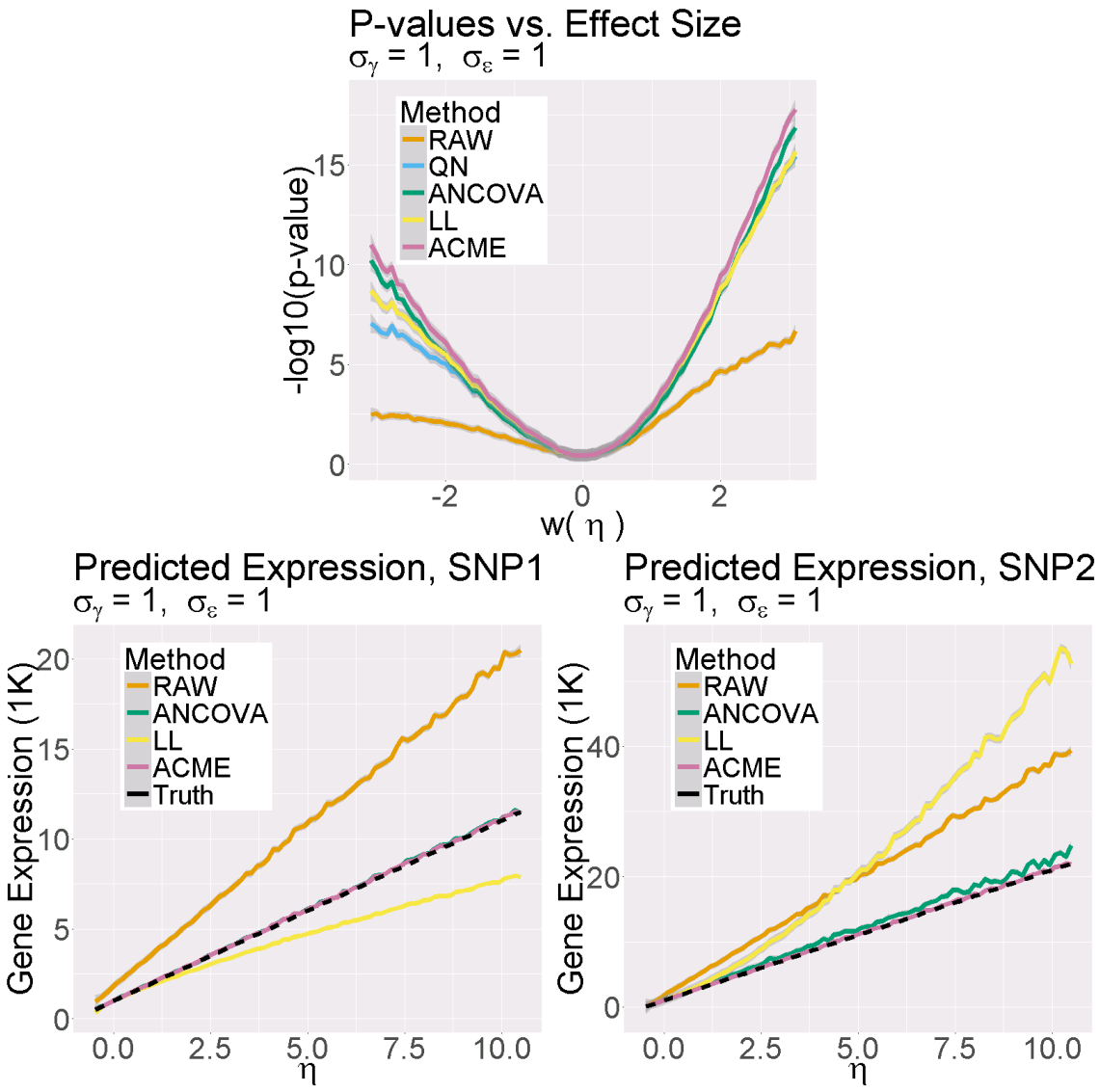

Figure 5.3: Results of large-scale simulation experiment. Left: $-\log 10$ F-test p-values as a function of η. Middle and right: predicated raw expression with one and two reference alleles, respectively.

The timing of procedure (1) will be of the same order as any other procedure based on least-squares (RAW, QN, and ANCOVA). The timing of procedure (2) is provided as a benchmark for procedure (3). All computations were performed on an Intel Xeon E5-2640 (2.50 GHz), using the `R` software language, and timed with the `microbenchmark` package.

The mean and standard deviation of computation times for the procedures, over the 1 million simulation instances, were as follows: least-squares at 0.129ms (0.233), BFGS at 2.687ms (1.270), custom ACME-eQTL at 0.470ms (0.285). So, the efficiency of the custom ACME-eQTL algorithm is quite comparable to that of least-squares estimating equations, and far outstrips stock optimization

methods. The complete package implementing our algorithm, including wrappers that employ parallelization to process full-genome results from massive-scale eQTL data, is available in the `ACMEeqtl` package on the CRAN respository.

5.1.6 Large-scale real data analysis

Though it is not the primary purpose of this section to provide novel real data analyses, in this section we show basic features of genome-wide cis-eQTL patterns, using ACME-eQTL model estimates from GTEx Pilot data. We calculated estimated effect sizes and F-based p-values for cis-acting gene-SNP pairs from the nine tissues described in Ardlie et al. (2015), having sample sizes between $n = 83$ and $n = 156$. Each tissue-specific data set had $p = 19$ covariates: sex and 3 genotype principal components, which were shared across all tissues; and 15 PEER (Probabalistic Estimation of Expression Residuals) factors computed from expression data (Stegle et al., 2012). These covariates are described in Ardlie et al. (2015).

Figure 5.4 shows the results of the full-tissue cis-eQTL ACME-eQTL analyses from Thyroid tissue. The left panel indicates that estimated effect sizes are larger when the average expression level is smaller. This aligns with the understanding that minor alleles will have smaller effect relative to baseline expression when the baseline is larger. The center panel shows that eQTLs tend to be more significant when the distance between the gene transcription start site (TSS) and SNP is smaller. This shows that ACME-eQTL effect sizes are consistent with established properties of cis-eQTLs (Ardlie et al., 2015). The right panel in Figure 5.4 shows a direct comparison of ACME-eQTL F p-values and p-values from the QN-linear model. We see that, although the results are correlated, p-values obtained from the two methods can differ by a few orders of magnitude. Furthermore, for the most significant pairs, ACME-eQTL p-values tend to be lower. This reflects the increased power of the ACME-eQTL model, discussed in Section 5.1.5. Overall, the right panel in Figure 5.4 suggests that using the QN-linear model for cis-eQTL analysis can yield inaccurate p-values.

Computation times for the real-data analyses were recorded for both the ACME-eQTL and QN model fitting procedures. The `R` packages `MatrixEQTL` and `ACMEeqtl` provide full-tissue cis-eQTL procedures for the QN and ACME-eQTL models (respectively). We timed the run of each procedure on each of the nine tissues. The results are shown in the bottom-right of Figure 5.4. We

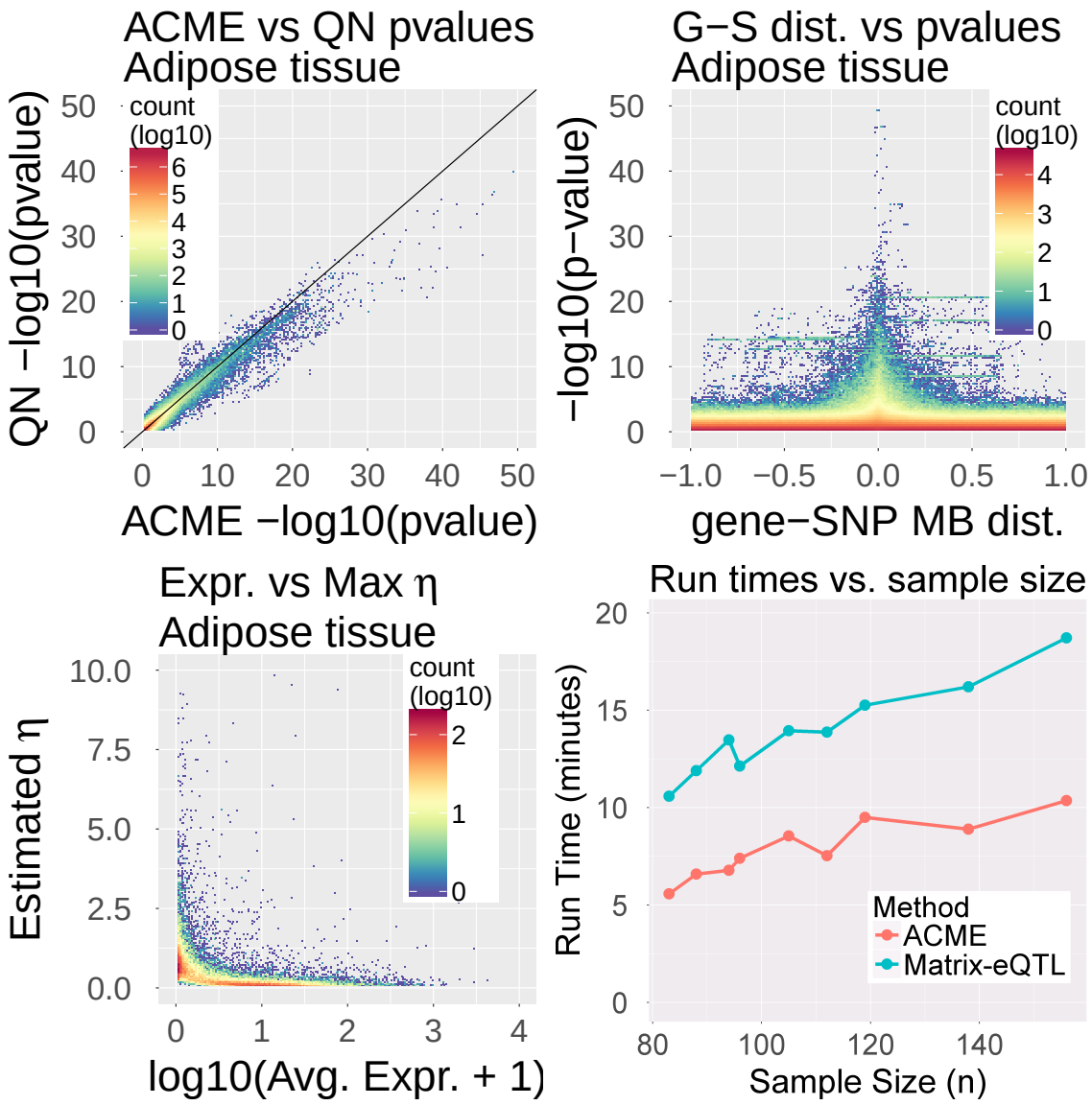

Figure 5.4: Results of genome-wide cis-eQTL ACME effect size estimations on Thyroid tissue ($n = 105$), from GTEx Pilot data. Bottom-left: Maximum gene-wise estimated effect size vs. log average expression level. Top-left: F p-values from the QN-linear model vs. F p-values from the ACME-eQTL model. Top-right: $-\log_{10}$ ACME-eQTL p-value vs. distance from gene TSS to SNP position. Bottom-right: Full-tissue procedure times of the Matrix-EQTL and ACME-eQTL fitting softwares.

note that the ACME-eQTL software benefits from parallelization that the Matrix-EQTL software currently does not employ. This explains why the ACME-eQTL full-tissue procedure is faster than Matrix-EQTL, even though the former based on an iterative optimization procedure.

5.1.7 Discussion

In this section we have proposed ACME-eQTL, a new model for the effect-size of cis-eQTLs between individual genes and SNPs. ACME-eQTL is based on a simple biological model for cis-eQTL action, and supported by careful analysis of real data. Our analyses reveal that existing approaches to cis-eQTL analysis, while useful for some purposes, do not reflect the most plausible relationship between allele count and gene expression. Goodness-of-fit tests using real data show that the systematic component of the ACME-eQTL model is the best-fitting description of cis-eQTL action among two-parameter models involving allele count alone. Thus, the ACME-eQTL model yields reliable cis-eQTL effect sizes that have a principled biological interpretation. We also show that the F-test for the significance of estimated effects derived from the model is robust to the types of violations of error assumptions encountered in real data. Finally, we showed that when real data closely follows the ACME-eQTL model (as the goodness-of-fit tests suggest), standard methods for effect size estimation provide insufficient power and estimation accuracy. We provided single-eQTL and full-tissue comparisons of computation times, showing that large-scale eQTL analyses with the ACME-eQTL model are extremely feasible. The `R` code for the ACME-eQTL model fitting algorithm is available in the `ACMEeqtl` package on the CRAN repository.

Though we did not explore the application of the ACME-eQTL model to trans-eQTL analysis, the ACME-eQTL model can in principle be applied to trans-eQTLs. However, for trans-eQTLs, dominance effects may be more plausible, while current sample sizes may be inadequate to fully investigate such effects. Thus we consider the use of the ACME-eQTL model for trans-eQTLs to be exploratory.

Another area of extension that we did not thoroughly explore is a multi-SNP ACME-eQTL model. However, it is not obvious that allelic additivity assumption should hold in a multi-SNP model. This is certainly an area for further investigation. For any proposed multi-SNP ACME-eQTL model, goodness-of-fit tests of the type presented in Section 2.3 would be one avenue for verification of biological accuracy, as the logical extension of allele-specific expression to multiple SNPs is to adopt a multiple *haplotype* model. To facilitate work on this project, we have implemented a step-wise fitting algorithm for a multi-SNP ACME-eQTL model, with software code included in the ACME-eQTL software package. These estimates can be used to consider preliminary

ACME-eQTL models that are additive in genotypes across SNPs, until more rigorous verification of a multi-SNP approach can be performed.

The primary motivation of our work was to place inferences for cis-eQTL effect sizes on a solid statistical foundation. We believe the ACME-eQTL model provides such a foundation, and can be readily implemented in current eQTL analyses. The results may be useful for investigations in which interpretable eQTL effect sizes are relevant, such as examining enrichment and overlap with genome-wide association studies (Zhu et al., 2016).

5.2 Future work: Bi-community detection for correlation networks

This section includes preliminary development and analysis of an NST method for bi-partite correlation networks called Correlation Bi-Community Extraction (CBCE). The CBCE method was originally motivated by eQTL data, as the set of genes and the set of genomic loci can be organized into two halves of a bi-partite correlation network. As discussed in Section 1.4.3, a community in a bi-partite network is a *pair* of node sets, one set from each side of the network, such that nodes from one set are strongly connected with nodes from the other. In this section, these pairs are termed *bi*-communities.

In the eQTL setting, a "true" bi-community may have some functional relationship with at least one phenotype of interest. Furthermore, as bi-communities contain many eQTLs, their relationships with phenotypes and GWAS results may be more complex and revealing than those of individual eQTLs. The network macro-structure of discovered bi-communities in eQTL networks may also have interesting interpretations in terms of phenotypes. For instance, in study of eQTL networks in lung expression data, Platig et al. (2015) found that "hub" genomic loci (loci with many significant eQTL connections) were *negatively* correlated with known diseases.

5.2.1 The NST Framework for Bi-partite Networks

Before introducing our approach to bi-partite correlation networks, we adapt the NST framework to bi-partite networks in general. A general complex bi-partite network with $[n]$ nodes can be written $\mathcal{G} = (N_1, N_2, \mathbf{D})$, with $N_1 \cup N_2 = [n]$, $N_1 \cap N_2 = \phi$, and $\mathbf{D}$ an arbitrary data object corresponding to the node set. Community detection for bi-partite networks involves the

nuance that associations revealed by $\mathbf{D}$ are judged *across* the network. That is, a bi-community $(C_1, C_2) \in 2^{N_1} \times 2^{N_2}$ is called "strongly connected" by virtue of high association *between* C_1 and C_2, but not *within* each set. We now give an NST notion of association in a bi-partite network, which naturally extends Definition 1 in Chapter 2:

Definition 23. *Given $a : [n] \times 2^{[n]} \mapsto \mathbb{R}$, a bi-community $(C_1, C_2) \subseteq 2^{N_1} \times 2^{N_2}$ satisfies*

(i) $a(u, C_2), a(v, C_1) > 0$ *for each* $u \in C_1$ *and* $v \in C_2$, *and*

(ii) $a(u, C_2), a(v, C_1) \leqslant 0$ *for each* $u \in C_1$ *and* $v \in C_2$.

As in the NST framework from Chapter 2, assume that we can specify a test statistic $T(u, B, \mathbf{D})$ for $a(u, B)$, and corresponding p-value $p(u, B, \mathbf{D})$. We now provide a natural modification to the NST update (see Section 2.3) to update a bi-community:

Core update $U_\alpha : 2^{N_1} \times 2^{N_2} \mapsto 2^{N_1} \times 2^{N_2}$

Given bi-partite network $\mathcal{G} = (N_1, N_2, \mathbf{D})$ and input bi-set $(B_1 \times B_2) \subseteq N_1 \times N_2$:

1. Calculate p-values $\mathbf{p} := \{p(u, B_2, \mathbf{D}) : u \in N_1\} \cup \{p(v, B_1, \mathbf{D}) : v \in N_2\}$.

2. Obtain threshold $\tau(\mathbf{p})$ from a multiple-testing procedure.

3. Return (B_1', B_2') with $B_1' := \{u : p(u, B_2, \mathbf{D}) \leqslant \tau(\mathbf{p})\}$ and $B_2' := \{v : p(v, B_1, \mathbf{D}) \leqslant \tau(\mathbf{p})\}$.

Note that the union of a bi-set $B_1 \cup B_2$ is just a subset of the full node set n. Hence, assuming that the set of p-values $\mathbf{p}$ are independent and uniformly distributed under an appropriate global null for $\mathbf{D}$, Theorem 4 guarantees that stable bi-communities occur with probability at most α under the null. The remaining components of the NST framework, in particular the Stable Community Search (SCS) algorithm detailed in Section 2.3, require no modifications for bi-partite networks. The rest of this section is devoted to specification of the association function, test statistic, null model, and p-values for bi-partite *correlation* networks. These will provide the core features of the CBCE method.

5.2.2 Notation and CBCE Framework

Let $\mathbf{X}$ be an $m \times n$ real-valued matrix, with rows representing samples and columns representing nodes. Throughout, we will assume that the columns $\mathbf{X}$ are centered at 0 and scaled to have unit variance. Each column of $\mathbf{X}$ corresponds to a node in a bi-partite network of interest with nodes $[n]$. For any node set $B \subseteq [n]$, let $\mathbf{X}_B$ denote the B-column-subset of $\mathbf{X}$. For $u \in [n]$, let $\mathbf{X}_u$ be the u-th column. A bi-partite correlation network has a concise representation as $\mathcal{G} := (N_1, N_2, \mathbf{X})$, as all sample correlations may be computed from $\mathbf{X}$.

In applications, $\mathbf{X}_u$ is the sample data (of dimension m) corresponding to the node u. Often, there will be a natural division of the nodes $[n]$ into disjoint and exhaustive sets $N_1 \cup N_2 = [n]$, a division which has some relation to the scientific setting of interest. In the eQTL setting, for instance, the bipartite division of interest is between SNPs and genes. The data matrix $\mathbf{X}$ is often best represented as a block matrix (X_1, X_2) of sub-dimensions $m \times |N_1|$ and $m \times |N_2|$. In practice, the data X_1 may have a much different character than X_2. In the eQTL setting, SNP data is binary, and gene data is continuous. This type of heterogeneity is fully allowed in the NST approach outlined in the following section.

5.2.3 SCS test statistic and p-value for bi-partite correlation networks

Assume that $\mathbf{X}$ has a true underlying correlation structure defined by the function $\rho : [n] \times [n] \mapsto [-1, 1]$. Arguably the most straightforward notion of node-to-set association in a correlation network is the true *average* correlation between the node and set. The CBCE method will be based on the average correlation association, explicitly defined as

$$a(u, B) := |B|^{-1} \sum_{v \in B} \rho(u, v). \tag{5.6}$$

This association function is identical to that presented in one of the illustrative examples of the NST framework, for standard correlation networks. Note that $a(u, B) = 0$ is the natural null assumption in the correlation network setting, as it implies zero average correlation between u and the nodes in B. Also, though $a(u, B)$ is defined for *any* node $u \in [n]$ and node-set $B \subseteq [n]$, in the

bi-partite community detection setting, it is only of interest to estimate $a(u, B)$ when u and B are from separate sides of the network, as in the modified SCS algorithm defined above.

For any nodes u and v, denote the sample correlation by $r(u, v) := (n-1)^{-1}\mathbf{X}_u^t\mathbf{X}_v$. Our NST test statistic for $a(u, B)$ is then the average sample correlation, which we write as

$$R(u, B, \mathbf{X}) := |B|^{-1} \sum_{v \in B} r(u, v). \tag{5.7}$$

For any given $u \in [n]$ and $B \subseteq [n]$, we use $R(u, B, \mathbf{X})$ to perform the hypothesis test

$$H_0 : a(u, B) = 0 \quad \text{vs.} \quad H_a : a(u, B) > 0. \tag{5.8}$$

Let $\mathbb{P}_\theta$ denote the joint distribution of the sample correlations $\{r(u, v) : u, v \in [n]\}$, where θ is a parameter that will depend on the distribution of $\mathbf{X}$. Let $\widetilde{\mathbf{X}}$ be a random data set from the distribution of $\mathbf{X}$. Then, assuming θ is specified so that $a(u, B) = 0$, we can quantify the evidence against the null hypothesis in (5.8) with the p-value

$$p(u, B, \mathbf{X}) = \mathbb{P}_\theta(R(u, B, \widetilde{\mathbf{X}}) > R(u, B, \mathbf{X})) \tag{5.9}$$

However, $\mathbb{P}_\theta$ is not known to have a closed-form. Even if a tractable form of $\mathbb{P}_\theta$ were available, it is unlikely the distribution of $R(u, B, \widetilde{\mathbf{X}})$ would be analytical. To approximate $\mathbb{P}_\theta$, Steiger and Hakstian (1982) provided a limiting normal distribution for the joint distribution of sample correlations, under quite general conditions. Explicitly, they established the following theorem:

Theorem 24 (Steiger and Hakstian (1982))**.** *Let $\{X_1, \ldots, X_n\}$ be a random vector with distribution $\mathbb{P}$ satisfying $\mathbb{E}|X_u|^4 < \infty$ for all $u \in [n]$. Let $\mathbf{X}$ be an $m \times n$ matrix, with rows containing independent samples from $\mathbb{P}$. Let $\mathbf{R}$ be a n^2-dimensional vector containing the (vectorized) entries of the sample correlation matrix of $\mathbf{X}$. Then there exists $\theta = (\rho, \Sigma)$ depending on $\mathbb{P}$ such that*

$$m^{1/2}(\mathbf{R} - \rho) \Rightarrow \mathcal{N}_{n^2}(\mathbf{0}_{n^2}, \Sigma)$$

In the theorem, ρ is of course the true variable correlations under $\mathbb{P}$. The entries of Σ depend on the fourth-order cross-moments. Explicitly, recall that the columns of $\mathbf{X}$ are assumed to be

centered and scaled, and define for $u, v, w, x \in [n]$ the following cross-moments:

$$\rho_{uv} := \mathbb{E}X_u X_v \quad \text{and} \quad \rho_{uvwx} := \mathbb{E}X_u X_v X_w X_x, \tag{5.10}$$

Steiger and Hakstian showed that a general element of Σ has the closed form

$$\Sigma_{uv,wx} := \rho_{uvwx} + \tfrac{1}{4}\rho_{uv}\rho_{wx}(\rho_{uuww} + \rho_{vvww} + \rho_{uuxx} + \rho_{vvxx})$$

$$- \tfrac{1}{2}\rho_{uv}(\rho_{uuwx} + \rho_{vvwx}) - \tfrac{1}{2}\rho_{wx}(\rho_{uvww} + \rho_{uvxx})$$

Hence, to compute the p-value $p(u, B, \mathbf{X})$ given in (5.9), we take $\mathbb{P}_\theta$ to be the limiting multivariate normal distribution given in Theorem 24, assuming that $a(u, B) = 0$ under $\mathbb{P}_\theta$. As the elements of Σ are unknown, plug-in sample moments from $\mathbf{X}$ are used in place of the expressions in (5.10). This approach is analogous to that in other implementations of the NST framework, as discussed in Section 2.2.

5.2.4 The CBCE method

The overall extraction method for bi-partite correlation networks, called Correlation Bi-Community Extraction (CBCE), is now defined. The CBCE method incorporates the Node-Set Testing SCS algorithm based on the correlation sum statistic $R(u, B, \mathbf{X})$, described above, in the same way that the CCME method incorporates the SCS algorithm based on $S(u, B, \mathcal{G})$, described in Section 3.4. In particular, stable cycles are dealt with as described in Section A.1, and overlapping stable communities are filtered based on the process described in Section B.2. However, the initialization procedure used for CBCE is unique to the setting of bi-partite correlation networks. The CBCE method obtains an initial bi-community with the following algorithm:

CBCE Initialization

Given a bi-partite correlation network $\mathcal{G} = (N_1, N_2, \mathbf{X})$ and initial node $u \in N_1$:

1. Calculate correlations $\{r(u, v) : v \in N_2\}$.

2. Calculate p-values $\mathbf{p}_2 := \{p(u, v) : v \in N_2\}$ based on correlation t-tests.

3. Obtain multiple-testing threshold $\tau(\mathbf{p}_2)$ and set $B_2 \leftarrow \{v \in N_2 : p(u, v) \leqslant \tau(\mathbf{p}_2)\}$.

4. Calculate p-values $\mathbf{p}_1 := \{p(u, B_2, \mathbf{X}) : u \in N_1\}$ from Equation 5.9.

5. Obtain multiple-testing threshold $\tau(\mathbf{p}_1)$ and set $B_1 \leftarrow \{u \in N_1 : p(u, B_2, \mathbf{X}) \leqslant \tau(\mathbf{p}_{0,1})\}$.

6. Return (B_1, B_2).

The CBCE initialization algorithm is essentially a bisected version of the bi-partite SCS algorithm. Though it is defined above for $u \in N_1$, it can be applied symmetrically to nodes $v \in N_2$, with reversed ordering of test-statistic computations. The full CBCE method can now be defined:

The CBCE Community Detection Method for Bipartite Correlation Networks

Given an observed bipartite correlation network $\mathcal{G}$:

1. Obtain initial bi-communities $\mathcal{B}_0 := \{(B_1, B_2)_u : u \in [n]\}$ via CBCE Initialization

2. Apply bi-partite SCS to each bi-community in $\mathcal{B}_0$, resulting in stable communities $\mathcal{C}$.

3. Remove sets from $\mathcal{C}$ that are empty or redundant.

We analyze the performance of CBCE on simulated and real data in the following sections.

5.2.5 Simulation results

This section contains a performance analysis of CBCE and some existing methods on a benchmarking simulation framework. Simulated networks are generated from a model that plants small bi-communities of varying size among many background nodes. Our model, choice of performance measures, competing methods, simulation settings, and results are described in the subsections below.

5.2.5.1 Simulation model

The simulation model we employ is meant to produce realistic bipartite correlation networks with heterogeneous bi-communities. Before describing the model, we first give a complete list of parameters. After, we describe each parameter and its role in the simulation model.

Table 5.1: Simulation model parameters

m: Dimension of node-wise Euclidean vectors	b: Number of bi-communities
$c_{\max}$: Max community size	$c_{\min}$: Min community size
g: Scaling factor for background node set	μ_β: Mean of regression parameters
p: Within-bi-community eQTL probability	ρ^*: Intra-correlation of N_1 nodes
σ^2: Regression noise variance	

Our simulation model produces a bi-partite correlation network $\mathcal{G} = (N_1, N_2, \mathbf{X})$ where $\mathbf{X}$ is a block-matrix (X_1, X_2), as described in 5.2.2. We now describe the "default" network model of our simulation framework, defined by pre-set values of the parameters above. In simulation settings (to be described later), we toggle parameters to assess their effect on the competing methods' accuracies. In the default network, X_1 and X_2 have row dimension $m = 200$. The node sets N_1 and N_2 are first populated by $b = 10$ disjoint bi-communities (C_1, C_2) satisfying $c_{\min} \leqslant |C_1|, |C_2| \leqslant c_{\max}$. The sizes of each half of each bi-community are chosen uniformly at random from $[c_{\min}, c_{\max}]$. The expected number of nodes in a bi-community from either N_1 or N_2 is thus $b \cdot (c_{\min} - c_{\max})/2$. Using now the parameter $g > 0$, we give $\lceil g \cdot b \cdot (c_{\min} - c_{\max})/2 \rceil$ background nodes to both N_1 and N_2. Thus, it can be said that the simulation model produces g "times" more background nodes than community nodes, on average. In the default model, $g = 1$.

So far, we have described the determination of $K = 10$ ground-truth bi-communities $\mathcal{C}^* := \{(C_{1,i}, C_{2,i}) \ : \ i \in [K], \ C_{1,i} \subseteq N_1, \ C_{2,i} \subseteq N_2\}$, and background node sets B_1 and B_2. We now describe the simulation of X_1 and X_2 based on this bi-community structure. Cross-correlations between X_1 and X_2 are generated via a regression model with censored coefficients. First, we generate the columns of X_1 as Normal random m-vectors with mean zero. For $u \neq v \in N_1$, let $\rho(u, v)$ denote the correlation between columns u and v from X_1. Then X_1 is generated with the

following correlation structure, depending on $\mathcal{C}^*$ and B_1:

$$\rho(u,v) = \begin{cases} \rho^*, & \text{if } u \text{ and } v \text{ are in the same bi-community half} \\ 0, & \text{if } u \text{ and } v \text{ are in different bi-community halves, or if } u, v \in B_1 \end{cases}$$

In the default model, ρ^* is set to 0.3. With X_1 in hand, the columns of X_2 are generated by the following process. For each $u \in N_1$ and $v \in N_2$, let $\delta(u,v)$ be equal to 1 if u and v are in a true bi-community together, and 0 otherwise. Let δ be the $|N_1| \times |N_2|$ matrix of $\delta(u,v)$'s. Let $\mathbf{A}$ be a random $|N_1| \times |N_2|$ matrix of independent Bernoulli random variables with success probability p. Let $\mathbf{E}$ be a random $|N_1| \times |N_2|$ matrix of independent exponential random variables with mean μ_β. Let $\mathbf{Z}$ be a random $m \times |N_2|$ matrix of $\mathcal{N}(0, \sigma^2 \cdot c)$ random variables. The constant c will be described in the paragraph below. Let $\circ$ denote the entry-wise (Hadamard) matrix product. Then X_2 is generated by the regression equation

$$X_2 = X_1^t \ (\delta \circ \mathbf{A} \circ \mathbf{E}) + \mathbf{Z} \tag{5.11}$$

Essentially, the above equation says that each node from N_2 chooses a p proportion of nodes (via $\mathbf{A}$) within its bi-community to be cross-neighbors, and then is generated by a weighted linear combination (via $\mathbf{E}$) of those neighbors, plus some noise (via $\mathbf{Z}$). The default parameter settings are $p = 0.5$, $\mu_\beta = 1$, and $\sigma^2 = 4$. The scaling constant c in the variance of $\mathbf{Z}$ approximates the variance of X_2 due to the regression on X_1, so that σ^2 controls the noise variance *relative* to the signal. The sample correlation matrix of a draw from the default model is shown in Figure 5.5. We can see ten clearly-defined bi-communities of varying size, and large background node sets approximately twice the size of the set of nodes belonging to bi-communities.

5.2.5.2 Performance measures

To assess the performance of a bi-community detection method, three measures are used:

1. **Best Match (BM):** Introduced by Goldberg et al. (2010), the Best Match (BM) performance measure captures the similarity between two collections of index sets. Explicitly, for integers $K_1, K_2 > 0$, let $\mathcal{C}_1$ be a K_1-collection $\{S_i : i \in [K_1]\}$ and $\mathcal{C}_2$ be a K_2-collection $\{S'_j : j \in [K_2]\}$.

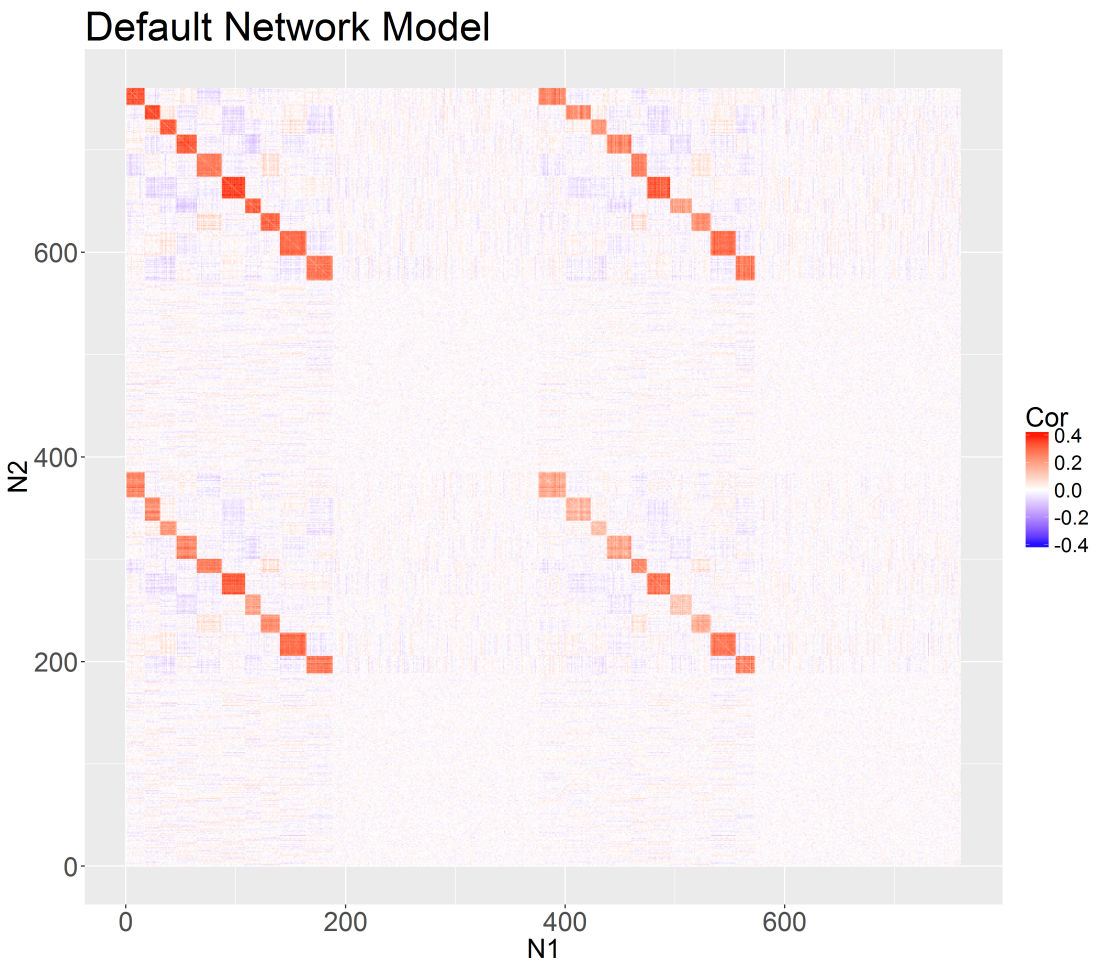

Figure 5.5: Correlation matrix of $\mathbf{X}$ from a draw from the default simulation model.

Let $s(S_i, S'_j)$ be an arbitrary similarity function for index sets. The best match score is then defined as

$$\mathrm{BM}(\mathcal{C}_1, \mathcal{C}_2) := \frac{\sum_{S \in \mathcal{C}_1} \max_{S \in \mathcal{C}_2} s(S, S') + \sum_{S' \in \mathcal{C}_2} \max_{S \in \mathcal{C}_1} s(S', S)}{K_1 + K_2}$$

Note that BM is symmetric even if s is not. Throughout, we compute BM via the Jaccard similarity defined $s(S, S') := |S \cap S'|/|S \cup S'|$.

2. **Background Jaccard (BJ):** The Jaccard similarity between the set of background nodes found by the method and the set of true background nodes.

3. **Runtime:** The computation time of the method, in seconds.

5.2.5.3 Competing methods

In addition to CBCE, we run two other bi-partite community detection approaches through our simulation framework. Given a bi-partite correlation network $\mathcal{G} := (N_1, N_2, \mathbf{D})$, each method depends on the computation of the cross-correlation matrix $\mathbf{C} := \mathbf{X}^t\mathbf{Y}$.

1. **Bipartite Recursively-Induced Modules (BRIM):** Barber (2007) extended the modularity score (see Section 1.2.3) to bi-partite networks, and introduced the BRIM method for community detection on bipartite networks as a technique for local maximization of the bi-partite modularity. As BRIM operates only on *binary* bi-partite networks, we introduce the following procedure to adapt the method in our setting:

   (i) Convert the entries of $\mathbf{C}$ to t-statistic p-values.

   (ii) Compute the Benjamini-Hochberg threshold $\tau = \tau(\mathbf{C})$ at level $\alpha = 0.1$.

   (iii) Dichotomize the entries of $\mathbf{C}$ into $0-1$ variables, where an entry becomes 1 if and only if it is less than τ.

   (iv) Apply BRIM to the binary bipartite network $\mathcal{G}' := (N_1, N_2, \mathbf{C})$.

   Note that after step 3, some nodes may have no cross-edges. We remove these nodes from $\mathcal{G}'$ and automatically assign them to background.

2. **Independent Row-Column k-means:** One potential solution to bi-community detection in correlation networks is to apply bi-clustering to $\mathbf{C}$. A common approach to bi-clustering is to cluster the rows and columns separately, called "Independent Row-Column Clustering" (IRRC) in Shabalin et al. (2009). In this paper we apply k-means IRRC alongside BRIM and CBCE. IRRC is somewhat ill-suited for the finding of bi-communities, as it there is no natural way to pair the row clusters with the column clusters. As such, we apply the following routine to the results of any IRRC method:

   (i) Given cross-correlation matrix $\mathbf{C}$, row clusters $\mathcal{S} := \{S_1, S_2, \ldots, S_K\}$ and column clusters $\mathcal{S}' := \{S'_1, S'_2, \ldots, S'_K\}$.

(ii) Let $\mathbf{C}_{ij}$ be the sub-matrix of $\mathbf{C}$ formed by the row-subset S_i and the column-subset S'_j. Compute a $K \times K$ matrix $\mathbf{M}$ with general entry $\mathbf{M}_{ij}$ defined as the entry-wise mean of $\mathbf{C}_{ij}$.

(iii) Let $m := \max \mathbf{M}_{ij}$, and let i_m, j_m be the indices of m in $\mathbf{M}$.

(iv) Define $C_1 := S_{i_m}$ and $C_2 := S'_{j_m}$. Add the bi-community (C_1, C_2) to a bi-community collection $\mathcal{C}$.

(v) Remove S_{i_m} and S'_{j_m} from $\mathcal{S}$ and $\mathcal{S}'$, respectively. Re-set $K \leftarrow K - 1$. If $K = 0$, terminate and return $\mathcal{C}$. Otherwise, return to step 2.

The algorithm above iteratively finds the strongest pairings of the row and column clusters. In each simulation, we set k, the number of row and column clusters that k-means will find, to the true number of bi-communities in the model that produced the simulation (including the background node set). We also chose the k-means background node set by choosing the bi-community from $\mathcal{C}$ with the closest Jaccard match to the *true* background node set. Each of these procedures can be viewed as "oracle" shortcuts to the k-means IRRC approach, and therefore are generous versions of its use in applications when the number of bi-communities or the background node set are unknown.

5.2.5.4 Simulation settings and results

In this section we present three settings in which parameters of the simulation model are toggled, to assess the competing methods' sensitivities to aspects of the model. In each setting, we move one parameter of the model along an even grid, simulating 50 instances of the model at each parameter value. The performance metrics described in Section 5.2.5.2 are then averaged over the 50 repetitions. We describe the settings and results below.

Increasing the noise variance σ^2. In the first simulation setting, we increase σ^2 significantly. We see that the BRIM method performs quite poorly, and that the overall accuracy of CBCE drops off more quickly than does k-means IRRC. However, this result should be weighed against the fact that the k-means IRCC approach involves oracle settings of the number of bi-modules and the identity of the background node set. Absent these settings, it may be more difficult to achieve similar performance. Furthermore, the CBCE method involves explicit signficance testing, which

will be more sensitive (in general) to the absence of signal than a basic optimization method like k-means.

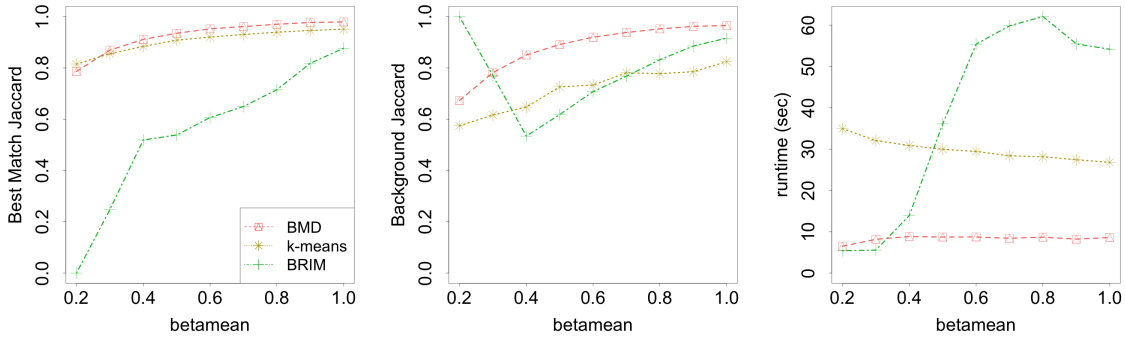

Figure 5.6: Simulation model instances with varying μ_β (betamean).

Decreasing the mean regression parameter μ_β. In the second simulation setting, the mean of the (random) regression parameters is allowed to tend to zero. We see that BRIM remains the least accurate performer, and CBCE and k-means perform comparably, with CBCE dipping a little below for low μ_β. Again, we temper this result with the fact that the k-means approach is generously informative, and that CBCE has built-in background node detection capability. Furthermore, CBCE is approximately three times faster than k-means IRCC in this simulation setting.

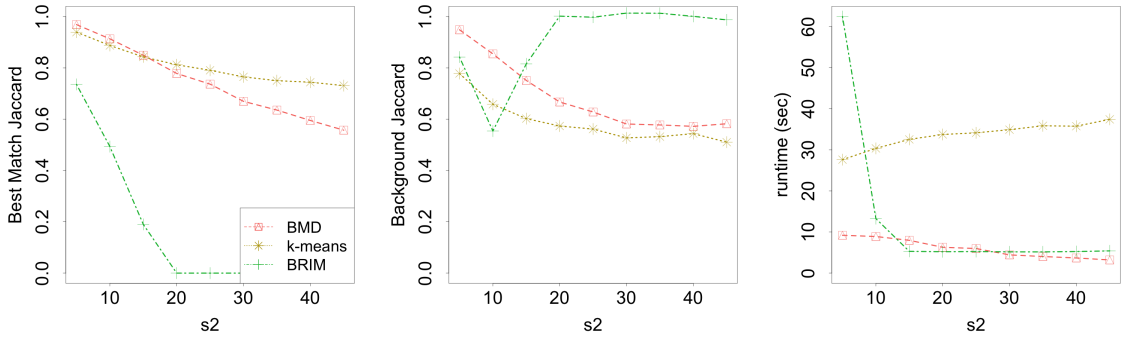

Figure 5.7: Simulation model instances with varying σ^2 (s2).

Increasing the proportion of background nodes vs. bi-community nodes g. In a third simulation setting, the ratio of the size of the background node set to the number of bi-community nodes is increased many-fold. In response to more background nodes in the model, the performance of both BRIM and k-means depreciate considerably, while the performance of CBCE remains near-optimal (see Figure 5.8). This displays the unique ability of a testing approach

to bi-community detection to accurately distinguish between background nodes and nodes in bi-communities. Such an ability is particularly important in large-scale genomic data, as important gene regulation sub-networks comprise only a small fraction of genome. Additionally, the size of the network increases linearly with g, and we see that the computation time of k-means increases at least quadratically in response. This displays some of the computational complications with naive clustering approaches to bi-clustering and bi-community detection, which are surmounted by the set-by-set testing approach inherent to CBCE.

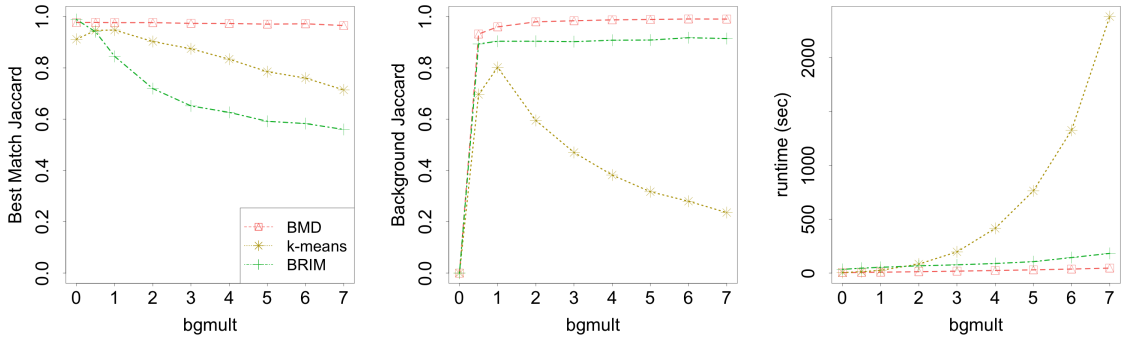

Figure 5.8: Simulation model instances with varying g ("bgmult").

5.2.6 Conclusion

As stated at the outset of this section, the CBCE method is preliminary. Much future work remains to improve the method and apply it to real data. In particular, CBCE is directly applicable to eQTL networks, with genomic loci and (on the other hand) genes forming two halves of a bi-partite correlation network. It will be of great interest to examine eQTL bi-community structure detected by CBCE, and to synthesize the results with those from Genome-Wide Association Studies (GWAS) and other existing features of the genome.

5.3 Acknowledgements

The GTEx data used for the analyses described in this article were from dbGaP accession number phs000424.v3.p1 (http://www.gtexportal.org). The GTEx Project was supported by the Common Fund of the Office of the Director of the National Institutes of Health (commonfund.nih.gov/GTEx). The work of various authors on this paper was supported in part by NIH

R01MH101819-01, NSF DMS1310002, EPA RD83574701, and NHGRI HG007840. The research and analyses in this paper were spurred along by many helpful conversations and phone calls with members and working groups from the GTEx Consortium.

CHAPTER 6
Future Work and Conclusion

The most immediate contributions of the preceding projects have been discussed within their corresponding chapters. Here, broader implications of this work are offered, along with future research directions. The main takeaway of this thesis, seen as a whole, is that the Node-Set Testing (NST) approach to community detection is both conceptually interesting and of wide practical use. It is the author's hope that it is applied to many more network data settings. There are many reasons for this:

1. The framework gives a flexible algorithm which adapts to the number of communities, the amount of community overlap, and the presence of background in natural networks.

2. Statistical testing is inherent to NST methods, providing error guarantees that do not usually come with classical community detection methods.

3. NST methods are conceptually simple to construct for new types of networks. The two necessary components of the framework, a null model and a test statistic, are directly born of two questions central to any community detection analysis: what does the absence of community structure look like, and with what measure could we seek such structure if it existed?

4. Theoretical analysis of NST methods reduce to theoretical analysis of the null model and test statistic. As displayed in this thesis, such analyses are tractable, and are based on natural conceptions of statistical consistency and error rates.

5. The NST methods in this thesis seem to outperform some standard methods on both classic simulations and simulations involving realistic features of networks, like overlapping communities and background nodes.

The established work on CCME, and the preliminary work on CBCE, should serve as examples of NST framework adaptations to specific types of networks with non-standard data types. Hopefully, many more specific adaptations will follow similarly. Some interesting theoretical questions remain unanswered, and some directions left un-pursued:

- As mentioned at the end of Chapter 2, it is unreasonable to assume that p-values are independent under a global OST null. What types of dependence arise from global nulls in various applied settings (e.g., weighted or correlation networks)? What multiple testing rules exist to account for such dependencies, and is global error of the SCS algorithm still controlled when they are applied?

- In Section 3.3.3, an asymptotic consistency theorem was given for the CCME algorithm. The proof follows an approach that should, in principle, apply to all other NST methods. Is there a general consistency framework for NST methods? Under what conditions on the NST test statistic, null model, and any given data-generating model with communities is a general SCS algorithm consistent?

- The NST methods pursued in this thesis were based on *one*-sided tests for association, which means that the communities sought are strictly assortative. Is it possible to implement a two-sided testing version of the NST testing framework? In what contexts are communities in which nodes are both *positively* and *negatively* associated of scientific interest?

I look forward to pursuing these questions in my immediate future.

APPENDIX A

NODE-SET TESTING SUPPLEMENTAL

A.1 Cycles in Fixed Point Search

As remarked in Section 2.3, it is possible for the SCS algorithm to reach a stable sequence $C_1, \ldots, C_J$ that is traversed by the update $U_\alpha(\cdot, \mathcal{G})$. If this happens, we apply the following routine to re-start the algorithm, or return the union of the sequence:

1. If $C_i \cap C_{i+1} = \phi$ for any $i \in [J]$, or if $C_J \cap C_1 = \phi$, terminate the iterations and do not extract a community.
2. Otherwise, define $C^* = \cup_{i=1}^{J} C_i$, and:
   (a) If C^* has been visited previously by SCS, extract C^* into $\mathcal{C}$.
   (b) Otherwise, re-initialize with C^*.

A.2 Proof of Theorem 4

For $k > 1$, define $\mathcal{C}_k := \{B \subseteq [n] : |B| = k\}$. For any $B \in \mathcal{C}_k$, let $p_{(j)}$ be the j-th ordered p-value from $\mathbf{p}(B)$. Then, by construction of the Benjamini-Hochberg procedure, B is a stable community if and only if the event

$$S_n(B, \alpha) := \left\{ p_{(k)} \leqslant \frac{k\alpha}{n} \right\} \bigcap \bigcap_{j>k} \left\{ p_{(j)} > \frac{(j+k)\alpha}{n} \right\} \tag{A.1}$$

occurs. Define $\mathcal{C}_k(\mathbf{D}, \alpha)$ to be the set of all stable communities of size k. Then, using the event in equation A.1 and a union bound,

$$\mathbb{P}_n\big(|\mathcal{C}_k(\mathbf{D}, \alpha)| > 0 \big) \leqslant \sum_{k=1}^{n} \sum_{B \in \mathcal{C}_k} \mathbb{P}_n(S_n(B, \alpha)) \tag{A.2}$$

It is therefore sufficient to show that

$$\sum_{k=1}^{n} \sum_{B \in \mathcal{C}_k} \mathbb{P}_n(S_n(B, \alpha)) \leqslant \alpha \tag{A.3}$$

The remainder of the proof proceeds by induction on n. For any $n' \leqslant n$, we assume $\mathbb{P}_{n'}$ satisfies assumption 1. When $n' = 1$, inequality A.3 is trivial. Thus, the induction hypothesis is that (A.3) holds for all $n' \leqslant n-1$. Fix $1 \leqslant k < n$ and $B \subseteq \mathcal{C}_k$. Let f be the pdf of $p_{(n)}$ under $\mathbb{P}$. Conditioning on $p_{(n)}$, we have

$$\mathbb{P}_n(S_n(B,\alpha)) = \int_\alpha^1 \mathbb{P}_n(S_n(B,\alpha)|p_{(n)} = x)f(x)dx + \int_0^\alpha \mathbb{P}_n(S_n(B,\alpha)|p_{(n)} = x)f(x)dx$$

Since $k < n$, the event $p_{(n)} \leqslant \alpha$ ensures that $S_n(B,\alpha)$ does not occur. Thus, the probability within the right-most integral is equal to zero. We now examine the event $S_n(B,\alpha)$ when $p_{(n)} = x > \alpha$. Note that for fixed $x > \alpha$, if $p_{(n)} = x$, $S_n(B,\alpha)$ will occur if and only if

$$\begin{aligned}
&\left\{p_{(k)} \leqslant \frac{k\alpha}{n}\right\} \bigcap \bigcap_{n>j>k} \left\{p_{(j)} > \frac{(j+k)\alpha}{n}\right\} \\
&\equiv \left\{\frac{p_{(k)}}{x} \leqslant \frac{k}{n-1}\frac{(n-1)\alpha}{nx}\right\} \bigcap \bigcap_{n>j>k} \left\{\frac{p_{(j)}}{x} > \frac{j+k}{n-1}\frac{(n-1)\alpha}{nx}\right\} \qquad \text{(A.4)}
\end{aligned}$$

Note that, conditioned on $p_{(n)} = x$, the random variables $p_{(1)}/x, \ldots, p_{(n-1)}/x$ are ordered independent uniform $[0,1]$. Thus, it is clear from line (A.4) and the definition of $S_n(B,\alpha)$ in line (A.1) that

$$\mathbb{P}_n\left(S_n(B,\alpha) \mid p_{(n)} = x\right) = \mathbb{P}_{n-1}\left(S_{n-1}\left(B, \frac{(n-1)\alpha}{nx}\right)\right) \qquad \text{(A.5)}$$

Therefore,

$$\begin{aligned}
\sum_{k=1}^{n-1}\sum_{B\in\mathcal{C}_k} \mathbb{P}_n(S_n(B,\alpha)) &= \sum_{k=1}^{n-1}\sum_{B\in\mathcal{C}_k} \int_\alpha^1 \mathbb{P}_n(S_n(B,\alpha)|p_{(n)} = x)f(x)dx \\
&= \sum_{k=1}^{n-1}\sum_{B\in\mathcal{C}_k} \int_\alpha^1 \mathbb{P}_{n-1}\left(S_{n-1}\left(B, \frac{(n-1)\alpha}{nx}\right)\right) f(x)dx \\
&= \int_\alpha^1 \sum_{k=1}^{n-1}\sum_{B\in\mathcal{C}_k} \mathbb{P}_{n-1}\left(S_{n-1}\left(B, \frac{(n-1)\alpha}{nx}\right)\right) f(x)dx
\end{aligned}$$

Applying the induction hypothesis to the integrand, we obtain

$$\sum_{k=1}^{n-1} \sum_{B \in \mathcal{C}_k} \mathbb{P}_n(S_n(B, \alpha)) \leqslant \int_\alpha^1 \frac{(n-1)\alpha}{nx} f(x) dx$$

Recall that f is the pdf of the n-th ordered p-value. Hence $f(x) = nx^{n-1}$, and

$$\int_\alpha^1 \frac{(n-1)\alpha}{nx} f(x) dx = \alpha(n-1) \int_\alpha^1 x^{n-2} dx = \alpha(1 - \alpha^{n-1})$$

This bounds the first $n-1$ elements of the sum in (A.3). For the n-th element, note that $\mathcal{C}_n$ contains only the element $B = [n]$, and that trivially $\mathbb{P}_n(S_n([n], \alpha)) = \alpha^n$. Thus overall,

$$\sum_{k=1}^{n} \sum_{B \in \mathcal{C}_k} \mathbb{P}_n(S_n(B, \alpha)) = \mathbb{P}_n(S_n([n], \alpha)) + \sum_{k=1}^{n-1} \sum_{B \in \mathcal{C}_k} \mathbb{P}_n(S_n(B, \alpha)) \leqslant \alpha^n + \alpha(1 - \alpha^{n-1}) = \alpha.$$

■

APPENDIX B

CCME SUPPLEMENTAL

B.1 Proof of Proposition 5

Equation 3.7 follows immediately from the observation in equation 3.3 and the definition of $r_{uv}(\mathbf{s})$. Next, define $e_{uv} := \mathbb{1}(\{u, v\} \in E)$. Note that

$$\mathbb{E}\left(W(u,v)|e_{uv}\right) = f_{uv}(\mathbf{d},\mathbf{s})e_{uv}, \quad \text{and} \quad \text{Var}(W(u,v)|e_{uv}) = \kappa f_{uv}(\mathbf{d},\mathbf{s})^2 e_{uv}.$$

Thus, using the law of total variance,

$$\begin{aligned}
\text{Var}(W(u,v)) &= f_{uv}(\mathbf{d},\mathbf{s})^2\text{Var}(e_{uv}) + \kappa f_{uv}(\mathbf{d},\mathbf{s})^2\mathbb{E}(e_{uv}) \\
&= f_{uv}(\mathbf{d},\mathbf{s})^2\tilde{r}_{uv}(\mathbf{d})(1-\tilde{r}_{uv}(\mathbf{d})) + \kappa f_{uv}(\mathbf{d},\mathbf{s})^2\tilde{r}_{uv}(\mathbf{d}) \\
&= r_{uv}(\mathbf{s})f_{uv}(\mathbf{d},\mathbf{s})\left(1-\tilde{r}_{uv}(\mathbf{d}) + \kappa\right)
\end{aligned}$$

Summing over $v \in B$ gives equation 3.8. ■

B.2 Filtering of $\mathcal{B}_0$ and $\mathcal{C}$

To filter through $\mathcal{B}_0$ and $\mathcal{C}$, we use an inference procedure based on a set-wise z-statistic, analogous to the node-set z-statistic presented in Section 3.3. Define $s(B) := \sum_{v\in B} s(v, B)$. Note that $s(B)$ has an easily derivable expectation and standard deviation under the continuous configuration model, which we denote (respectively) by $\mu(B|\theta)$ and $\sigma(B, |\theta)$. We define the corresponding z-statistic and an approximate p-value by

$$z(B|\theta) := \frac{s(B) - \mu(B|\theta)}{\sigma(B|\theta)}, \qquad p(B|\theta) := 1 - \Phi(z(B|\theta))$$

Before initializing the SCS algorithm on sets in $\mathcal{B}_0$, we compute the p-value above for each member set, and remove any that are not significant at FDR level $\alpha = 0.05$. This greatly reduces the

number of extractions CCME must perform, and reduces the probability of convergence on small, spurious communities.

We also use $z(B|\theta)$ to filter near-matches in $\mathcal{C}$, once all FPS extractions have terminated and empty sets removed. To do so, we require an overlap "tolerance" parameter $\tau \in [0, 1]$. First, we create a (non-symmetric) $|\mathcal{C}| \times |\mathcal{C}|$ matrix O with general element $O_{ij} := |C_i \cap C_j|/|C_i|$, which measures the proportional overlap of C_i into C_j. After setting the diagonal of O to zero, the filtering proceeds as follows:

1. Find indices $i \neq j$ corresponding to the maximum entry of O.

2. If $O_{ij} < \tau$, terminate filtering.

3. Remove either C_i or C_j from $\mathcal{C}$, whichever has the smaller $z(B|\theta)$.

4. Re-compute O, set its diagonal to zero, and return to step 1.

For all simulations and real-data analyses in this paper, we employed this algorithm with $\tau = 0.9$. To further decrease the computation time of CCME, as we proceed through $\mathcal{B}_0$, we skip sets that were formed from nodes that have already been extracted into $\mathcal{C}$. We find that, in practice, none of these adjustments harm CCME's ability to find statistically significant overlapping communities. Indeed, the simulation results mentioned in Section 3.5.2.2 show that CCME outperforms competing methods with overlap capabilities.

B.3 Simulation Framework Preliminaries

In this section and the following sections we describe the benchmarking simulation framework used for the performance analysis of CCME competitor methods in Section 3.5 of the main document. In Table B.1, we list and name the complete list of parameters controlling the simulated networks:

B.4 Simulation of community nodes

The framework is capable of simulating networks with or without background nodes. For now, we describe the simulation procedure without background nodes, i.e. with $n_b = 0$. Later, we

Table B.1: Simulation model parameters

n: Number of nodes in communities	n_b: Number of nodes in background
$m_{\max}$: Max community size	$m_{\min}$: Min community size
τ_1: Power-law for degree parameters	τ_2: Power-law for community sizes
k: Mean of degree parameter power-law	$k_{\max}$: Maximum degree parameter
s_e: Within-community edge signal	s_w: Within-community weight signal
o_n: Number of nodes in multiple communities	o_m: Number of memberships for overlap nodes
F: Distributions of edge weights	σ^2: Variance parameter for $\mathcal{P}$
β: Power-law for strength parameters	

describe how to simulate a network with background nodes, which involves a slight modification to the procedure in this section. Regardless of the presence of background nodes, the first step is to determine community sizes and node memberships.

B.4.1 Community structure and node degree/strength parameters

In this section we describe how to obtain the community assignments of the n community-nodes. The goal is to obtain a cover $\mathcal{C} := \{C_1, \dots C_K\}$ of the nodes $[n]$. The following steps to obtain $\mathcal{C}$ are almost exactly as those from the benchmark in Lancichinetti and Fortunato (2009), used extensively in Lancichinetti et al. (2011) and Xie et al. (2013).

1. Each of the o_n overlapping node will have o_m memberships. Let $n_m := n + o_n(o_m - 1)$ be the number of node *memberships* present in the network.

2. Draw community sizes from a power law with maximum value $m_{\max}$, minimum value $m_{\min}$, and exponent $-\tau_2$, until the sum of community sizes is greater than or equal to n_m. If the sum is greater than n_m, we reduce the sizes of the communities proportionally until the sum is equal to n_m.

3. Form a bipartite graph of community markers on one side and node markers on the other. Each community marker has number of empty node slots given by step (b), and each node has a number of memberships given by step (a). Sequentially pair node memberships and community node slots uniformly at random, without replacement, until every node membership is paired with a community. This process is a bipartite version of the standard configuration model. For more details, see Lancichinetti and Fortunato (2009).

With the community assignments in hand, simulation of the network proceeds according to the Weighted Stochastic Block Model as outlined in Section 3.5 from the main text. We describe choices for particular components of this model in the following section.

B.4.2 Simulation of edges and weights

As described in Section 3.5, we set the $\mathbf{P}$ and $\mathbf{M}$ matrices to have diagonals equal to s_e and s_w (respectively, see Table B.1), and off-diagonals equal to 1. We note that this homogeneity facilitates creating networks with overlapping communities. With variance in the diagonal of $\mathbf{P}$, for example, it would not be obvious with what probability to connect overlapping nodes that overlap to two of the same communities, simultaneously. It remains to obtain the strength and degree propensity parameters ψ and ϕ; we do so analogously to the simulation framework in Lancichinetti et al. (2011). We first draw ϕ from a power law with exponent τ_1, mean k, and maximum $k_{\max}$ (see Table B.1). Next we set ψ by the formula $\psi(u) = \phi(u)^{\beta+1}$ (this is mentioned in Section 3.5).

It is worth noting here that, under the model given below, the expected degree of node u is *approximately* $\phi(u)$ and the expected strength *approximately* $\psi(u)$. Therefore, heterogeneity/skewness in ϕ and ψ induce heterogeneity/skewness in the degrees and strengths of the simulated networks. However, for reasons that will be made clear in the sections to follow, we prefer to have (at least) the total expected degree and total expected strength of the simulated networks match ϕ_T and ψ_T, respectively. As such, after drawing ϕ from its power law and determining ψ from the aforementioned formula, we scale these vectors so their sums match the total expected degree and strength of $\mathcal{G}$. The scaling constants depend on $\mathbf{P}$ and $\mathbf{M}$ and are easily derivable from the model's generative algorithm (described in Section 3.3.3.1 of the main text).

B.4.3 Parameter settings

Here we list the "default" settings of the simulation model, mentioned in Section 3.5 of the main text. The following choices for parameters were made regardless of the simulation setting: $\tau_2 = -2$, $k = \sqrt{n}$, $k_{\max} = 3k$ (three settings which make the degree/strength distributions skewed and the network sparse), $\beta = 0.5$ (to induce a non-trivial power law between strengths and degrees), $\tau_1 = -1$, $m_{min} = n/5$, $m_{max} = 3m_{max}/2$ (settings which produce between 3 and 7 communities

per network with skewed size distribution), and $\sigma^2 = 1/2$. Other parameter choices are specific to the simulation settings, and described in Section 3.5 of the main text.

B.5 Background node simulation

If $n_b > 0$, we generate a network with n community nodes, and then add n_b background nodes, generating all remaining edges and weights according to the continuous configuration null model introduced in the main text. First, we obtain node-wise parameters for all $n + n_b$ nodes, yielding vectors ϕ and ψ as in Section B.4. In a simulated network without background, $\phi(u)$ and $\psi(u)$ are approximately $\mathbb{E}[d(u)]$ and $\mathbb{E}[s(u)]$, respectively. To ensure that this remains the case in a network for which background nodes are added after the simulation of community nodes, we must split up each $\phi(u)$ and $\psi(u)$ into community and background portions. A few other adjustments must also be made after the simulation of community nodes. To this end, define

- $[n]_C := \{1, \ldots, n\}$; $[n]_B := \{n+1, \ldots, n+n_b\}$

  $\rightarrow$ community and background node sets

- $\phi_{C,T} := \sum_{[n]_C} \phi(u)$; $\phi_{B,T} := \sum_{[n]_B} \phi(u)$

  $\rightarrow$ target total degrees of community and background nodes

- $\phi_C(u) := \frac{\phi_{C,T}}{\phi_T} \phi(u)$; $\phi_B(u) := \frac{\phi_{B,T}}{\phi_T} \phi(u)$

  $\rightarrow$ target edge-counts between u and the community and background nodes

- $\phi_{1,T} := \sum_{[n]_C} \phi_C(u)$; $\phi_{2,T} := \sum_{[n]_B} \phi_B(u)$

  $\rightarrow$ target total degrees of community and background *subnetworks*

- $d^o_C(u) := \sum_{v \in [n]_C} \mathbb{1}(\{u, v\} \in E)$; $d^o_B(u) := \sum_{v \in [n]_B} \mathbb{1}(\{u, v\} \in E)$

  $\rightarrow$ observed edge-counts between u and the community and background nodes

The above definitions exist analogously for the strength parameters ψ (replacing "d" with "s" where appropriate). The word "target" indicates that we will set up the background simulation model so that these values are the approximate expected values of the graph statistics they represent.

B.5.1 Adjusted community-node simulation model

The only adjustment to be made to the simulation of community nodes, described in Section B.4.2, is that the degree and strength parameters are set to a certain *fraction* of their original values. This accounts for the eventual addition of background nodes, where the remaining (random) part of each nodes degree and strength is to be simulated. So, the community-node simulation (if background nodes are to be added later) follows the process descibed in Section B.4 with degree parameters $\{\phi_C(1), \ldots, \phi_C(n)\}$ and strength parameters $\{\psi_C(1) \ldots \psi_C(n)\}$.

B.5.2 Edges and weights for background

For the simulation of the background nodes (following the community nodes) our goal is to specify adjusted degree/strength parameters ϕ' and ψ' given the observed edge-sums $\{d^o_C(1), \ldots, d^o_C(n)\}$ and weight-sums $\{s^o_C(1), \ldots, s^o_C(n)\}$ from the community nodes. In what follows we describe this specification for ϕ' only; the specification for ψ' is exactly analogous. We first represent ϕ'_T, which we have yet to determine, into community and background totals:

$$\phi'_T = \phi'_{C,T} + \phi'_{B,T}$$

Since the background subnetwork has not yet been generated, we make the specification $\phi'(u) := \phi(u)$ for all $u \in [n]_B$, and hence $\phi'_{B,T} = \phi_{B,T}$ is known. To address $\phi'_{C,T}$, note that for each community node $u \in [n]_C$, $\phi'(u)$ may be represented similarly:

$$\phi'(u) = \phi'_C(u) + \phi'_B(u)$$

This reduces the problem of specifying $\phi'(u)$ to specifying $\phi'_C(u)$ and $\phi'_B(u)$. Since the community node subnetwork has already been generated, we set $\phi'_C(u) \leftarrow d^o_C(u)$. Next, recalling that $\phi_B(u) := \frac{\phi_{B,T}}{\phi_T}\phi(u)$, we make the specification $\phi'_B(u) := \frac{\phi_{B,T}}{\phi'_T}\phi(u)$ (which must be solved for via ϕ'_T, in the following). So, in total, we have

$$\phi'(u) = \begin{cases} d^o_C(u) + \frac{\phi_{B,T}}{\phi'_T}\phi(u), & u \in [n]_C \\ \phi(u), & u \in [n]_B \end{cases}$$

Therefore we can solve for ϕ'_T with the equation

$$\begin{aligned}\phi'_T &:= \sum_{u\in[n]_C\cup[n]_B} \phi'(u) \\ &= \sum_{u\in[n]_C}\left[d^o_C(u) + \frac{\phi_{B,T}}{\phi'_T}\phi(u)\right] + \sum_{u\in[n]_B}\phi(u) \\ &= d^o_{C,T} + \frac{\phi_{B,T}}{\phi'_T}\phi_{C,T} + \phi_{B,T}\end{aligned}$$

Where $d^o_{C,T} := \sum_{u\in[n]_C} d^o_C(u)$. The solution for ϕ'_T from this quadratic is

$$\phi'_T = \frac{\phi_{B,T} + d^o_{C,T}}{2} + \sqrt{\frac{(\phi_{B,T} + d^o_{C,T})^2}{4} + \phi_{C,T}\phi_{B,T}} \tag{B.1}$$

which then immediately gives the full vector ϕ'. We can now simulate the remaining edges in the network. Specifically, for each $u \in [n]_B$ and each $v \in [n]_C \cup [n]_B$, we simulate an edge according to

$$\mathbb{P}\left(\{u,v\} \in E\right) = \frac{\phi'(u)\phi'(v)}{\phi'_T} \text{ independent across node pairs} \tag{B.2}$$

We solve for ψ' analogously. Then for each $u \in [n]_B$ and each $v \in [n]$, we simulate an edge weight according to

$$W(u,v) = \begin{cases} f_{uv}(\phi',\psi')\xi_{uv}, & \{u,v\} \in E \\ 0, & \{u,v\} \notin E \end{cases}$$

where $\xi \sim F$, is as it was for the generation of the community node subnetwork.

The above simulation steps correspond precisely to the continuous configuration model with parameters $(\phi', \psi', \mathcal{P}, \theta)$. Some basic computational trials have shown that, for large networks, the solution for ϕ'_T is quite close to ϕ_T. Therefore, for each $u \in [n]_B$, $\mathbb{E}[d(u)]$ is almost exactly $\phi(u)$, i.e. what it would be under the model in B.4.2, without background nodes. The same holds for the strengths and expected strengths. Together with equation B.2, this implies the background nodes are behaving according to the continuous configuration model, even as they are a sub-network within a larger network with communities.

To illustrate these points, we simulated a sample network from the default framework with parameters $n = 5,000$, $n_b = 1,000$, $s_e = s_w = 3$, disjoint communities, and other parameters

specified by B.4.3. These settings are akin to what was used in Section 3.5 of the main text. First we plotted ϕ' and ψ' against the empirical strengths and degrees with lowess curves to check the match. Figure $B.1$ shows the match is quite close. Second, for each node $u \in [n]$ and for each node

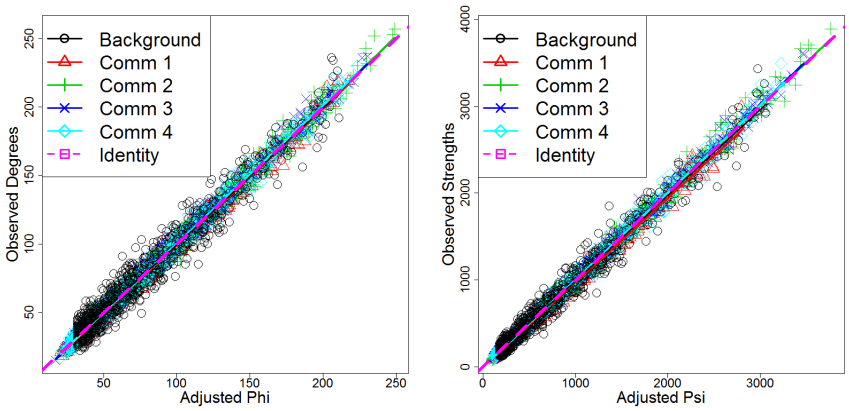

Figure B.1: Empirical degrees/strengths vs. adjusted parameters for the example network

block B (either a true community or the background node set) we may calculate the empirical z-score for $s(u : B)$ as described in Section 3.3 of the main text. The z-score for $s(u : B)$ is a measure of connection significance, with respect to the continuous configuration model, between u and B. Let K be the number of true communties in the network. For each $i, j = 1, \ldots, K + 1$, where $K + 1$ is the index of the background node block, we computed the empirical average of z-statistics between nodes u from node block i the node block B corresponding to index j. Theses empirical averages can be arranged in a $(K + 1) \times (K + 1)$ matrix showing the average inter-block connectivities of the network. In Figure B.2 we display a visualization of this matrix, which shows preferential connection within communities, and null connection between the background nodes and all blocks.

B.6 Proof of Theorem 6 and supporting lemmas.

Here we give the proof of Theorem 6 in Section 3.3.2. We start with supporting lemmas. Recall the definition of the average degree parameter λ_n, the normalized r^{th}-moment $L_{n,r}$, and other associated definitions from Section 3.3.2. For the purposes of the results below, we define the

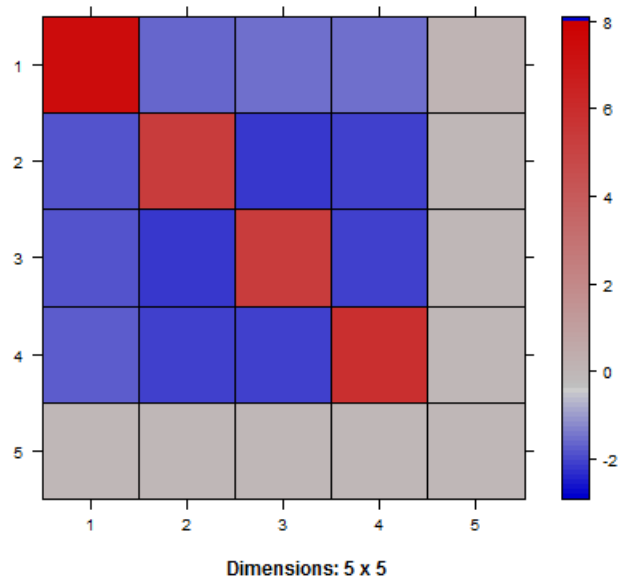

Figure B.2: Average empirical z-statistics between nodes and node blocks

following generalization of $L_{n,r}$, given a node set $B_n \subseteq [n]$ and $b_n := |B_n|$:

$$L_{n,r}(B_n) := b_n^{-1} \sum_{u \in B_n} \{d_n(u)/\lambda_n\}^r$$

Note that $L_{n,r}([n]) = L_{n,r}$. Recall that in the setting of Theorem 6, the node set B_n is chosen uniformly from the node set $[n]$. The first result involves a *deterministic* sequence $\{B_n\}_{n \geqslant 1}$:

Lemma 25. *For each $n > 1$, let $\mathcal{G}_n$ be generated by the continuous configuration model with parameters $\theta_n = (\mathbf{d}_n, \mathbf{s}_n, \kappa_n)$ and common weight distribution F. Fix a node sequence $\{u_n\}_{n>1}$ with $u_n \in [n]$ and a positive integer sequence $\{b_n\}_{n>1}$ with $b_n \leqslant n$. Suppose the parameter sequence $\{d_n(u_n)\}_{n \geqslant 1}$ satisfies*

$$\frac{d_n(u_n) b_n}{n} \to \infty \text{ as } n \to \infty$$

Fix $\varepsilon > 0$ as in Assumption 3, and choose $\delta \in (0,1)$ such that $2\beta\delta < \varepsilon$. Fix a sequence of sets $\{B_n\}_{n>1}$ with $|B_n| = b_n$ for all n, and suppose that for $r = 2\beta+1$ and $r = \beta(2+\delta)+1$, the sequence $\{L_{n,r}(B_n)\}_{n>1}$ is bounded away from zero and infinity. Then

$$\frac{S(u_n, B_n) - \mu_n(u_n, B_n|\theta_n)}{\sigma_n(u_n, B_n|\theta_n)} \Rightarrow \mathcal{N}(0,1) \quad as \quad n \to \infty$$

Proof. In what follows, the functions r_{uv} and $\tilde{r}_{uv}$ from Section 3.1 will be used extensively. Note that for any nodes u, v, $\mathbb{E}W(u,v) = r_{uv}(\mathbf{s})$. Thus by the classical Lyapunov central limit theorem it suffices to show that

$$\frac{\sum\limits_{v \in B_n} \mathbb{E}|W(u_n, v) - r_{u_n v}(\mathbf{s}_n)|^{2+\delta}}{\left(\sqrt{\sum\limits_{v \in B_n} \mathbb{E}\left\{(W(u_n, v) - r_{u_n v}(\mathbf{s}_n))^2\right\}}\right)^{2+\delta}} \to 0 \tag{B.3}$$

as n tends to infinity. The following derivations hold for any fixed $n > 1$, so we suppress dependence on n from u_n, and B_n, and similar expressions. In what follows, we use the slight abuse of notation $E_{uv} := \mathbb{1}(\{u, v\} \in E)$. For the numerator of (B.3), we have

$$\begin{aligned}
\mathbb{E}|W(u,v) - r_{uv}(\mathbf{s})|^{2+\delta} &= \left(\frac{r_{uv}(\mathbf{s})}{\tilde{r}_{uv}(\mathbf{d})}\right)^{2+\delta} \mathbb{E}\left(|\xi_{uv} E_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right) \\
&= f_{uv}(\mathbf{d}, \mathbf{s})^{2+\delta} \cdot \mathbb{E}\left(|\xi_{uv} E_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right),
\end{aligned} \tag{B.4}$$

by definition of the model in Section 3.2.1. Moreover, by the law of total variance,

$$\begin{aligned}
\mathbb{E}(|\xi_{uv} E_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}) &= (1 - \tilde{r}_{uv}(\mathbf{d}))\tilde{r}_{uv}(\mathbf{d})^{2+\delta} + \tilde{r}_{uv}(\mathbf{d})\mathbb{E}|\xi_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta} \\
&= \left\{(1 - \tilde{r}_{uv}(\mathbf{d}))\tilde{r}_{uv}(\mathbf{d})^{1+\delta} + \mathbb{E}|\xi_{uv} - \tilde{r}_{uv}(\mathbf{d})|^{2+\delta}\right\} \cdot \tilde{r}_{uv}(\mathbf{d}) \\
&\leqslant C \cdot \tilde{r}_{uv}(\mathbf{d})
\end{aligned} \tag{B.5}$$

for some positive constant C, by Assumption 5. Next, we note that by Assumption 2, there exist positive constants $A < B$ such that for all $v \in [n]$,

$$A \cdot d_n(v)^\beta \leqslant \frac{s_n(v)}{d_n(v)} \leqslant B \cdot d_n(v)^\beta,$$

for n sufficiently large. Thus, if $r_{uv}(\mathbf{d}) \leqslant 1$, $\tilde{r}_{uv}(\mathbf{d}) = r_{uv}(\mathbf{d})$, and

$$f_{uv}(\mathbf{d}, \mathbf{s}) = \frac{r_{uv}(\mathbf{s})}{\tilde{r}_{uv}(\mathbf{d})} = \left(\frac{d_T}{s_T}\right) \frac{s(u)s(v)}{d(u)d(v)} \leqslant B \cdot \left(\frac{d_T}{s_T}\right) \{d(u)d(v)\}^\beta. \tag{B.6}$$

If $r_{uv}(\mathbf{d}) > 1$, $\tilde{r}_{uv}(\mathbf{d}) = 1$, and by Assumption 4 there exists B' such that

$$\begin{aligned} f_{uv}(\mathbf{d}, \mathbf{s}) = \frac{s(u)s(v)}{s_T} &\leqslant B \cdot \left(\frac{d(u)d(v)}{s_T}\right) \{d(u)d(v)\}^{\beta} \\ &= B \cdot \left(\frac{d_T}{s_T}\right) r_{uv}(\mathbf{d})\{d(u)d(v)\}^{\beta} \leqslant B' \cdot \left(\frac{d_T}{s_T}\right) \{d(u)d(v)\}^{\beta}. \end{aligned} \tag{B.7}$$

Therefore, combining (B.5)-(B.7) with (B.4), there exists C' such that

$$\begin{aligned} \mathbb{E}|W(u,v) - r_{uv}(\mathbf{s})|^{2+\delta} &\leqslant C' \left(\frac{d_T}{s_T}\right)^{2+\delta} \cdot \{d(u)d(v)\}^{\beta(2+\delta)} \tilde{r}_{uv}(\mathbf{d}) \\ &= C' \left(\frac{d_T}{s_T}\right)^{2+\delta} \cdot \{d(u)d(v)\}^{\beta(2+\delta)} \frac{d(u)d(v)}{d_T} \\ &\leqslant C' \cdot d_T^{1+\delta} s_T^{-(2+\delta)} \cdot \{d(u)d(v)\}^{\beta(2+\delta)+1} \end{aligned} \tag{B.8}$$

A similar analysis of the summands in the denominator of (B.3) gives

$$\mathbb{E}\left\{(W(u,v) - r_{uv}(\mathbf{s}))^2\right\} \geqslant C'' \cdot d_T s_T^{-2} \cdot \{d(u)d(v)\}^{2\beta+1} \tag{B.9}$$

for appropriately chosen C''. Let $b = |B|$. Combining (B.8) and (B.9), with some algebra, we find that the left side of (B.3) is (up to a constant) less than

$$\begin{aligned} &\left(\frac{d(u)}{d_T}\right)^{-\delta/2} \cdot \frac{\sum_{v \in B} d(v)^{\beta(2+\delta)+1}}{\left(\sum_{v \in B} d(v)^{2\beta+1}\right)^{1+\delta/2}} \\ = &\left(\frac{d(u)}{d_T} b\lambda\right)^{-\delta/2} \cdot \frac{b^{-1} \sum_{v \in B} (d(u)/\lambda)^{\beta(2+\delta)+1}}{\left\{b^{-1} \sum_{v \in B} (d(u)/\lambda)^{2\beta+1}\right\}^{1+\delta/2}} \\ = &\left(\frac{d(u)}{d_T} b\lambda\right)^{-\delta/2} \cdot \frac{L_{n,\beta(2+\delta)+1}(B)}{(L_{n,2\beta+1}(B))^{1+\delta/2}} \;=\; O\left\{\left(\frac{d(u)}{d_T} b\lambda\right)^{-\delta/2}\right\} \end{aligned} \tag{B.10}$$

where the final term follows from our assumptions on $L_{n,\beta(2+\delta)+1}(B_n)$ and $L_{n,2\beta+1}(B_n)$. By definition, $d_{n,T} = n\lambda_n$, so the final expression above is $O\left\{(d_n(u_n)b_n/n)^{-\delta/2}\right\} = o(1)$ by assumption. Thus (B.3) holds and the result follows. ■

We now proceed with the proof of Theorem 6. Proposition 25 yields the CLT for $\hat{s}(u_n : B_n)$ for a deterministic sequence of vertex sets $\{B_n\}_{n\geqslant 1}$ satisfying regularity properties. The remainder of the argument shows that if B_n is selected uniformly at random then, under the assumptions of Theorem 6, these regularity properties are satisfied with high probability. We begin with a few preliminary definitions and results.

Definition 26. *A sequence of random variables $\{X_n\}_{n\geqslant 1}$ is said to be* asymptotically uniformly integrable *if*

$$\lim_{M\to\infty} \limsup_{n\to\infty} \mathbb{E}\left\{|X_n|\mathbb{1}(|X_n| > M)\right\} = 0$$

Theorem 27. *Let $f : \mathbb{R}^k \mapsto \mathbb{R}^k$ be measurable and continuous at every point in a set C. Suppose $X_n \xrightarrow{w} X$ where X takes its values in an interval C. Then $\mathbb{E}f(X_n) \to \mathbb{E}f(X)$ if and only if the sequence of random variables $f(X_n)$ is asymptotically uniformly integrable.*

Proof. See Asymptotic Statistics (Van der Vaart 2000), page 17. ■

We now give a technical lemma (needed for a subsequent result) which uses Theorem 27:

Lemma 28. *Let $X_1, X_2, \ldots$ be non-negative random variables and let $s, \varepsilon > 0$. If the sequences $\{\mathbb{E}X_n^s\}_{n\geqslant 1}$ and $\{\mathbb{E}X_n^{s+\varepsilon}\}_{n\geqslant 1}$ are bounded away from zero and infinity, then $\{\mathbb{E}X_n^r\}_{n\geqslant 1}$ is bounded away from zero and infinity for every $r \in (0, s+\varepsilon)$.*

Proof. Suppose by way of contradiction that there exists $t \in (0, s+\varepsilon)$ such that $\liminf_n \mathbb{E}X_n^t = 0$. Then $\lim_k \mathbb{E}X_{n_k}^t = 0$ along a subsequence $\{n_k\}$. As the random variables $X_{n_k}^t$ are non-negative, $X_{n_k}^t \xrightarrow{d} 0$, and it follows from the continuous mapping theorem that $X_{n_k} \xrightarrow{w} 0$. As $M^{\varepsilon/s} X_n^s \mathbb{1}(X_n^s > M) \leqslant X_n^{s+\varepsilon}$, we find that

$$\lim_{M\to\infty} \limsup_{k\to\infty} \mathbb{E}\{X_{n_k}^s \mathbb{1}(X_{n_k}^s > M)\} \;\leqslant\; \lim_{M\to\infty} M^{-\varepsilon/s} \limsup_{k\to\infty} \mathbb{E}(X_{n_k}^{s+\varepsilon}) = 0$$

as $\mathbb{E}(X_n^{s+\varepsilon})$ is bounded by assumption. It then follows from Theorem 27 and the fact that $X_{n_k}^s \xrightarrow{w} 0$ that $\mathbb{E}X_{n_k}^s \to 0$ as $k \to \infty$, violating our assumption that $\mathbb{E}X_n^s$ is bounded away from zero. We conclude that $\mathbb{E}X_n^r$ is bounded away from zero for $r \in (0, s+\varepsilon)$. On the other hand, if $r \in (0, s+\varepsilon)$

then for each $n \geqslant 1$

$$\mathbb{E}\{X_n^r \mathbb{1}(X_n > 1)\} \;\leqslant\; \mathbb{E}\{X_n^{s+\varepsilon} \mathbb{1}(X_n > 1)\} \;\leqslant\; \sup_n \mathbb{E}\{X_n^{s+\varepsilon}\}$$

As the last term is finite by assumption and $\mathbb{E}\{X_n^r \mathbb{1}(X_n \leqslant 1)\}$ is at most one, it follows that $\mathbb{E}(X_n^r)$ is bounded. ■

Lemma 29. *Suppose a degree parameter sequence $\{\mathbf{d}_n\}_{n\geqslant 1}$ satisfies Assumption 3 from Section 3.3.2. For each n, let B_n be a randomly chosen subset of $[n]$ of size b_n, where $b_n \to \infty$. Fix $\varepsilon > 0$ as in Assumption 3, and choose δ so that $2\beta\delta < \varepsilon$. Then for every $r \in (0,\ \beta(2+\delta)+1]$, there exists an interval $I_r = (a_r, b_r)$ with $0 < a_r < b_r < \infty$ such that $\mathbb{P}\{L_{n,r}(B_n) \in I_r\} \to 1$ as $n \to \infty$.*

Remark: Note that the function $L_{n,r}(\cdot)$ is non-random. The probability appearing in the conclusion of Lemma 29 depends only on the random choice of the vertex set B_n.

Proof. Let D_n and D_n' be drawn uniformly-at-random from $\mathbf{d}_n$ without replacement, and fix $r \in (0,\ \beta(2+\delta)]$. A routine calculation gives

$$\mathrm{Var}\{L_{n,r}(B_n)\} = b_n^{-1}\lambda_n^{-2r}\left[\mathrm{Var}\{D_n^r\} + \{b_n - 1\}\mathrm{Cov}\{D_n^r, (D_n')^r\}\right].$$

Note that $\mathbb{E}(D_n^r) = \lambda_n^r L_{n,r}$ and $\mathbb{E}(D_n^{2r}) = \lambda_n^{2r} L_{n,2r}$, so $\mathrm{Var}(D_n^r) = \lambda_n^{2r}(L_{n,2r} - L_{n,r})$. Furthermore, a simple calculation shows that $\mathrm{Cov}\{D_n^r, (D_n')^r\}$ is negative for every r, and therefore $\mathrm{Var}\{L_{n,r}(B_n)\} \leqslant b_n^{-1}(L_{n,2r} - L_{n,r})$. Our choice of δ ensures that $2r < 4\beta + 2 + \varepsilon$, and it then follows from Lemma 28 and Assumption 3 that $L_{n,2r}$ and $L_{n,r}$ are bounded. Thus $\mathrm{Var}\{L_{n,r}(B_n)\} = O(b_n^{-1})$. Define $\Delta := \liminf_n\ L_{n,r}/2$, which is positive by Assumption 3, and let

$$I_r := \left(\liminf_{n\to\infty} L_{n,r} - \Delta,\ \limsup_{n\to\infty} L_{n,r} + \Delta\right) \tag{B.11}$$

As $\mathbb{E}\{L_{n,r}(B_n)\} = L_{n,r}$, an application of Chebyshev's inequality yields the bound

$$\begin{aligned}\mathbb{P}\{L_{n,r}(B_n) \notin I_r\} \;&\leqslant\; \mathbb{P}\{|L_{n,r}(B_n) - \mathbb{E}[L_{n,r}(B_n)]| > \Delta/2\} \\ &\leqslant\; \frac{4\mathrm{Var}\{L_{n,r}(B_n)\}}{\Delta^2} \;=\; O(b_n^{-1}).\end{aligned}$$

As b_n tends to infinity with n, the result follows. ■

B.6.1 Completing the proof of Theorem 6.

Let ε and δ be as in Proposition 25 and Lemma 29. Note that since $d_n(u_n) \leqslant n$ for all n, our assumption that $b_n d_n(u_n)/n \to \infty$ implies $|B_n| = b_n \to \infty$. Hence by lemma 29, we have that for both $r = \beta(2+\delta)+1$ and $r = 2\beta+1$, there exists a positive, finite interval I_r such that $\mathbb{P}\{L_{n,r}(B_n) \in I_r\} \to 1$ as $n \to \infty$. Thus given any subsequence $\{n_k\}_{k\geqslant 1}$ we can find a further subsequence $\{n'_k\}_{k\geqslant 1}$ such that $L_{n'_k,r}(B_{n'_k}) \in I_r$ almost surely as $k \to \infty$, which means this sequence is bounded away from zero and infinity in k. Now using Proposition 25, for almost every ω we have

$$\frac{S_{n'_k}(u_{n'_k}, B_{n'_k}) - \mu_{n'_k}(u_{n'_k}, B_{n'_k}|\theta_{n'_k})}{\sigma_{n'_k}(u_{n'_k}, B_{n'_k}|\theta_{n'_k})} \Rightarrow \mathcal{N}(0,1) \text{ as } k \to \infty$$

Applying the subsequence principle completes the proof. ■

B.7 Proof of Theorems 8-9 and supporting lemmas.

Throughout this section, notation and conventions from Section 3.3.3.1 of the main document will be used, though we suppress dependence on n for convenience. Further recall functions r and f from Section 3.1. The following additional notation will be used throughout this section:

- Define $\phi_T := \sum_{v\in[n]} \phi(v)$ and $\psi_T := \sum_{v\in[n]} \psi(v)$. For each $j \in [K]$, define $\tilde{\pi}^0_j := \sum_{v\in\mathcal{C}_j} \phi(v)/\phi_T$ and $\tilde{\pi}_j := \sum_{v\in\mathcal{C}_j} \psi(v)/\psi_T$. Let $\tilde{\boldsymbol{\pi}}^0$ and $\tilde{\boldsymbol{\pi}}$ be the associated vectors.
- Let $\langle\cdot,\cdot\rangle$ denote the vector dot-product. For a general symmetric matrix $\mathbf{A}$, let $\mathbf{A}[i,j]$ be the i,j-th entry, and $\mathbf{A}[i]$ the i-th column. Define $\mathbf{H} := \mathbf{P}\cdot\mathbf{M}$, the entry-wise product.
- Let $D(u), S(u)$ be the random degree, strength of node $u \in [n]$, let $\tilde{d}(u)$, $\tilde{s}(u)$ be the corresponding expectations, and let $\mathbf{D}, \mathbf{S}, \bar{\mathbf{d}}, \bar{\mathbf{s}}$ be the associated n-vectors. Define $\bar{s}_T := \sum_{v\in[n]} \bar{s}(v)$ and $\bar{d}_T := \sum_{v\in[n]} \bar{d}(v)$.

We now define a *empirical* population version of the variance estimate:

Definition 30. *Fix $n > 1$ and let E and W be the edge set and weight function from $\mathcal{G}_n$, the n-th random weighted network from the sequence in the setting of Theorem 8. Let $\mathbf{x}, \mathbf{y}$ be arbitrary*

n-vectors with positive entries. For nodes $u, v \in [n]$, define

$$V_{uv}(\mathbf{x}, \mathbf{y}) := (W(u,v) - f_{uv}(\mathbf{x}, \mathbf{y}))^2, \qquad v_{uv}(\mathbf{x}, \mathbf{y}) := \mathbb{E}\left\{V_{uv}(\mathbf{x}, \mathbf{y}) \middle| \{u, v\} \in E\right\}.$$

Define the empirical *population variance estimator as follows:*

$$\kappa_*(\mathbf{x}, \mathbf{y}) := \frac{\sum_{uv \in E} v_{uv}(\mathbf{x}, \mathbf{y})}{\sum_{uv \in E} f_{uv}(\mathbf{x}, \mathbf{y})^2}$$

The estimator $\kappa_*(\mathbf{x}, \mathbf{y})$ is called "empirical" because it depends on the random edge set E. Despite this, it has a deterministic bound, a fact which is part of Lemma 31. Throughout the remaining results, denote $\Theta := (\mathbf{D}, \mathbf{S}, \hat{\kappa}(\mathbf{D}, \mathbf{S}))$ and $\theta_* := (\bar{\mathbf{d}}, \bar{\mathbf{s}}, \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}))$, where the estimator $\hat{\kappa}$ is the estimator from Section 3.2.2.

Recall the definition of the asymptotic order of the average degree $\lambda_n := n\rho_n$, from Section 3.3.3.2 in the main text. With this and the conventions above, Lemma 31 establishes basic facts about the WSBM:

Lemma 31. *Fix $n > 1$, and let $\mathcal{G}_n$ be a random network generated by a WSBM. For all nodes $u, v \in [n]$, under Assumptions 6 and 7,*

(1) $\bar{d}(u) = \lambda_n \phi(u) \langle \tilde{\boldsymbol{\pi}}^0, \mathbf{P}[c(u)] \rangle$ *and* $\bar{s}(u) = \lambda_n \psi(u) \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[c(u)] \rangle$

(2) $m_-^2 \leqslant \bar{d}(u)/\lambda_n \leqslant m_+^2$ *and* $m_-^3 \leqslant \bar{s}(u)/\lambda_n \leqslant m_+^3$

(3) $m_- \leqslant \bar{d}_T/n\lambda_n \leqslant m_+$ *and* $m_-^2 \leqslant \bar{s}_T/n\lambda_n \leqslant m_+^2$

(4) $m_-^4/m_+^1 \leqslant r_{uv}(\bar{\mathbf{d}})/\rho_n \leqslant m_+^4/m_-^1$ *and* $m_-^6/m_+^2 \leqslant r_{uv}(\bar{\mathbf{s}})/\rho_n \leqslant m_+^6/m_-^2$

(5) $m_-^2/m_+^2 \leqslant f_{uv}(\phi, \psi) \leqslant m_+^2/m_-^2$ *and* $m_-^{10}/m_+^3 \leqslant f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant m_+^{10}/m_-^3$

(6) $0 \leqslant V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant (\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2$

(7) $0 \leqslant \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant g(\eta, m_-, m_+)$ *where g is a deterministic function.*

(8) There exist global constants $0 < m_1 < m_2 < \infty$ independent of n such that for any node set $B \subseteq [n]$,

$$m_1 |B| \rho_n \;\leqslant\; \mu(u, B|\bar{\mathbf{s}}),\; \sigma(u, B|\theta_*)^2 \;\leqslant\; m_2 |B| \rho_n$$

Proof. For (1), we have

$$\bar{s}(u) := \mathbb{E}S(u) = \sum_{j\in[K]}\sum_{v\in\mathcal{C}_j}\mathbb{E}W(u,v) = \sum_{j\in[K]}\sum_{v\in\mathcal{C}_j}\rho_n r_{uv}(\phi)\mathbf{H}[c(u),j]$$
$$= \rho_n\sum_{j\in[K]}\phi(u)n\tilde{\pi}_j\mathbf{H}[c(u),j] = \lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}},\mathbf{H}[c(u)]\rangle$$

An identical calculation yields the expression for $\bar{d}(u)$. The inequalities in (2) then follow from Assumption 6. For (3), we again apply Assumption 6 to the equation

$$\bar{s}_T = \sum_{i\in[K]}\sum_{v\in\mathcal{C}_i}\bar{s}(u) = \sum_{i\in[K]} n\lambda_n\phi(u)\langle\tilde{\boldsymbol{\pi}},\mathbf{H}[i]\rangle = n\lambda_n\tilde{\boldsymbol{\pi}}^T\mathbf{H}\tilde{\boldsymbol{\pi}}$$

An identical equation yields the inequality for $\bar{d}_T$. (2) and (3) directly yield the inequalities in (4). Note that Assumption 6 implies $m_-^2 \leqslant nr_{uv}(\phi), nr_{uv}(\psi) \leqslant m_+^2$, which yields the first inequality of (5). The second inequality of (5) follows from (4). For part (6), note that by Assumption 7 and the first inequality in (5), we have

$$W(u,v) := f_{uv}(\phi,\psi)\xi(u,v) \leqslant (m_+^2/m_-^2)\eta \tag{B.12}$$

The second inequality in (5) then yields (6). For part (7), recalling the definition of $\kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})$ from Definition 30, note first that, by (6), $0 \leqslant v_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}) \leqslant (\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2$. Thus, by the second inequality (5),

$$0 \leqslant \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}}) := \frac{\sum_{uv\in E} v_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})}{\sum_{uv\in E} f_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})} \leqslant \frac{(\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2}{m_-^{10}/m_+^3}$$

For part (8), recall that

$$\mu(u,B|\bar{\mathbf{s}}) := \sum_{v\in B} r_{uv}(\bar{\mathbf{s}})$$

The first inequality in (8) follows from applying the second inequality in (4). Similarly,

$$\sigma(u,B|\theta_*)^2 := \sum_{v\in B} r_{uv}(\bar{\mathbf{s}})f_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})(1-\tilde{r}_{uv}(\bar{\mathbf{d}}) + \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}}))$$

The second inequality in part (8) follows from parts (4), (5), and (7). ■

The next lemma shows that, if the degrees and strengths of $\mathcal{G}_n$ are bounded around their expected values, the empirical estimate of variance is bounded around the conditional population estimate, and the coefficient of variation of $S_n(u, B)$ is bounded around its population value.

Lemma 32. *Fix $n > 1$. Suppose Assumption 6 holds. Define*

$$M(\mathbf{D}, \mathbf{S}) := \max_{u \in [n]} \left\{ |S(u) - \bar{s}(u)|, |D(u) - \bar{d}(u)| \right\} \leqslant \lambda_n t. \tag{B.13}$$

Then the following statements hold:

(1) There exists small enough $t > 0$ such that if $M(\mathbf{D}, \mathbf{S}) \leqslant t$,

$$\left| \hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \right| = \left| \frac{\sum_{uv \in E} V_{uv}(\mathbf{D}, \mathbf{S}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_{uv \in E} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + |E| \rho_n O(t)} \right| + \rho_n O(t)$$

(2) Fix a constant $\varepsilon > 0$ independent of n. Assume $|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})| \leqslant \varepsilon$. Then then there exists small enough $t > 0$ (not depending on ε) such that if $M(\mathbf{D}, \mathbf{S}) \leqslant t$, for all $B \subseteq [n]$, we have

$$\left| \frac{\mu(u, B|\Theta)}{\sigma(u, B|\Theta)} - \frac{\mu(u, B|\theta_*)}{\sigma(u, B|\theta_*)} \right| = \sqrt{|B|\rho_n} O(t)$$

Proof. $M(\mathbf{D}, \mathbf{S}) \leqslant \lambda_n t$ implies there exists a n-vector $\mathbf{a}_t$ with components in the interval $[-1, 1]$ such that $S(u) = \bar{s}(u) + \lambda_n t a_t(u)$. Therefore, defining $\bar{a}_t := n^{-1} \sum_v a_t(v)$,

$$\begin{aligned}
r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}}) &= \frac{\{\bar{s}(u) + \lambda_n a_t(u) t\}\{\bar{s}(v) + \lambda_n a_t(v) t\}}{\bar{s}_T + n\lambda_n \bar{a}_t t} - \frac{\bar{s}(u)\bar{s}(v)}{\bar{s}_T} \\
&= \frac{\bar{s}_T\{\bar{s}(u) a_t(v) + \bar{s}(v) a_t(u) + \lambda_n a_t(u) a_t(v) t\}\lambda_n t - \bar{s}(u)\bar{s}(v) n \lambda_n \bar{a}_t t}{\bar{s}_T\{\bar{s}_T + n\lambda_n \bar{a}_t t\}} \\
&= \left\{ \frac{\bar{s}(u) a_t(v) + \bar{s}(v) a_t(u) + \lambda_n a_t(u) a_t(v) t - r_{uv}(\bar{s}) n \bar{a}_t}{\bar{s}_T + n\lambda_n \bar{a}_t t} \right\} \lambda_n t
\end{aligned}$$

Using parts (2)-(4) of Lemma 31, for sufficiently small t we have

$$\left| r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}}) \right| \leqslant \frac{2\lambda_n m_+^3 + \lambda_n t + \rho_n (m_+^6 / m_-^2) n}{n\lambda_n m_-^2 - n\lambda_n t} \lambda_n t = \frac{2m_+^3 + t + (m_+^6 / m_-^2)}{m_-^2 - t} \rho_n t$$

Therefore,

$$|r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}})| = \rho_n O(t) \tag{B.14}$$

as $t \to 0$. By a similar argument, $|r_{uv}(\mathbf{D}) - r_{uv}(\bar{\mathbf{d}})| = \rho_n O(t)$. It follows that

$$|f_{uv}(\mathbf{D}, \mathbf{S}) - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})| = \rho_n O(t). \tag{B.15}$$

Therefore, using Equations B.14-B.15 and part (7) of Lemma 31,

$$\begin{aligned}
V_{uv}(\mathbf{D}, \mathbf{S}) &:= (W(u,v) - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&= (W(u,v) - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}))^2 + 2(W(u,v) - f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}))(f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S})) \\
&+ (f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&= V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + 2V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})(f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S})) + (f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - f_{uv}(\mathbf{D}, \mathbf{S}))^2 \\
&\leqslant V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + \rho_n O(t) + \rho_n^2 O(t^2) = V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + \rho_n O(t)
\end{aligned}$$

Define $V_T := \sum_{uv \in E} V_{uv}(\mathbf{D}, \mathbf{S})$ and $\bar{V}_T := \sum_{uv \in E} V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})$. The above inequality implies that $V_T = \bar{V}_T + |E|\rho_n O(t)$. Define $g_T := \sum_{uv \in E} f_{uv}(\mathbf{D}, \mathbf{S})^2$ and $\bar{g}_T := \sum_{uv \in E} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2$. A similar bound gives $g_T = \bar{g}_T + |E|\rho_n O(t)$. Finally, define $\bar{v}_T := \sum_{uv \in E} v_{uv}(\bar{\mathbf{s}}, \bar{\mathbf{d}})$. Then

$$\begin{aligned}
\left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| &= \left|\frac{V_T}{g_T} - \frac{\bar{v}_T}{\bar{g}_T}\right| = \left|\frac{\bar{V}_T + |E|\rho_n O(t)}{\bar{g}_T + |E|\rho_n O(t)} - \frac{\bar{v}_T}{\bar{g}_T}\right| \\
&= \left|\frac{\bar{V}_T + |E|\rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T}\{\bar{g}_T + |E|\rho_n O(t)\}}{\bar{g}_T + |E|\rho_n O(t)}\right| \\
&\leqslant \left|\frac{\bar{V}_T - \bar{v}_T}{\bar{g}_T + |E|\rho_n O(t)}\right| + \left|\frac{|E|\rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T}|E|\rho_n O(t)}{\bar{g}_T + |E|\rho_n O(t)}\right|
\end{aligned}$$

Note that $\bar{v}_T(E)/|E|$ and $\bar{g}_T(E)/|E|$ are, each, by parts (5) and (6) of Lemma 31, bounded above and below by constants independent of E, t, and n. Therefore,

$$\left|\frac{|E|\rho_n O(t) - \frac{\bar{v}_T}{\bar{g}_T}|E|\rho_n O(t)}{\bar{g}_T + |E|\rho_n O(t)}\right| \leqslant \frac{\rho_n O(t)}{\bar{g}_T/|E| + \rho_n O(t)} = \rho_n O(t)$$

This proves part 1. For part 2, first recall that $\mu(u, B|\Theta) \equiv \mu(u, B|\mathbf{S}) := \sum_{v \in B} r_{uv}(\mathbf{S})$. Therefore by Equation B.14, we have

$$|\mu(u, B|\Theta) - \mu(u, B|\theta_*)| = \Big|\sum_{v \in B} r_{uv}(\mathbf{S}) - r_{uv}(\bar{\mathbf{s}})\Big| = |B|\rho_n O(t) \tag{B.16}$$

Recall further that

$$\sigma(u, B|\Theta)^2 := \sum_{v \in B} r_{uv}(\mathbf{S}) f_{uv}(\mathbf{D}, \mathbf{S}) \left(1 - r_{uv}(\mathbf{D}) + \hat{\kappa}(\mathbf{D}, \mathbf{S})\right)$$

Using some straightforward algebra and applying Equations B.14-B.15, we have

$$\begin{aligned} \left|\sigma(u, B|\Theta)^2 - \sigma(u, B|\theta_*)^2\right| &= |B| \left(1 + \left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right|\right) \rho_n O(t) \\ &= |B| \rho_n O(t) \end{aligned} \tag{B.17}$$

where the second line follows from the assumption that $|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})| \leqslant \varepsilon$. We will now bound $\sigma(u, B|\Theta)$ close to $\sigma(u, B|\theta_*)$ using Equation B.17 and a Taylor expansion. Define the function $h(x, \sigma) := \sqrt{\sigma^2 + x}$. For fixed σ, a Taylor expansion around $x = 0$ gives $h(x, \sigma) = \sigma + \sum_{k=1}^{\infty} (-1)^k \frac{x^k}{k! \sigma^{2k-1}}$. Setting $x = \sigma(u, B|\Theta)^2 - \sigma(u, B|\theta_*)^2$ and $\sigma = \sigma(u, B|\theta_*)$ and applying Equation B.17, we obtain

$$\begin{aligned} \sigma(u, B|\Theta) &= h(x, \sigma(u, B|\theta_*)) \\ &= \sigma(u, B|\theta_*) + \sum_{k=1}^{\infty} (-1)^k \frac{|B|^k \rho_n^k O(t^k)}{k! \sigma(u, B|\theta_*)^{2k-1}} \end{aligned} \tag{B.18}$$

Part (8) of Lemma 31 implies that $\sigma(u, B|\theta_*) \asymp \sqrt{|B|\rho_n}$. Equation B.18 therefore gives

$$\sigma(u, B|\Theta) = \sigma(u, B|\theta_*) + \sqrt{|B|\rho_n} O(t) \tag{B.19}$$

using Equations B.16 and B.19, we write

$$\left| \frac{\mu(u, B|\Theta)}{\sigma(u, B|\Theta)} - \frac{\mu(u, B|\theta_*)}{\sigma(u, B|\theta_*)} \right| = \left| \frac{\mu(u, B|\theta_*) + |B|\rho_n O(t)}{\sigma(u, B|\theta_*) + \sqrt{|B|\rho_n} O(t)} - \frac{\mu(u, B|\theta_*)}{\sigma(u, B|\theta_*)} \right| \tag{B.20}$$

As shorthands, define $\bar{\mu}_n := \mu(u, B|\theta_*)/|B|\rho_n$ and $\bar{\sigma}_n := \sigma(u, B|\theta_*)/\sqrt{|B|\rho_n}$. Part (8) of Lemma 31 implies that $\bar{\mu}_n, \bar{\sigma}_n \asymp 1$. Thus, using Equation B.20 and dividing through by the appropriate

factors,

$$\left|\frac{\mu(u,B|\Theta)}{\sigma(u,B|\Theta)} - \frac{\mu(u,B|\theta_*)}{\sigma(u,B|\theta_*)}\right| = \sqrt{|B|\rho_n}\left|\frac{\bar{\mu}_n + O(t)}{\bar{\sigma}_n + O(t)} - \frac{\bar{\mu}_n}{\bar{\sigma}_n}\right|$$
$$= \sqrt{|B|\rho_n}O(t)$$

This completes part 2. ■

The proof of Lemma 8 from the main text (below) makes use of Lemma 32 by showing that its assumption holds with high probability, for appropriate t.

B.7.1 Proof of Lemma 8 from the main text

Throughout, we will sometimes suppress dependence on n for notational convenience. Recall that $A(u,B|\mathbf{S}) := S(u,B) - \mu(u,B|\mathbf{S})$, the deviation of the CCME test statistic from its expected value under the continuous configuration model. Recalling that $\Theta := (\mathbf{D},\mathbf{S},\hat{\kappa}(\mathbf{D},\mathbf{S}))$, define also the random Z-statistic

$$Z(u,B|\Theta) := \frac{A(u,B|\mathbf{S})}{\sigma(u,B|\Theta)}. \tag{B.21}$$

Define the random p-value

$$P(u,B|\Theta) := 1 - \Phi(Z(u,B|\Theta)). \tag{B.22}$$

The random variable $P(u,B|\Theta)$ is the random version of the p-value $p(u,B_n|\theta)$ obtained from the approximation in Equation (3.12) of the main document. As a consequence of the Benjamini-Hochberg procedure, the event $\{U_\alpha(B_n,\mathcal{G}) = C_n\}$ will occur if

$$\begin{aligned} &P(u,B_n|\Theta) \leqslant q\alpha, \quad \text{for all } u \in C_n, \quad \text{and} \\ &P(u,B_n|\Theta) > q\alpha, \quad \text{for all } u \notin C_n, \end{aligned} \tag{B.23}$$

since by assumption $|C_n| > qn$. Let h be the density function of a standard-Normal. By a well-known inequality for the CDF of a standard-Normal, if $Z(u,B_n|\Theta) > 0$,

$$P(u,B_n|\Theta) \leqslant \frac{1}{Z(u,B_n|\Theta)}h(Z(u,B_n|\Theta)). \tag{B.24}$$

By symmetry, if $Z(u, B_n|\Theta) < 0$, then

$$P(u, B_n|\Theta) \geqslant 1 + \frac{1}{Z(u, B_n|\Theta)} h(Z(u, B_n|\Theta)). \tag{B.25}$$

We therefore analyze the concentration properties of $Z(u, B_n|\Theta)$ and apply Inequalities B.24 and B.25 to show that for sufficiently large n, the event in Equation B.23 occurs with high probability. We will focus on the first line of B.23 first; the second is shown similarly. For the derivation below we use the following shorthands: $Y \equiv S(u, B_n|\Theta)$, $\mu \equiv \mu(u, B_n|\mathbf{S})$, $\sigma := \sigma(u, B_n|\Theta)$, $\bar{y} \equiv \mathbb{E}Y$, $\bar{\mu} \equiv \mu(u, B_n|\theta_*)$, and $\bar{\sigma} := \sigma(u, B_n|\theta_*)$. Note

$$\begin{aligned} Z(u, B_n|\Theta) := \frac{Y - \mu}{\sigma} = \frac{Y - \bar{\mu}}{\bar{\sigma}} - \left(\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right) = \frac{\bar{y} - \bar{\mu}}{\bar{\sigma}} + \frac{Y - \bar{y}}{\bar{\sigma}} - \left(\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right) \\ \geqslant \frac{\bar{y} - \bar{\mu}}{\bar{\sigma}} - \left|\frac{Y - \bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \end{aligned} \tag{B.26}$$

Define

$$\bar{z}(u, B_n|\theta_*) := \frac{\bar{y} - \bar{\mu}}{\bar{\sigma}} = \lambda_n \frac{\tilde{a}(u, B_n|\bar{\mathbf{s}})}{\sigma(u, B_n|\theta_*)}$$

where $\tilde{a}(u, B_n|\bar{\mathbf{s}})$ is the normalized population version of $A(u, B_n|\mathbf{S})$, as defined in Equation 3.14 from the main text. The definition above works with Equation B.26 to produce the illustrative inequality

$$Z(u, B_n|\Theta) \geqslant \bar{z}(u, B_n|\theta_*) - \left|\frac{Y - \bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right|. \tag{B.27}$$

Inequality B.27 exemplifies that, if the right-hand terms vanish, $Z(u, B_n|\Theta)$ can be approximated by a population version. Our analysis therefore reduces to bounding the right-hand order terms in probability.

Explicitly, consider that by part (8) of Lemma 31, there exists $m_2 > 0$ such that $\sigma(u, B_n|\theta_*)^2 \leqslant m_2 n\rho_n = m_2\lambda_n$. Combining this with the crucial assumption on $\tilde{a}(u, B_n)$ from line 3.15 from the main text, we have that for all $u \in C_n$,

$$\bar{z}(u, B_n|\theta_*) = \lambda_n \frac{\tilde{a}(u, B_n|\bar{\mathbf{s}})}{\sigma(u, B_n|\theta_*)} \geqslant \sqrt{\lambda_n} \frac{\Delta}{\sqrt{m_2}} \tag{B.28}$$

Therefore, the rest of the proof is mainly dedicated to showing that the final two terms in line (B.27) are $o_P(\sqrt{\lambda_n})$. This will imply that $Z(u, B_n|\Theta) = \Omega_P(\sqrt{\lambda_n})$ and, using Inequality B.24, that

$\{P(u, B_n|\Theta) \leqslant q\alpha,\ \forall\ u \in C_n\}$ has probability approaching 1.

Step 1: $|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}| = O_P(\sqrt{\log n})$

For $t > 0$, define the event

$$\mathcal{E}_1(t) := \left\{ \max_{u\in[n]} |S(u) - \bar{s}(u)|, \max_{u\in[n]} \left|D(u) - \bar{d}(u)\right| \leqslant \lambda_n t \right\} \tag{B.29}$$

For $b > 0$, define $t_n(b) := \sqrt{\frac{\kappa \log n}{\lambda_n}}$. By part 1 of Lemma 32, the event $\mathcal{E}_1(t_n(b))$ implies

$$\left|\hat{\kappa}(\mathbf{D}, \mathbf{S}) - \kappa_*(\bar{\mathbf{d}}, \bar{\mathbf{s}})\right| = \left| \frac{\sum_{uv\in E} V_{uv}(\mathbf{D}, \mathbf{S}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_{uv\in E} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 + |E|\rho_n O(t)} \right| + \rho_n O(t_n(b)) \tag{B.30}$$

We analyze the right-hand side of Equation (B.30) when the edge set E is a fixed, arbitrary edge set E_o. By Lemma 31 (5)-(6), we have

$$0 \leqslant V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) \leqslant (\eta m_+^2/m_-^2 + m_+^{10}/m_-^3)^2 \quad \text{and} \quad \sum_{uv\in E_o} f_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})^2 \geqslant \frac{m_-^{10}}{m_+^3}|E_o|.$$

The edge weights $\{W(u, v) : \{u, v\} \in E_o\}$ are independent, so by Bernstein's Inequality,

$$\mathbb{P}\left(\left| \frac{\sum_{E_o} V_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}}, \bar{\mathbf{s}})}{\sum_{E_o} f_{uv}(\bar{\mathbf{s}}, \bar{\mathbf{d}})^2} \right| > \sqrt{\tfrac{\kappa \log n}{|E_o|}} \right) \leqslant 2 \exp\left\{ \frac{-2\kappa \log n}{2a_1 + \frac{2}{3}a_2\sqrt{\frac{\kappa \log n}{|E_o|}}} \right\} \tag{B.31}$$

where $a_1, a_2 > 0$ are constants depending only on m_+, m_-, and η. We now bound the size of the edge set E in probability. It is easily derivable from the statement of the WSBM and Assumption 6 that there exist constants a_3, a_4 depending on m_+ and m_- such that $\mathbb{E}|E| = a_3 n\lambda_n$ and $\mathrm{Var}|E| = a_4 n\lambda_n$. Therefore, by another application of Bernstein's Inequality,

$$\mathbb{P}\left(||E| - a_3 n\lambda_n| > \sqrt{n\lambda_n \kappa \log n} \right) \leqslant 2 \exp\left\{ \frac{-2\kappa \log n}{2a_4 + \frac{2}{3}\sqrt{\frac{\kappa \log n}{n\lambda_n}}} \right\} \tag{B.32}$$

Applying this to inequality (B.31), the law of total probability gives

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{\sum_E V_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}}) - v_{uv}(\bar{\mathbf{d}},\bar{\mathbf{s}})}{\sum_E f_{uv}(\bar{\mathbf{s}},\bar{\mathbf{d}})^2}\right| > \sqrt{\frac{\kappa\log n}{a_3 n\lambda_n - \sqrt{n\lambda_n\kappa\log n}}}\right) \\
&\leqslant 2\exp\left\{\frac{-2\kappa\log n}{2a_1 + \frac{2}{3}a_2\sqrt{\frac{\kappa\log n}{a_3 n\lambda_n - \sqrt{n\lambda_n\kappa\log n}}}}\right\} + 2\exp\left\{\frac{-2\kappa\log n}{2a_4 + \frac{2}{3}\sqrt{\frac{\kappa\log n}{n\lambda_n}}}\right\} = O(n^{-b}) \qquad \text{(B.33)}
\end{aligned}$$

for sufficiently large n. Along with Equation (B.30), this implies there exists a constant A_0 depending on parameter constraints such that

$$\mathbb{P}\left\{\left|\hat{\kappa}(\mathbf{D},\mathbf{S}) - \kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right| \leqslant A_0\left(\sqrt{\frac{b\log n}{n\lambda_n}} + \rho_n t_n(b)\right)\right\} \geqslant \mathbb{P}(\mathcal{E}_1(t_n(b))) - O(n^{-b}) \qquad \text{(B.34)}$$

for sufficiently large n. We now assess $\mathbb{P}(\mathcal{E}_1(t_n(b)))$. Note that for all $u \in [n]$, $\mathrm{Var}(S(u)) = O(\lambda_n)$. Furthermore, recall from Inequality B.12 (in the proof of Lemma 31) that $W(u,v) \leqslant m_+^2\eta/m_-^2$ for all $u, v \in [n]$. For fixed $b > 0$, Bernstein's Inequality therefore gives, for any $u \in [n]$,

$$\mathbb{P}\left(|S(u) - \bar{s}(u)| > \sqrt{\kappa\log n\lambda_n}\right) \leqslant 2\exp\left\{\frac{-2a_0\kappa\log n}{2 + \frac{2}{3}\sqrt{\frac{\kappa\log n}{\lambda_n}}}\right\}, \qquad \text{(B.35)}$$

where a_0 is a constant independent of n. The constant a_0 may be chosen so that, similarly,

$$\mathbb{P}\left(\left|D(u) - \bar{d}(u)\right| > \sqrt{\kappa\log n\lambda_n}\right) \leqslant 2\exp\left\{\frac{-2a_0\kappa\log n}{2 + \frac{2}{3}\sqrt{\frac{\kappa\log n}{\lambda_n}}}\right\} \qquad \text{(B.36)}$$

Applying a union bound, equations (B.35) and (B.36) give

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_1(t_n(b))) &\geqslant 1 - 2n\exp\left\{\frac{-2a_0\kappa\log n}{2 + \frac{2}{3}\sqrt{\frac{\kappa\log n}{\lambda_n}}}\right\} - 2n\exp\left\{\frac{-2a_0\kappa\log n}{2 + \frac{2}{3}\sqrt{\frac{\kappa\log n}{\lambda_n}}}\right\} \\
&= 1 - O(n^{-b+1}) \qquad \text{(B.37)}
\end{aligned}$$

for sufficiently large n. Returning to the inequality in (B.34), we therefore have

$$\mathbb{P}\left\{\left|\hat{\kappa}(\mathbf{D},\mathbf{S})-\kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right| \leqslant A_0\left(\sqrt{\tfrac{b\log n}{n\lambda_n}}+\rho_n t_n(b)\right)\right\} \geqslant \mathbb{P}(\mathcal{E}_1(t_n(b))) - O(n^{-b}) \geqslant 1-O(n^{-b+1}) \tag{B.38}$$

for sufficiently large n. Recall that by assumption, $\lambda_n/\log n \to \infty$. Thus $t_n(b)\to 0$, and

$$\sqrt{\tfrac{b\log n}{n\lambda_n}}+\rho_n t_n(b) = t_n(b)/\sqrt{n}+\rho_n t_n(b) \leqslant 1/\sqrt{n} = o(1).$$

Thus, Inequality B.38 implies that

$$\mathbb{P}\left(\left|\hat{\kappa}(\mathbf{D},\mathbf{S})-\kappa_*(\bar{\mathbf{d}},\bar{\mathbf{s}})\right| \leqslant \varepsilon\right) \geqslant 1-O(n^{-b+1}), \tag{B.39}$$

for sufficiently large n. For $\varepsilon>0$, define the event $\mathcal{E}_2(\varepsilon) := \left\{\left|\hat{\kappa}(\mathbf{D},\mathbf{S})-\kappa(\bar{\mathbf{d}},\bar{\mathbf{s}})\right| \leqslant \varepsilon\right\}$. By part 2 of Lemma 3, the event $\mathcal{E}_1(t_n(b))\cap\mathcal{E}_2(\varepsilon)$ implies

$$\begin{aligned}\left|\frac{\mu}{\sigma}-\frac{\bar{\mu}}{\bar{\sigma}}\right| := \left|\frac{\mu(u,B_n|\mathbf{S})}{\sigma(u,B_n|\Theta)}-\frac{\mu(u,B_n|\bar{\mathbf{s}})}{\sigma(u,B_n|\theta_*)}\right| &= \sqrt{|B_n|\rho_n}O(t_n(b)) \\ &\leqslant \sqrt{\lambda_n}O(t_n(b)). \\ &= O(\sqrt{\kappa\log n})\end{aligned} \tag{B.40}$$

Therefore, there exists a constant $A_2>0$ such that, by Inequalities B.37 and B.39,

$$\mathbb{P}\left(\left|\frac{\mu}{\sigma}-\frac{\bar{\mu}}{\bar{\sigma}}\right| \leqslant A_2\sqrt{\kappa\log n}\right) = 1-O(n^{-b+1}) \tag{B.41}$$

for sufficiently large n. This completes Step 1.

Step 2: $|\frac{Y-\bar{y}}{\bar{\sigma}}| = O_P(\sqrt{\log n})$.

Note that, as for Inequality B.35, Bernstein's Inequality gives

$$\mathbb{P}\left(|S(u,B_n) - \mathbb{E}S(u,B_n)| > \sqrt{\kappa \log n \lambda_n}\right) \leqslant 2\exp\left\{\frac{-2a_0\kappa\log n}{2+\frac{2}{3}\sqrt{\frac{\kappa\log n}{\lambda_n}}}\right\} \tag{B.42}$$

By Lemma 31 part (8), there exists $m_2 > 0$ such that $\sigma(u,B_n|\theta_*)^2 \leqslant m_2\lambda_n$. Thus,

$$\left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| := \left|\frac{S(u,B_n)-\mathbb{E}S(u,B_n)}{\sigma(u,B_n|\theta_*)}\right| \geqslant \left|\frac{S(u,B_n)-\mathbb{E}S(u,B_n)}{m_2\sqrt{\lambda_n}}\right|,$$

so by Inequality B.42, we have for sufficiently large n that

$$\mathbb{P}\left(\left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| \leqslant \sqrt{\frac{b\log n}{m_2}}\right) \geqslant 1 - O(n^{-b}). \tag{B.43}$$

This completes Step 2.

We now recall inequality B.27:

$$Z(u,B_n|\Theta) \geqslant \bar{z}(u,B_n|\theta_*) - \left|\frac{Y-\bar{y}}{\bar{\sigma}}\right| - \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right|.$$

In step 1, we showed that there exists a constant A_2 depending only on the fixed WSBM model parameters such that for any fixed $b > 1$, for large enough n, $\left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \leqslant A_2\sqrt{b\log n}$ with probability $1 - O(n^{-b+1})$. In step 2, we showed that there exists a constant m_2 depending only on the fixed WSBM model parameters such that for any fixed $b > 1$, for large enough n, $|\frac{Y-\bar{y}}{\bar{\sigma}}| \leqslant \sqrt{b\log n/m_2}$ with probability $1-O(n^{-b})$. Recall furthermore from inequality B.28 that $\bar{z}(u,B_n|\theta_*) \geqslant \Delta\sqrt{\lambda_n/m_2}$, where Δ is from condition 3.15 in the statement of the Theorem. We can therefore write that for any fixed $b > 1$, for sufficiently large n,

$$Z(u,B_n|\Theta) \geqslant \Delta\sqrt{\lambda_n/m_2} - \sqrt{b\log n/m_2} - A_2\sqrt{b\log n} = A_3\sqrt{\lambda_n} - A_4\sqrt{b\log n}$$

with probability at least $1-O(n^{-b+1})$. Now, by assumption, $|C_n| \geqslant qn$. Therefore, using Inequality B.24 and a union bound, we can write that for any fixed $b > 1$, for sufficiently large n,

$$\max_{u \in C_n} P(u, B_n|\Theta) \leqslant \exp\{-(A_3\sqrt{\lambda_n} - A_4\sqrt{b \log n})^2\} \tag{B.44}$$

with probability at least $1 - O(n^{-b+2})$. Note that for any fixed b, the right-hand-side of inequality B.44 vanishes, due to the assumption that $\lambda_n / \log n \to \infty$. Thus, for $b > 2$, inequality B.44 implies that for large enough n (now depending on choice of b), the event $\{P(u, B_n|\Theta) \leqslant q\alpha, \ \forall \ u \in C_n\}$ has probability $1 - O(n^{-b+2}) \to 1$.

It can be similarly shown that the second half of the event in (B.23) has probability approaching 1. Instead of Inequality B.27 we (similarly) derive

$$Z(u, B_n|\Theta) \leqslant \bar{z}(u, B_n|\theta_*) + \left|\frac{Y - \bar{y}}{\bar{\sigma}}\right| + \left|\frac{\mu}{\sigma} - \frac{\bar{\mu}}{\bar{\sigma}}\right| \tag{B.45}$$

This is useful because if $u \notin C_n$, assumption (3.15) ensures that $\tilde{a}_n(u, B_n|\bar{\mathbf{s}}) < -\Delta$, and hence

$$\bar{z}(u, B_n|\theta_*) := \frac{\bar{y} - \bar{\mu}}{\bar{\sigma}} = \lambda_n \frac{\tilde{a}(u, B_n|\bar{\mathbf{s}})}{\sigma(u, B_n|\theta_*)} \leqslant \lambda_n \frac{-\Delta}{\sigma(u, B_n|\theta_*)} \leqslant \sqrt{\lambda_n} \frac{-\Delta}{\sqrt{m}_1}$$

where the last inequality follows from part (8) of Lemma 31. Steps 1 and 2 therefore work to show that for any fixed $b > 1$, for large enough n,

$$Z(u, B_n|\Theta) \leqslant -\Delta\sqrt{\lambda_n/m_2} + \sqrt{b \log n/m_2} + A_2\sqrt{b \log n} = A_3\sqrt{\lambda_n} - A_4\sqrt{b \log n}$$

With probability $1 - O(n^{-b+1})$. Inequality B.25 then implies that

$$\mathbb{P}\left(\max_{u \notin C_n} P(u, B_n|\Theta) \geqslant 1 - \exp\{-(A_3\sqrt{\lambda_n} - A_4\sqrt{b \log n})^2\}\right) \geqslant 1 - O(n^{-b+2}) \tag{B.46}$$

With reasoning identical to the result for $u \in C_n$, this implies that for any $b > 2$, for large enough $n(b)$, the event $\{P(u, B_n|\Theta) > q\alpha, \ \forall \ u \notin C_n\}$ has probability at least $1 - O(n^{-b+2}) \to 1$. Applying a union bound to the event in (B.23) completes the proof. ■

B.7.2 Proof of Theorem 9 from the main text

We will show that if condition 3.16 holds, then condition 3.15 holds when $B_n = C_n = C_{j,n}$ simultaneously across $j \in [K]$. This involves representing condition 3.15 in terms of the model parameters when $B_n = C_n = C_{j,n}$. Specifically, we derive the normalized population deviation $\tilde{a}(u, C_{j,n}|\bar{\mathbf{s}}) := (\mathbb{E}S(u, C_{j,n}) - \mu(u, C_{j,n}|\bar{\mathbf{s}}))/\lambda_n$. First, note that for any fixed $j \in [K]$, part (1) of Lemma 31 gives

$$\sum_{v \in C_{j,n}} \bar{s}(v) \;=\; \lambda_n \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[j] \rangle \cdot \sum_{v \in C_{j,n}} \psi(u) \;=\; n\lambda_n \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[j] \rangle \tilde{\pi}_j$$

and thus

$$\bar{s}_T := \sum_{v \in [n]} \bar{s}(v) = \sum_{j \in [K]} \sum_{v \in C_{j,n}} \bar{s}(v) = n\lambda_n \sum_{j \in [K]} \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[j] \rangle \tilde{\pi}_j = n\lambda_n \tilde{\boldsymbol{\pi}}^t \mathbf{H} \boldsymbol{\pi}.$$

Therefore, again applying part (1) of Lemma 31,

$$\begin{aligned} \mu(u, C_{j,n}|\bar{\mathbf{s}}) := \sum_{v \in C_{j,n}} r_{uv}(\bar{\mathbf{s}}) = \bar{s}(u) \sum_{v \in C_{j,n}} \frac{\bar{s}(v)}{\bar{s}_T} &= \bar{s}(u) \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[j] \rangle \tilde{\pi}_j}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}} \\ &= \lambda_n \psi(u) \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[c(u)] \rangle \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[j] \rangle \tilde{\pi}_j}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}}. \end{aligned}$$

Secondly,

$$\mathbb{E}S(u, C_{j,n}) = \sum_{v \in C_{j,n}} \mathbb{E}W(u, v) = \sum_{v \in C_{j,n}} \rho_n r_{uv}(\psi) \mathbf{H}[c(u), j] = \lambda_n \psi(u) \mathbf{H}[c(u), j] \tilde{\pi}_j.$$

Thus,

$$\tilde{a}(u, C_{j,n}|\bar{\mathbf{s}}) := \frac{\mathbb{E}S(u, C_{j,n}) - \mu(u, C_{j,n}|\bar{\mathbf{s}})}{\lambda_n} = \psi(u) \tilde{\pi}_j \left(\mathbf{H}[c(u), j] - \frac{\langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[c(u)] \rangle \langle \tilde{\boldsymbol{\pi}}, \mathbf{H}[j] \rangle}{\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}} \right). \quad \text{(B.47)}$$

If $u \in C_{i,n}$, the expression in the parentheses from the right-hand-side of (B.47) is the i, j-th element of the matrix $\mathbf{H} - \mathbf{H}\tilde{\boldsymbol{\Pi}}\mathbf{H}/\tilde{\boldsymbol{\pi}}^t \mathbf{H} \tilde{\boldsymbol{\pi}}$, with $\tilde{\boldsymbol{\Pi}} := \tilde{\boldsymbol{\pi}} \tilde{\boldsymbol{\pi}}^t$. By Assumption 6, $\psi(u)\pi_i \geqslant m_-$ for all $u \in [n]$ and $i \in [K]$, and $\tilde{\pi}_j$ is fixed. Thus, condition (3.16) ensures that 3.15 holds when $C_n = C_{j,n}$, simultaneously across $j \in [K]$. Assumption 6 also ensures that there exists $q > 0$ such that for all

$j \in [K]$ and $n > 1$, $|C_{j,n}| > qn$. This allows us to apply Lemma 4 to the sequences $B_n = C_n = C_{j,n}$, for each $j \in [K]$. A union bound proves the result. ■

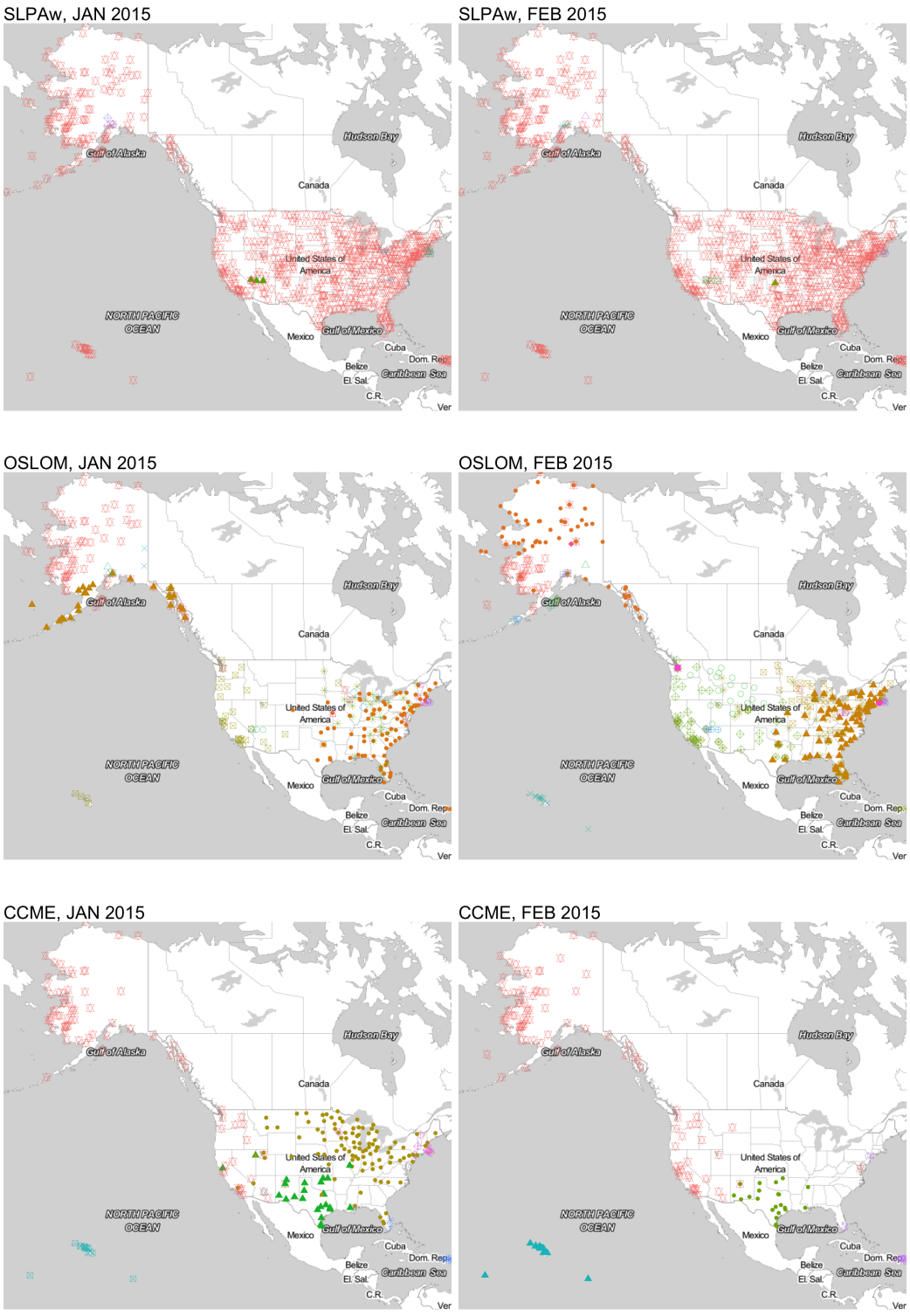

Figure B.3: SLPAw, OSLOM, and CCME results from January and Februrary 2015 U.S. airport networks. Maps created with `ggmap` (Kahle and Wickham, 2013)

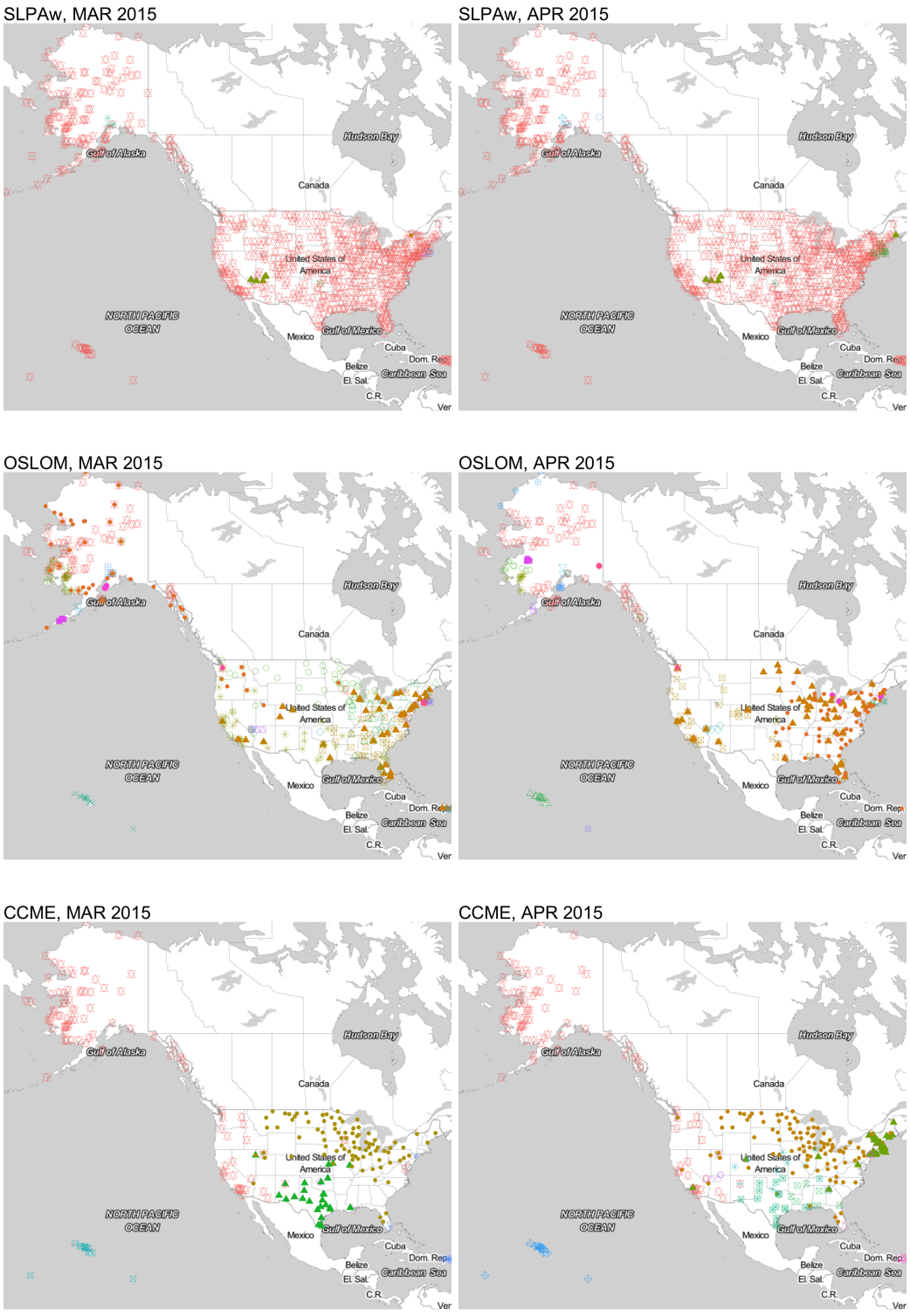

Figure B.4: SLPAw, OSLOM, and CCME results from March and April 2015 U.S. airport networks. Maps created with `ggmap` (Kahle and Wickham, 2013)

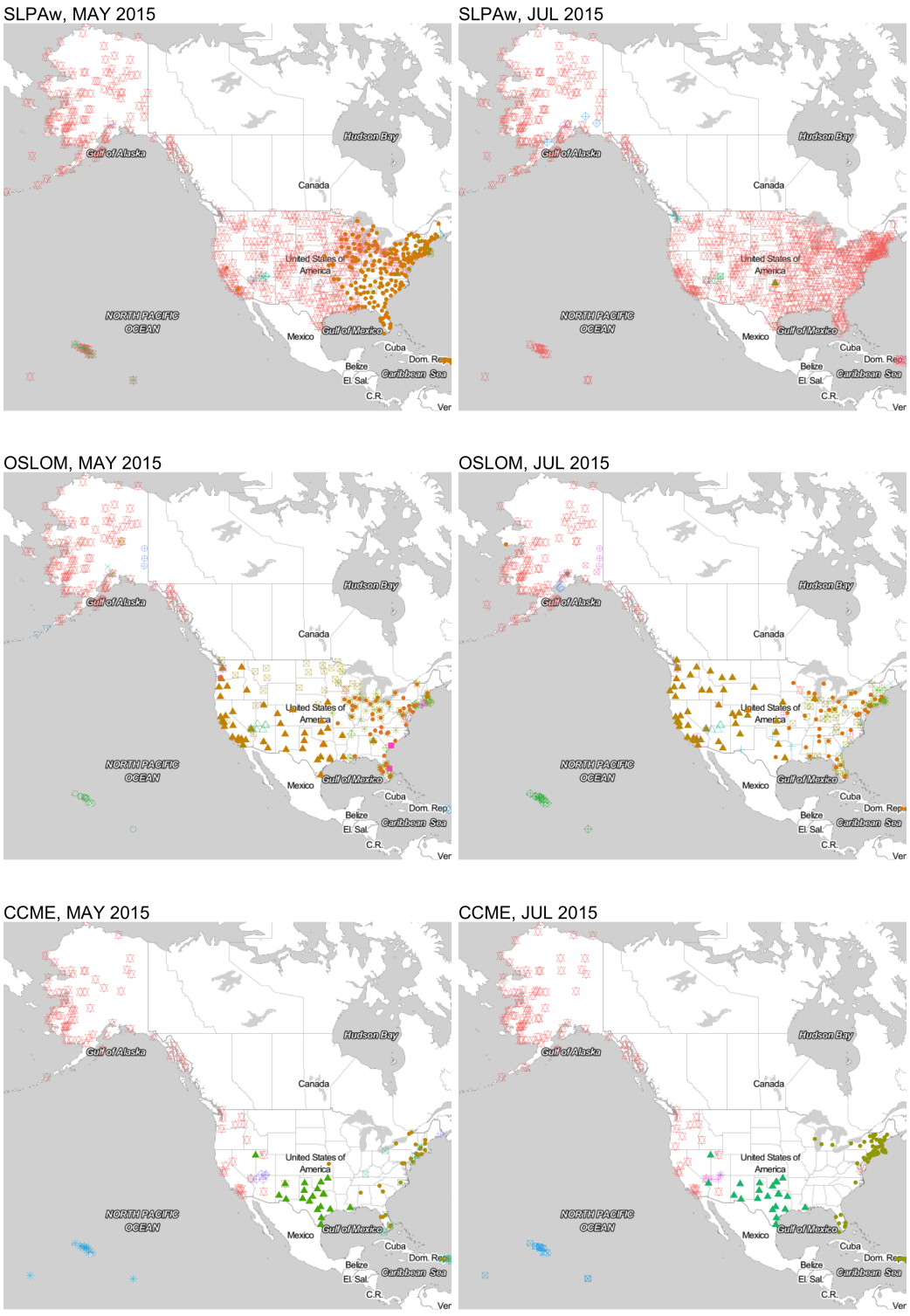

Figure B.5: SLPAw, OSLOM, and CCME results from May and July U.S. airport networks. Maps created with `ggmap` (Kahle and Wickham, 2013)

APPENDIX C

MULTI-LAYER EXTRACTION SUPPLEMENTAL

C.1 Proof of Lemma 15

It is easy to show that for any 2×2 symmetric matrix A and 2-vectors $\mathbf{x}$, $\mathbf{y}$,

$$(\mathbf{x}^T A\mathbf{x})(\mathbf{y}^T A\mathbf{y}) - (\mathbf{x}^T A\mathbf{y})^2 = (\mathbf{x}_1\mathbf{y}_2 - \mathbf{x}_2\mathbf{y}_1)^2 \det(A)$$

Fix $B \subseteq [n]$ and let s, ρ, and v correspond to B, as in Definition 13. Then for any $\ell \in [L]$, using the fact that $\kappa_\ell := \pi^T P_\ell \pi$ and the identity above, we have

$$\begin{aligned} v^t P_\ell v - \frac{(\pi^t P_\ell \pi)^2}{\kappa_\ell} &= \frac{\kappa_\ell v^t P_\ell v}{\kappa_\ell} - \frac{(v^t P_\ell \pi)^2}{\kappa_\ell} = \frac{(\pi^t P_\ell \pi)(v^t P_\ell v) - (v^t P_\ell \pi)^2}{\kappa_\ell} \\ &= (\pi_1(1-\rho) - \pi_2\rho)^2 \frac{\det P_\ell}{\kappa_\ell} = (\pi_1 - \rho)^2 \frac{\det P_\ell}{\kappa_\ell} \end{aligned}$$

Recall that $q_\ell(B) := \frac{s}{\sqrt{2}} \left(v^t P_\ell v - (\pi^t P_\ell \pi)^2/\kappa_\ell\right)$ and $H_*(B, L) = |L|^{-1} \left(\sum_\ell q_\ell(B)\right)^2$. Part 1 follows by summation over L. For part 2, note that $\pi_1^2 P_\ell(1,1) + \pi_2^2 P(2,2) \geqslant 2\pi_1\pi_2\sqrt{P_\ell(1,1)P_\ell(2,2)}$. Therefore,

$$\begin{aligned} \kappa_\ell &= \pi_1^2 P_\ell(1,1) + 2\pi_1\pi_2 P_\ell(1,2) + \pi_2^2 P(2,2) \\ &\geqslant 2\pi_1\pi_2 \left(\sqrt{P_\ell(1,1)P_\ell(2,2)} + P_\ell(1,2)\right) \\ &\geqslant 2\pi_1\pi_2 \left(\sqrt{P_\ell(1,1)P_\ell(2,2)} + P_\ell(1,2)\right)\left(\sqrt{P_\ell(1,1)P_\ell(2,2)} - P_\ell(1,2)\right) \\ &= 2\pi_1\pi_2\delta \geqslant \pi_1\delta \end{aligned}$$

Thus $\delta \leqslant \frac{\det P_\ell}{\kappa_\ell} \leqslant \frac{1}{\pi_1\delta}$. Part 2 follows. ■

C.2 Proof of Lemma 16

Define $g : 2^{[n]} \mapsto \mathbb{R}$ by $g(B) := \frac{s(B)}{2}(\rho(B) - \pi_1)^4$. Recall the function $\phi(L)$ defined in Lemma 15. Note that part 1 of Lemma 15 implies $H_*(B, L) = |L|\phi(L)g(B)$. It is therefore sufficient to show that there exists a constant $a > 0$ such that for sufficiently small t, $B \in \mathcal{R}(t)^c$ implies $g(B) < g(C_{1,n}) - at$. We will show this separately for the $\pi_1 < \pi_2$ and $\pi_1 = \pi_2$ cases.

Part 1 ($\pi_1 < \pi_2$): Define the intervals $I_1 := [0, \pi_1]$, $I_2 := (\pi_1, \pi_2]$, and $I_3 := (\pi_2, 1]$. We trisect $2^{[n]}$, the domain of g, with the collections $\mathcal{D}_{i,n} := \{B \subseteq [n] : s(B) \in I_i\}$, for $i = 1, 2, 3$. We will prove that the inequality $g(B) < g(C_{1,n}) - at$ holds for all $B \in \mathcal{R}(t)$ on each of those collections. We will continually rely on the fact that $B \in \mathcal{R}(t)$ implies at least one of the inequalities (I) $|s(B) - \pi_1| > t$ or (II) $1 - \rho(B) < t$ is true.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$ and inequality (I) is true. Then $s(B) < \pi_1 - t$, and

$$\begin{aligned} g(B) := \frac{s(B)^2}{2}(\rho(B) - \pi_1)^4 &\leqslant \frac{s(B)^2}{2}(1 - \pi_1)^4 \text{ (since } \pi_1 \leqslant 1/2) \\ &< \frac{(\pi_1 - t)^2}{2}(1 - \pi_1)^4 = \frac{\pi_1^2}{2}(1 - \pi_1)^4 - 2t(1 - \pi_1)^4 + o(t) \\ &< \frac{\pi_1^2}{2}(1 - \pi_1)^4 - t(1 - \pi_1)^4 = g(C_{1,n}) - t(1 - \pi_1)^4 \end{aligned} \tag{C.1}$$

for sufficiently small t. If inequality (II) is true, then

$$(\rho(B) - \pi_1)^4 \leqslant \max\{(1 - t - \pi_1)^4, \pi_1^4\} = \max\{(\pi_2 - t)^4, \pi_1^4\} = (\pi_2 - t)^4$$

for sufficiently small t, as $\pi_1 < \pi_2$. Therefore,

$$g(B) \leqslant \frac{\pi_1^4}{2}(\pi_2 - t)^4 = \frac{\pi_1^4}{2}\pi_2^4 - 4\pi_2^3 t + o(t) < g(C_{1,n}) - 2\pi_2^3 t \tag{C.2}$$

for sufficiently small t. Thus for all $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$, $g(B) < g(C_{1,n}) - a_1 t$ with $a_1 = \min\{(1 - \pi_1)^4, 2\pi_2^3\}$.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{2,n}$ and inequality (I) is true. Then $s(B) > \pi_1 + t$. Note that $0 \leqslant \rho(B)|B| \leqslant |C_{1,n}|$, yielding the useful inequality

$$0 \leqslant \rho(B) \leqslant \pi_1/s(B). \tag{C.3}$$

Subtracting through by π_1 gives

$$(\rho(B) - \pi_1)^4 \leqslant \max\{\pi_1^4, \pi_1^4(1/s(B) - 1)^4\} = \pi_1^4(1/s(B) - 1)^4.$$

Therefore,

$$g(B) \leqslant \frac{s(B)^2}{2}\pi_1^4(1/s(B) - 1)^4 = \frac{\pi_1^4}{2}(1/\sqrt{s(B)} - \sqrt{s(B)})^4 < \frac{\pi_1^4}{2}(1/\sqrt{\pi_1 + t} - \sqrt{\pi_1 + t})^4, \tag{C.4}$$

since $F(x) := (1/\sqrt{x} - \sqrt{x})^4$ is decreasing on $(0, 1]$, and $s(B) > \pi_1 + t$. Note that

$$\frac{d}{dt}\left(\frac{1}{\sqrt{\pi_1 + t}} - \sqrt{\pi_1 + t}\right)^4 = -3\left(\frac{1}{\sqrt{\pi_1 + t}} - \sqrt{\pi_1 + t}\right)^3\left[\frac{1}{2(\pi_1 + t)^{3/2}} + \frac{1}{2\sqrt{\pi_1 + t}}\right]. \tag{C.5}$$

By Taylor's theorem, this implies that

$$(1/\sqrt{\pi_1 + t} - \sqrt{\pi_1 + t})^4 = (1/\sqrt{\pi_1} - \sqrt{\pi_1})^4 - a_2 t + o(t) < (1/\sqrt{\pi_1} - \sqrt{\pi_1})^4 - a_2 t/2$$

for sufficiently small t, where a_2 is the right-hand-side of (C.5) at $t = 0$. Note further that $(1/\sqrt{\pi_1} - \sqrt{\pi_1})^4 = (\pi_2/\sqrt{\pi_1})^4 = \pi_2^4/\pi_1^2$. Putting these facts together with inequality (C.4), we obtain

$$g(B) < \frac{\pi_1^4}{2}\frac{\pi_2^4}{\pi_1^2} - a_2 t/2 = \frac{\pi_1}{2}\pi_2^4 - a_2 t/2 = g(C_{1,n}) - a_2 t/2 \tag{C.6}$$

If inequality (II) is true, $\rho(B) < 1 - t$. If $\rho(B) \leqslant \pi_1$, $(\rho(B) - \pi_1)^4$ is maximized when $\rho(B) = 0$, so that, since $s(B) \leqslant \pi_2$,

$$g(B) \leqslant \frac{\pi_2^2}{2}\pi_1^4 = g(C_{1,n}) + \frac{\pi_2^2}{2}\pi_1^4 - \frac{\pi_1^2}{2}\pi_2^4 = g(C_{1,n}) + \frac{\pi_1^2\pi_2^2}{2}(\pi_2^2 - \pi_1^2) < g(C_{1,n}) - t \tag{C.7}$$

for sufficiently small t, since π_1 is fixed. If $\rho(B) > \pi_1$, note that inequality (C.3) implies $s(B) \leqslant \pi_1/s(B)$. Therefore,

$$g(B) \leqslant \frac{\pi_1^2}{2\rho(B)^2}(\rho(B) - \pi_1)^4 = \frac{\pi_1^2}{2}(\sqrt{\rho(B)} - \pi_1/\sqrt{\rho(B)})^4 < \frac{\pi_1^2}{2}(\sqrt{1-t} - \pi_1/\sqrt{1-t})^4 \tag{C.8}$$

since $G(x) := (\sqrt{x} - \pi_1/\sqrt{x})^4$ is increasing on $(\pi_1, 1]$. A similar Taylor expansion argument to that yielding inequality (C.6) yields, for a constant a_3 depending only on π_1,

$$g(B) < \frac{\pi_1^2}{2}(1 - \pi_1)^4 - a_3 t/2 = g(C_{1,n}) - a_3 t/2, \tag{C.9}$$

for sufficiently small t. Pulling together inequalities (C.6), (C.7), and (C.9), we have that for all $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$, $g(B) < g(C_{1,n}) - a_4$ with $a_4 := \min\{a_2/2, 1, a_3/2\}$.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{3,n}$. Note that $|B| - |C_{2,n}| \leqslant |B \cap C_{1,n}| \leqslant |C_{1,n}|$. Dividing through by $|B|$ yields the useful inequality

$$1 - \pi_2/s(B) \leqslant \rho(B) \leqslant \pi_1/s(B). \tag{C.10}$$

Subtracting inequality (C.10) by π_1 gives

$$\pi_2(1 - 1/s(B)) \leqslant \rho(B) - \pi_1^4 \leqslant \pi_1(1/s(B) - 1).$$

Since $\pi_1 < \pi_2$, this implies that $(\rho(B) - \pi_1)^4 \leqslant \pi_2^4(1 - 1/s(B))^4$. Therefore,

$$g(B) \leqslant \frac{s(B)^2}{2}\pi_2^4(1/s(B) - 1)^4 = \frac{\pi_2^4}{2}(1/\sqrt{s(B)} - \sqrt{s(B)})^4 < \frac{\pi_2^4}{2}(1/\sqrt{\pi_2} - \sqrt{\pi_2})^4, \tag{C.11}$$

since $F(x) := (1\sqrt{x} - \sqrt{x})^4$ is decreasing on $I_3 := (\pi_2, 1]$ and $s(B) \in I_3$. Note that $\sqrt{\pi_2} - 1/\sqrt{\pi_2} = -\pi_1/\sqrt{\pi_2}$. Therefore,

$$g(B) < \frac{\pi_2^4}{2}\frac{\pi_1^4}{\pi_2^2} = \frac{\pi_2^2}{2}(0 - \pi_1)^4 = g(C_{2,n}) < g(C_{1,n}) - t$$

for t sufficiently small. Thus, for $a := \min\{a_1, a_4, 1\}$, for sufficiently small t we have $g(B) < g(C_{1,n}) - at$ whenever $B \in \mathcal{R}(t)$. This completes the proof in the case $\pi_1 < \pi_2$.

Part 2 ($\pi_1 = \pi_2$): Recall that when $\pi_1 = \pi_2$ we define $\mathcal{R}(t)$ by

$$\mathcal{R}(t) := \{B \subseteq [n] : |s(B) - \pi_1| \vee \rho(B) \vee [1 - \rho(B)] \leqslant t\}$$

Hence we will use the fact that $B \in \mathcal{R}(t)$ implies at least one of the inequalities (I) $|s(B) - \pi_1| > t$ or (II) $t < \rho(B) < 1 - t$ is true. Define the intervals $I_1 := [0, \pi_1]$, $I_2 := (\pi_1, 1]$. We bisect $2^{[n]}$, the domain of g, with the collections $\mathcal{D}_{i,n} := \{B \subseteq [n] : s(B) \in I_i\}$, for $i = 1, 2$. We will prove that the inequality $g(B) < g(C_{1,n}) - at$ holds for all $B \in \mathcal{R}(t)$ on each of those collections.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$ and inequality (I) is true. Then the same derivation yielding inequality (C.1) gives $g(B) < g(C_{1,n}) - t(1 - \pi_1)^4$ for sufficiently small t. If inequality (II) is true, then

$$(\rho(B) - \pi_1)^4 \leqslant \max\{(1 - t - \pi_1)^4, (\pi_1 - t)^4\} = \max\{(\pi_2 - t)^4, (\pi_1 - t)^4\} = (\pi_2 - t)^4,$$

since $\pi_1 = \pi_2$. Therefore, inequality (C.2) remains intact. Both inequalities hold on I_2 as well, for the roles of π_1 and π_2 may be interchanged, and the derivations treated symmetrically. This completes the proof in the case $\pi_1 = \pi_2$. ■

C.3 Proof of Lemma 18

Recall the definitions of set modularity and population set modularity from Definitions 4.2 and 4.9. Define $W := \sum_{\ell \in [L]} \widehat{Q}_\ell(B)$ and $w := \sum_{\ell \in [L]} \mathcal{Q}_\ell(B)$. Note that by Part 1 of Lemma 15, $q_\ell(B) \geqslant 0$ regardless of B, a fact which will allow the application of Lemma 35 in what follows. We have $\widehat{H}(B, L) = |L|^{-1} W_+^2$, $\mathcal{H}(B, L) = |L|^{-1} w^2$, and for any B such that $|B| \geqslant n\varepsilon$,

$$\mathbb{P}_n\left(\left|\widehat{H}(B, L) - \mathcal{H}(B, L)\right| > \frac{4|L|t}{n^2} + \frac{52|L|}{\kappa n}\right) = \mathbb{P}_n\left(\left|W_+^2 - w^2\right| > \frac{4|L|^2 t}{n^2} + \frac{52|L|^2}{\kappa n}\right)$$

$$\leqslant \mathbb{P}_n\left(\max_{\ell \in [L]} \left|\widehat{Q}_\ell(B) - \mathcal{Q}_\ell(B)\right| > \frac{t}{n^2} + \frac{13}{\kappa n}\right) \leqslant 4|L| \exp\left(-\kappa^2 \frac{\varepsilon t^2}{16 n^2}\right)$$

for large enough $t > 0$, where the first inequality follows from Lemma 35 for large enough n, and the second inequality follows from Lemma 38 and a union bound. Applying a union bound over sets $B \in \mathcal{B}_n$ yields the result. ■

C.4 Proof of Lemma 21

Assume first that $k > 1$. By definition, $B \in N_{n,k}(A)$ implies that at least one of $d_h(B, C_1) \leqslant A \cdot n \cdot b_{n,k-1}$ or $d_h(B, C_2) \leqslant A \cdot n \cdot b_{n,k-1}$ is true. Suppose the first inequality holds. Since $d_h(B, C_1) = |B \setminus C_1| + |C_1 \setminus B|$, we have the inequality

$$\big||B| - n\pi_1\big| = \big||B| - |C_1|\big| \leqslant \big||B| - |B \cap C_1| - |C_1| + |B \cap C_1|\big| \leqslant \big||B| - |B \cap C_1|\big| + \big||C_1| - |B \cap C_1|\big| = |B \setminus C_1| + |C_1 \setminus B| \leqslant A \cdot n \cdot b_{n,k-1}$$

Alternatively, if $d_h(B, C_2) \leqslant A \cdot n \cdot b_{n,k-1}$, we have the same bound for $\big||B| - n\pi_2\big|$. Therefore, since $\pi_1 \leqslant \pi_2$, $B \in N_{n,k}(A)$ implies that $|B| \geqslant n\pi_1 - A \cdot n \cdot b_{n,k-1}$. Since $b_{n,k-1} = o(1)$ as $n \to \infty$ and $\varepsilon < \pi_1$, this implies that for large enough n, $N_{n,k}(A) \subseteq \mathcal{B}_n(\varepsilon)$. By Lemma 18, therefore, for large enough n, we have

$$\mathbb{P}_n\left(\sup_{N_{n,k}(A)} \left|\widehat{H}(B, L) - \mathcal{H}(B, L)\right| > \frac{4|L|t}{n^2} + \frac{52|L|}{\kappa n}\right) \leqslant 4|L||N_{n,k}(A)| \exp\left(-\kappa^2 \frac{\varepsilon t^2}{16n^2}\right) \tag{C.12}$$

for all $t > 0$. We now bound the right-hand side of inequality (C.12) with t replaced by $t_n := n^{1+\frac{1}{2^k}} (\log n)^{1-\frac{1}{2^k}}$. Note that

$$\frac{t_n^2}{n^2} \;=\; \frac{1}{n^2} n^{2+\frac{1}{2^{k-1}}} (\log n)^{2-\frac{1}{2^{k-1}}} \;=\; n \cdot n^{\frac{1}{2^{k-1}}-1} (\log n)^{1-\frac{1}{2^{k-1}}} \log n \;=\; n \cdot b_{n,k-1} \log n.$$

Furthermore, by Corollary 34 (see Appendix C.5) we have $|N_{n,k}(A)| \leqslant 2\exp\left[3A \cdot n \cdot b_{n,k-1}\log\left(1/b_{n,k-1}\right)\right]$. These facts yield the bound

$$\begin{aligned}
|N_{n,k}(A)|\exp\left(-\kappa^2\frac{\varepsilon t_n^2}{16n^2}\right) &\leqslant 2\exp\left\{-\kappa^2\frac{\varepsilon}{16}n\cdot b_{n,k-1}\left[\log n - \frac{16}{\kappa^2\varepsilon}3A\log\left(1/b_{n,k-1}\right)\right]\right\} \\
&\leqslant 2\exp\left(-\kappa^2\frac{\varepsilon}{32}n\cdot b_{n,k-1}\log n\right) \text{ (for large } n\text{, since } 1/b_{n,k-1} = o(n)) \\
&< 2\exp\left(-\kappa^2\frac{\varepsilon}{32}n\gamma_n^{1-\varepsilon}\log n\right)
\end{aligned}$$

where the final inequality follows from the choice of k satisfying $\frac{1}{2^{k-1}} < \varepsilon$. Therefore,

$$4|L||N_{n,k}(A)|\exp\left(-\kappa^2\frac{\varepsilon t_n^2}{16n^2}\right) \leqslant 2\exp\left\{-\frac{\kappa^2\varepsilon}{32}n\gamma_n^{1-\varepsilon}\log n + O(\log|L|)\right\} \tag{C.13}$$

for large enough n. Notice now that $t_n/n^2 = b_{n,k}$ vanishes slower than $1/n$, and is therefore the leading order term in the expression $\frac{4|L|t_n}{n^2} + \frac{52|L|}{\kappa n}$ (see equation C.12). Hence for large enough n we have $\frac{4|L|t_n}{n^2} + \frac{52|L|}{\kappa n} \leqslant 5|L|b_{n,k}$. Combining this observation with lines (C.12) and (C.13) proves the result in the case $k > 1$.

If $k = 1$, assume $A = \varepsilon$. By definition, then (see Definition 20), $N_{n,k}(A) = \mathcal{B}_n(\varepsilon)$. Returning to inequality (C.12), we note that $\log|\mathcal{B}_n(\varepsilon)| = O(n)$, and thus we can derive the bound (C.13) with the same choice of $t_n := n^{1+\frac{1}{2^k}}(\log n)^{1-\frac{1}{2^k}} = n\sqrt{n\log n}$. The rest of the argument goes through unaltered. ■

C.5 Technical Results

Lemma 33. *Fix $\pi_1 \in [0,1]$. For each n, let $C_1 \subseteq [n]$ be an index set of size $\lfloor n\pi_1\rfloor$. Let $C_2 := [n]\backslash C_1$. Let $\gamma_n \in [0,1]$ be a sequence such that $\gamma_n \to 0$ and $n\gamma_n \to \infty$. Then for large enough n,*

$$|N(C_1,\gamma_n)| \leqslant \exp\{3n\gamma_n\log(1/\gamma_n)\}$$

Proof. Define the boundary of a neighborhood of $C \subseteq [n]$ by

$$\partial N(C,r) := \{B \subseteq [n] : d_h(B,C) = \lfloor nr\rfloor\}.$$

Note that any $B \subseteq [n]$ may be written as the disjoint union $B = \{C_2 \cap B\} \cup \{C_1 \cap B\}$. Since $C_1 \cap B = C_1 \setminus \{C_1 \setminus B\}$, for fixed $k \in [n]$ it follows that each set $B \in \partial N(C, k/n)$ is uniquely identified with choices of $|C_2 \cap B|$ indices from C_2 and $|C_1 \setminus B|$ indices from C_1 such that

$$|B \cap C_2| + |C_1 \setminus B| = |B \setminus C_1| + |C_1 \setminus B| = d_h(B, C_1) = k$$

Therefore, we have the equality

$$|\partial N(C_1, k)| \;=\; \sum_{m=0}^{k} \left[\binom{|C_2|}{m} + \binom{|C_1|}{k-m} \right] \tag{C.14}$$

Note that for positive integers K, N with $K < N/2$, properties of the geometric series yield the following bound:

$$\begin{aligned} \binom{N}{K}^{-1} \sum_{m=0}^{K} \binom{N}{m} = \sum_{m=0}^{K} \frac{(N-K)!K!}{(N-m)!m!} &= \sum_{m=0}^{K} \prod_{j=m+1}^{K} \frac{j}{N-j+1} \\ &< \sum_{m=0}^{K} \left(\frac{K}{N-K+1} \right)^m < \frac{N-(K-1)}{N-(2K-1)} \end{aligned} \tag{C.15}$$

For sufficiently small K/N, the right-hand side of inequality (C.15) is less than 2, and thus $\sum_{m=0}^{K} \binom{N}{m} < 2\binom{N}{K}$ if $K \ll N$. We apply this inequality to equation (C.14). Choose n large enough so that $n\gamma_n < \frac{1}{2}\min\{|C_1|, |C_2|\}$, which is possible since $\gamma_n \to 0$. Then for fixed $k \leqslant n\gamma_n$, we have that $|\partial N(C_1, k)| < 2\left[\binom{|C_2|}{k} + \binom{|C_1|}{k}\right]$ for large enough n. By another application of the inequality derived from (C.15), using the fact that $n\gamma_n = o(n)$, we therefore obtain

$$\begin{aligned} |N(C_1, \gamma_n)| = \sum_{k=0}^{\lfloor n\gamma_n \rfloor} |\partial N(C_1, k)| &< \sum_{k=0}^{\lfloor n\gamma_n \rfloor} 2\left[\binom{|C_2|}{k} + \binom{|C_1|}{k} \right] \\ &< 4\left[\binom{|C_2|}{\lfloor n\gamma_n \rfloor} + \binom{|C_1|}{\lfloor n\gamma_n \rfloor} \right] \leqslant 8\binom{n}{\lfloor n\gamma_n \rfloor} \end{aligned}$$

As $\binom{N}{K} \leqslant \left(\frac{N \cdot e}{K}\right)^K$, we have

$$|N(C_1, \gamma_n)| \leqslant \exp\left\{\log(8) + n\gamma_n \left[\log(e) + \log(1/\gamma_n)\right]\right\} \leqslant \exp\{3n\gamma_n \log(1/\gamma_n)\}$$

for large enough n, since $1/\gamma_n \to \infty$. ■

Here we give a short Corollary to Lemma 33 which directly serves the proof of Lemma 21. Recall $N_{n,k}(A)$ from Definition 20 in Section 4.2.3.3.

Corollary 34. *Fix an integer $k > 1$. For large enough n,*

$$|N_{n,k}(A)| \leqslant 2\exp\left[3A \cdot n \cdot b_{n,k-1}\log\left(1/b_{n,k-1}\right)\right]$$

Proof. The corollary follows from a direct application of Lemma 33 to $N(C_1, A\cdot b_{n,k-1})$ and $N(C_2, A\cdot b_{n,k-1})$. ■

Lemma 35. *Let $x_1, \ldots, x_k \in (0,1)$ be fixed and let $X_1, \ldots, X_k$ be arbitrary random variables. Define $W := \sum_i X_i$ and $w := \sum_i x_i$. Then for t sufficiently small, $\mathbb{P}(|W_+^2 - w^2| > 4k^2t) \leqslant \mathbb{P}(\max_i |X_i - x_i| > t)$.*

Proof. Define $D_i := |X_i - x_i|$ and fix $t < \min_i x_i$. Then if $\max_i D_i \leqslant t$, all X_i's will be positive, and thus $W_+ = W$ and $|W - w| \leqslant kt$, by the triangle inequality. Therefore $\max_i D_i \leqslant t$ implies that

$$|W_+^2 - w^2| = |(W-w)^2 + 2w(W-w)| \leqslant k^2t^2 + 2wkt \leqslant k^2t^2 + 2k^2t \tag{C.16}$$

Thus by the law of total probability, we have

$$\mathbb{P}(|W_+^2 - w^2| > 4k^2t) \leqslant \mathbb{P}(\{|W_+^2 - w^2| > 4k^2t\} \cap \{\max_i D_i \leqslant t\}) + \mathbb{P}(\max_i D_i > t)$$

Inequality (C.16) implies that for sufficiently small t, the first probability on the right-hand side above is equal to 0. The result follows. ■

In what follows we state and prove Lemma 38, a concentration inequality for the modularity of a node set (see Definition 4.2) from a single-layer SBM with n nodes and two communities. We first give a few short facts about the 2-community SBM. For for all results that follow, let s, ρ, and v (see Definition 13) correspond to the fixed set $B \subseteq [n]$ in each result (though sometimes we will make explicit the dependence on B). Define a matrix V by $V(i,j) := P(i,j)(1 - P(i,j))$ for $i = 1, 2$, where P is the probability matrix associated with the 2-block SBM.

Lemma 36. *Consider a single-layer SBM with $n > 1$ nodes, two communities, and parameters P and π_1. Fix a node set $B \subseteq [n]$ with $|B| \geqslant \alpha n$ for some $\alpha \in (0,1)$. Then*

1. $\left|\mathbb{E}[Y(B)] - \binom{|B|}{2} v^t P v\right| \leqslant 3|B|/2$
2. $\left|\mathbb{E}\left[\sum_{u\in B} \widehat{d}(u)\right] - |B| n v^t P \pi\right| \leqslant |B|$
3. $Var\left[\sum_{u\in B} \widehat{d}(u)\right] \leqslant 9|B|n$

Proof. For part 1, note that by definition,

$$\mathbb{E}\left[Y(B)\right] = \sum_{u,v\in B: u<v} \mathbb{P}\left((u,v) \in \widehat{E}\right) = \frac{1}{2} \sum_{u\neq v: u,v\in B} \mathbb{P}\left((u,v) \in \widehat{E}\right)$$

The right-hand sum can be expressed the sum of the entries of a 2×2 symmetric block matrix with zeroes on the diagonal. In this matrix, the upper diagonal block is of size $|B \cap C_1|$ with off-diagonal entries equal to $P(1,1)$. The lower diagonal block is of size $|B \cap C_2|$ with off-diagonal entries equal to $P(2,2)$. The off-diagonal blocks have entries equal to $P(1,2)$. Therefore, summing over blocks and accounting for the zero diagonal, we have

$$\begin{aligned}\mathbb{E}\left[Y(B)\right] = \frac{1}{2}\Big[|B \cap C_1|^2 P(1,1) + |B \cap C_1||B \cap C_2| P(1,2) + |B \cap C_2|^2 P(2,2)\Big] \\ - \frac{1}{2}\Big[|B \cap C_1| P(1,1) + |B \cap C_2| P(2,2)\Big]\end{aligned}$$

By dividing and multiplying by $|B|^2$ and collapsing cross-products, we get

$$\begin{aligned}\mathbb{E}\left[Y(B)\right] &= \frac{|B|^2}{2}\left[v^t P v - \frac{\rho P(1,1) + (1-\rho) P(2,2)}{|B|}\right] \\ &= \binom{|B|}{2}\left[1 + \frac{1}{|B|-1}\right]\left[v^t P v - \frac{\rho P(1,1) + (1-\rho) P(2,2)}{|B|}\right] \\ &= \binom{|B|}{2}\left[v^t P v - \frac{\rho P(1,1) + (1-\rho) P(1,2)}{|B|} + \frac{v^t P v}{|B|-1} - \frac{\rho P(1,1) + (1-\rho) P(2,2)}{|B|(|B|-1)}\right]\end{aligned}$$

Part 1 follows by carrying out the multiplication by $\binom{|B|}{2}$ in the last expression.

For part 2, let $P(\cdot, i)$ denote the i-th column of P. Note that $\mathbb{E}\left[\widehat{d}(u)\right] = n\pi^T P(\cdot, c_u) - P(c_u, c_u)$, with $c_u \in \{1, 2\}$ denoting the community index of u. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\sum_{u\in B} \widehat{d}(u)\right] &= \sum_{u\in B} \mathbb{E}\left[\widehat{d}(u)\right] = \sum_{u\in B\cap C_1} \mathbb{E}\left[\widehat{d}(u)\right] + \sum_{u\in B\cap C_2} \mathbb{E}\left[\widehat{d}(u)\right] \\
&= |B\cap C_1|\left[n\pi^T P(\cdot,1) - P(1,1)\right] + |B\cap C_2|\left[n\pi^T P(\cdot,2) - P(2,2)\right] \\
&= |B|\left[n\rho\pi^T P(\cdot,1) + n(1-\rho)\pi^T P(\cdot,2) - \rho P(1,1) - (1-\rho)P(2,2)\right] \\
&= |B| n v^t P\pi - |B|\big[\rho P(1,1) + (1-\rho)P(2,2)\big]
\end{aligned}$$

which completes part 2.

Finally, for part 3, we have

$$\operatorname{Var}\left[\sum_{u\in B} \widehat{d}(u)\right] = \operatorname{Var}\left[2Y(B)\right] + \sum_{u,v:u\in B, v\in B^C} \operatorname{Var}\left[\widehat{X}(u,v)\right]. \tag{C.17}$$

We address these two terms separately. For the first term, a calculation analogous to that from part 1 yields that $|\operatorname{Var}[Y(B)] - \binom{|B|}{2} v^t V v| \leqslant 3|B|/2$. Defining $\bar{v} := (\rho(B^C), 1-\rho(B^C))^t$, it is easy to show that $\sum_{u,v:u\in B, v\in B^C} \operatorname{Var}\left[\widehat{X}(u,v)\right] = |B||B^C| v^t V \bar{v}$, which is simply the sum of variances of all edge indicators for edges from B to B^C. Applying these observations to equation (C.17), we have

$$\begin{aligned}
\operatorname{Var}\left[\sum_{u\in B} \widehat{d}(u)\right] &\leqslant 4\binom{|B|}{2} v^t V v + 12|B|/2 + |B||B^C| v^T V \bar{v} \\
&\leqslant |B|\big[2(|B|-1) v^t V v + 6 + |B^C| v^t V \bar{v}\big] \leqslant 9|B| n
\end{aligned}$$

■

Lemma 37. *Under a single-layer SBM with $n > 1$ nodes, two communities, and parameters P and π_1, define $\kappa := \pi^T P \pi$. Then for large enough n, $\mathbb{P}\left(\left|2|\widehat{E}| - n^2\kappa\right| > t + 4n\right) \leqslant 2\exp\left\{-\frac{t^2}{n^2}\right\}$ for any $t > 0$.*

Proof. Note that $|\widehat{E}| = Y([n])$. Thus part 1 of Lemma 36 with $B = [n]$ yields $\left|\mathbb{E}[|\widehat{E}|] - \binom{n}{2}\kappa\right| \leqslant 3n/2$ for large enough n. As $n^2/2 = \binom{n}{2} + n/2$, by the triangle inequality,

$$\left|\mathbb{E}[|\widehat{E}|] - \frac{n^2}{2}\kappa\right| \leqslant \left|\mathbb{E}[|\widehat{E}|] - \binom{n}{2}\kappa\right| + \frac{n}{2} \leqslant 2n$$

Thus for any $t > 0$, Hoeffding's inequality gives

$$\begin{aligned}\mathbb{P}\left(\left|2\widehat{E} - n^2\kappa\right| > t + 4n\right) &\leqslant \mathbb{P}\left(\left|2\widehat{E} - n^2\kappa\right| > t + 2\left|\mathbb{E}[|\widehat{E}|] - \frac{n^2}{2}\kappa\right|\right) \\ &\leqslant \mathbb{P}\left(\left|2\widehat{E} - 2\mathbb{E}[|\widehat{E}|]\right| > t\right) \leqslant 2\exp\left\{-2\frac{t^2}{4\binom{n}{2}}\right\} \leqslant 2\exp\{-t^2/n^2\}\end{aligned}$$

■

Lemma 38. *Consider a single-layer 2-block SBM having $n > 1$ nodes and parameters P and π. Fix $\alpha \in (0, 1)$ and $B \subseteq [n]$ such that $|B| \geqslant \alpha n$. Then for large enough n we have*

$$\mathbb{P}_n\left(\left|\widehat{Q}(B) - \mathcal{Q}(B)\right| > \frac{t}{n^2} + \frac{8}{\kappa n}\right) \leqslant 4\exp\left(-\frac{\kappa^2\alpha t^2}{16n^2}\right) \tag{C.18}$$

for any $t > 0$.

Proof. With notation laid out in Section 4.1.2, define

$$\widetilde{Q}(B) := n^{-1}\binom{|B|}{2}^{-1/2}(Y(B) - \widetilde{\mu}(B)) \tag{C.19}$$

where

$$\widetilde{\mu}(B) := \frac{\sum_{u,v\in B:u<v}\widehat{d}(u)\widehat{d}(v)}{n^2\kappa} \tag{C.20}$$

We will prove the inequality in three steps: *Step 1*: bounding $\left|\widehat{Q}(B) - \widetilde{Q}(B)\right|$ in probability; *Step 2*: deriving a concentration inequality for $\widetilde{Q}(B)$; and *Step 3*: showing that $\left|\mathbb{E}\left[\widetilde{Q}(B)\right] - \mathcal{Q}(B)\right|$ is eventually bounded by a constant.

Step 1. As $\sum_{u,v\in B;u<v}\widehat{d}(u)\widehat{d}(v)\leqslant\sum_{u\in B}\widehat{d}(u)^2$, we have

$$\sum_{u,v\in B;u<v}\widehat{d}(u)\widehat{d}(v)\leqslant\sqrt{\sum_{u,v\in B;u<v}\widehat{d}(u)\widehat{d}(v)}\sqrt{\sum_{u\in B}\widehat{d}(u)^2}\leqslant n\binom{|B|}{2}^{1/2}|\widehat{E}|$$

Therefore,

$$\left|\widehat{Q}(B)-\widetilde{Q}(B)\right|=n^{-1}\binom{|B|}{2}^{-1/2}\left|\frac{(2|\widehat{E}|-n^2\kappa)\sum_{u,v\in B;u<v}\widehat{d}(u)\widehat{d}(v)}{2|\widehat{E}|n^2\kappa}\right|\leqslant\frac{\left|2|\widehat{E}|-n^2\kappa\right|}{2n^2\kappa}\qquad\text{(C.21)}$$

Combining the inequality in (C.21) with Lemma 37, for any $t>0$,

$$\mathbb{P}\left(\left|\widehat{Q}(B)-\widetilde{Q}(B)\right|>\frac{t}{2n^2}+\frac{2}{\kappa n}\right)\leqslant\mathbb{P}\left(\left|2|\widehat{E}|-n^2\kappa\right|>\kappa t+4n\right)\quad\leqslant 2\exp\left(-\frac{\kappa^2t^2}{n^2}\right).\qquad\text{(C.22)}$$

Step 2. This step relies on McDiarmid's concentration inequality. Recall from Section 4.1.1 that $\widehat{X}(u,v)$ denotes the indicator of edge presence between nodes u and v. Note that node pairs have a natural, unique ordering along the upper-diagonal of the adjacency matrix. Define $ord\{u,v\}=2(u-1)+(v-1)$, for $\{u,v\}\in[n]^2$ with $u<v$ (e.g. $ord\{1,2\}=1$, $ord\{1,3\}=2$, etc.). For all $n>1$ and $i\leqslant n(n-1)/2$, define $\widehat{Z}(i):=\widehat{X}(u,v)$ such that $ord\{u,v\}=i$. If $ord\{u,v\}=i$, we call $\{u,v\}$ the "i-th ordered node pair". Define the set

$$\mathcal{I}(B):=\{i:\text{the }i\text{-th ordered node pair has at least one node in }B\}$$

and let $\widehat{\mathcal{Z}}(B):=\{\widehat{Z}(i):i\in I(B)\}$. Note that the proxy score $\widetilde{Q}(B)$ is a function $f(z_1,z_2,\ldots)$ of the indicators $\widehat{\mathcal{Z}}(B)$.

Consider a *fixed* indicator set $\mathcal{Z}(B)$. For each $j\in I(B)$, define $\mathcal{Z}^j(B):=\{Z^j(i):i\in I(B)\}$ with

$$Z^j(B):=\begin{cases}Z^j(i)=1-Z(i), & i=j\\ Z^j(i)=Z(i), & i\neq j\end{cases}\qquad\text{(C.23)}$$

To apply McDiarmid's inequality, we must bound $\Delta(j):=|f(\mathcal{Z}(B))-f(\mathcal{Z}^j(B))|$ uniformly over $j\in\mathcal{I}(B)$. Fix $j\in\mathcal{I}(B)$ and let $\{u',v'\}$ be the j-th ordered edge. Without loss of generality, we assume $Z(j)=1$. Since $f(\mathcal{Z}(B))=Q(B)$, $f(\mathcal{Z}(B))$ has a representation in terms of $Y(B)$ and

$\widetilde{\mu}(B)$. We let $Y^j(B)$ and $\widetilde{\mu}^j(B)$ correspond to $f(\mathcal{Z}(B)^j)$. Notice that

$$n\binom{|B|}{2}^{1/2}\Delta(j) = \left|Y(B) - Y^j(B) - \left[\widetilde{\mu}(B) - \widetilde{\mu}^j(B)\right]\right| \tag{C.24}$$

We bound the right hand side of equation (C.24) in two cases: (i) $u', v' \in B$, and (ii) $u' \notin B, v' \in B$. In case (i), $Y(B) - Y^j(B) = 1$, and

$$\begin{aligned}\widetilde{\mu}(B) - \widetilde{\mu}^i(B) &= \frac{\sum_{u,v\in B; u\neq v} d(u)d(v) - d^j(u)d^j(v)}{n^2\kappa} = \frac{d(u')d(v') - d^j(u')d^j(v')}{n^2\kappa} \\ &= \frac{d(u')d(v') - (d(u')-1)(d(v')-1)}{n^2\kappa} = \frac{d(u') + d(v') - 1}{n^2\kappa},\end{aligned}$$

which is bounded in the interval $(0, 1)$ for large enough n. Thus in case (i), $\Delta(j) \leqslant 2\binom{|B|}{2}^{-1/2}$ by the triangle inequality, for large enough n. In case (ii), $Y(B) - Y'(B) = 0$, and

$$\begin{aligned}\widetilde{\mu}(B) - \widetilde{\mu}^j(B) &= \frac{\sum_{u,v\in B; u\neq v} d(u)d(v) - d^j(u)d^j(v)}{n^2\kappa} = \frac{\sum_{u\in B; u\neq v'} d(u)\left[d(v') - d^j(v')\right]}{n^2\kappa} \\ &= \frac{\sum_{u\in B; u\neq v'} d(u)}{n^2\kappa} \leqslant \frac{n|B|}{n^2\kappa} \leqslant \kappa^{-1}\end{aligned}$$

Hence due to equation (C.24), we have for sufficiently large n that

$$\Delta(j) \leqslant n^{-1}\binom{|B|}{2}^{-1/2} \cdot \max\{2, \kappa^{-1}\} \leqslant n^{-1}\binom{|B|}{2}^{-1/2} \cdot 2 \cdot \kappa^{-1} \tag{C.25}$$

for all $j \in \mathcal{I}(B)$, as $\kappa \leqslant 1$. Since $|\mathcal{I}(B)| = \binom{|B|}{2} + |B||B^C| \leqslant n|B|$, McDiarmid's bounded-difference inequality implies that for sufficiently large n,

$$\begin{aligned}\mathbb{P}\left(\left|\widetilde{Q}(B) - \mathbb{E}\left[\widetilde{Q}(B)\right]\right| > \frac{t}{n}\right) &= 2\exp\left(\frac{-t^2}{n|B|\Delta(j)}\right) \leqslant 2\exp\left(-\kappa^2\frac{n^2\binom{|B|}{2}t^2}{4n^3|B|}\right) \\ &\leqslant 2\exp\left(-\kappa^2\frac{(|B|-1)t^2}{8n}\right) \leqslant 2\exp\left(-\kappa^2\frac{\alpha t^2}{16}\right)\end{aligned}$$

for any $t > 0$. Replacing t by t/n gives

$$\mathbb{P}\left(\left|\widetilde{Q}(B) - \mathbb{E}\left[\widetilde{Q}(B)\right]\right| > \frac{t}{n^2}\right) \leqslant 2\exp\left(-\kappa^2 \frac{\alpha t^2}{16n^2}\right) \tag{C.26}$$

Step 3. Turning our attention to $\mathbb{E}[\widetilde{Q}(B)]$, recall that $n\binom{|B|}{2}^{1/2}\widetilde{Q}(B) = Y(B) - \widetilde{\mu}(B)$ and that $\widetilde{\mu}(B) := \sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v)/(n^2\kappa)$. As in previous lemmas, we will shorthand the quantities $s(B), \rho(B)$, and $v(B)$, by s, ρ, and v (respectively). Note that

$$\begin{aligned}\mathbb{E}\left[2\cdot \sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v)\right] &= \mathbb{E}\left\{\left[\sum_{u\in B}\widehat{d}(u)\right]^2 - \sum_{u\in B}\widehat{d}^2(u)\right\} \\ &= \mathrm{Var}\left[\sum_{u\in B}\widehat{d}(u)\right] + \mathbb{E}\left[\sum_{u\in B}\widehat{d}(u)\right]^2 - \sum_{u\in B}\mathbb{E}\left[\widehat{d}^2(u)\right] \end{aligned} \tag{C.27}$$

Part 3 of Lemma 36 gives $\mathrm{Var}\left[\sum_{u\in B}\widehat{d}(u)\right] \leqslant 9sn^2$. Furthermore, for $u \in C_i$ we have

$$\begin{aligned}\mathbb{E}\left[\widehat{d}^2(u)\right] &= \mathrm{Var}\left[\widehat{d}(u)\right] + \mathbb{E}\left[\widehat{d}(u)\right]^2 \\ &= n\pi^T V(\cdot, i) - V(i,i) + n^2\left[\pi^T P(\cdot, i) - P(i,i)\right]^2,\end{aligned}$$

and therefore $\sum_{u\in B}\mathbb{E}\left[\widehat{d}^2(u)\right] \leqslant 2sn^3$. Finally, Part 2 of Lemma 36 gives $\left|\mathbb{E}\left[\sum_{u\in B}\widehat{d}(u)\right] - |B|nv^TP\pi\right| \leqslant |B|$. By expansion, this implies there exists a constant a with $|a| < 3$ such that for large enough n, $\mathbb{E}\left[\sum_{u\in B}\widehat{d}(u)\right]^2 = s^2n^4(v^tP\pi)^2 + as^2n^3$. Therefore overall, line (C.27) implies there exists a constant b with $|b| < 6$ such that for large enough n, $\mathbb{E}\left[2\cdot\sum_{u,v\in B;u<v}\widehat{d}(u)\widehat{d}(v)\right] = s^2n^4(v^TP\pi)^2 + bsn^3$. Therefore, using the definition of $\widetilde{\mu}(B)$,

$$\begin{aligned}\mathbb{E}\left[\widetilde{\mu}(B)\right] &= s^2n^4\frac{(v^tP\pi)^2 + b(sn)^{-1}}{2n^2\kappa} = \binom{sn}{2}\left[1 + \frac{1}{sn-1}\right]\left[\frac{(v^tP\pi)^2}{\kappa} + \frac{b}{\kappa sn}\right] \\ &= \binom{sn}{2}\left[\frac{(v^tP\pi)^2}{\kappa} + \frac{b}{\kappa sn} + \frac{(v^tP\pi)^2}{\kappa(sn-1)} + \frac{b}{\kappa sn(sn-1)}\right] \\ &= \binom{sn}{2}\left[\frac{(v^tP\pi)^2}{\kappa} + \frac{1}{\kappa sn}\left(b + \frac{sn(v^tP\pi)^2 + b}{sn-1}\right)\right] = \binom{sn}{2}\left[\frac{(v^tP\pi)^2}{\kappa} + \frac{c_1}{\kappa sn}\right]\end{aligned} \tag{C.28}$$

for a constant c_1 with $|c_1| < 8$, for large enough n. Now, part 1 of Lemma 36 gives that $\left|\mathbb{E}[Y(B)] - \binom{|B|}{2} v^t P v\right| \leqslant 3|B|/2$ for large enough n. Thus there exists a constant c_2 with $|c_2| < 3$ such that for large enough n, $\mathbb{E}[Y(B)] = \binom{|B|}{2}\left[v^t P v + \frac{c_2}{sn}\right]$. Thus

$$n\mathbb{E}\left[\widetilde{Q}(B)\right] = \binom{|B|}{2}^{-1/2}(\mathbb{E}[Y(B)] - \mathbb{E}[\tilde{\mu}(B)]) = \frac{sn}{\sqrt{2}}\left[v^t P v - \frac{(v^t P \pi)^2}{\kappa} + \frac{1}{sn}\left(c_1/\kappa + c_2\right)\right]$$

$$* \left(\sqrt{1 - \frac{1}{sn}}\right) = \frac{sn}{\sqrt{2}}\left[v^t P v - \frac{(v^t P \pi)^2}{\kappa}\right] + \frac{c_1/\kappa + c_2}{\sqrt{2}}\left(\sqrt{1 - \frac{1}{sn}}\right)$$

Thus there exists a constant c with $|c| \leqslant |c_1|/\kappa + |c_2| < 8/\kappa + 3$ such that for large enough n, $\mathbb{E}[\widetilde{Q}(B)] = \mathcal{Q}(B) + c/n$. This completes Step 3.

Completion of the proof: We now recall the results of the three steps:

(i) For large enough n, we have $\mathbb{P}\left(\left|\widehat{Q}(B) - \widetilde{Q}(B)\right| > \frac{t}{2n^2} + \frac{2}{\kappa n}\right) \leqslant 2\exp\left(-\frac{\kappa^2 t^2}{n^2}\right)$

(ii) $\mathbb{P}\left(\left|\widetilde{Q}(B) - \mathbb{E}\left[\widetilde{Q}(B)\right]\right| > \frac{t}{n^2}\right) \leqslant 2\exp\left(-\kappa^2 \frac{\alpha t^2}{16 n^2}\right)$

(iii) There exists c with $|c| < 8/\kappa + 3$ such that for large enough n, $\mathbb{E}\left[\widehat{Q}(B)\right] = q(B) + c/n$

Noting that $\alpha/16 < 1$, we apply a union bound to the results of steps (i) and (ii):

$$\mathbb{P}\left(\left|\widehat{Q}(B) - \mathbb{E}\left[\widetilde{Q}(B)\right]\right| > \frac{t}{n^2} + \frac{2}{\kappa n}\right) \leqslant 4\exp\left(-\frac{\kappa^2 \alpha t^2}{16 n^2}\right) \tag{C.29}$$

Applying the inequality $|x - a| \geqslant |x| - |a|$ with (iii) and some algebra gives the result. ■

APPENDIX D
ACME SUPPLEMENTAL

D.1 Pre-processing gene read counts

Let x_{ij} denote elements of the original count matrix, where $i = 1, \ldots, n$ indexes samples and $j = 1, \ldots, T$ indexes genes. Let $l_i = \sum_j x_{ij}$ be the library size for sample i. The overall entrywise mean is $\bar{x} = \frac{\sum_i \sum_j x_{ij}}{Tn}$. The final normalized count matrix elements are $c_{ij} = \frac{x_{ij}}{l_i} T\bar{x}$. This process results in a standardized matrix with constant column sums. Similar normalization is performed by software such as DESeq2 (Love et al., 2014), but with additional attention to nonlinear scaling relationships.

D.2 Sampling scheme for residual and goodness-of-fit tests

We created a sub-sampled eQTL data set comprised of equally-sized groups of null, weak, medium, and strong eQTLs. To determine the groups, we used the detection p-value associated with the QN-linear model (described in Section 5.1.1) as an *a priori* measure of eQTL association strength within a fixed dataset. Although our intention is to provide a new effect size measure, this prior stratification provides a refined view of ACME-eQTL model behavior at various levels of association evidence. The four groups of cis-eQTL pairs were defined as follows: "null" eQTLs with $-\log_{10}$ p-value in $[0, 5)$; "weak" eQTLs with $-\log_{10}$ p-value in $[5, 10)$; "medium" eQTLs with $-\log_{10}$ p-value in $[10, 15)$; and "strong" eQTLs with $-\log_{10}$ p-value in $[15, \infty)$. The sub-sampled data were obtained by sampling 10,000 pairs from each group, uniformly-at-random.

D.3 Framework for direct null simulation

The preliminary data set used in Section 2.3 contained data from 40,000 unique gene-SNP pairs. When calculating the residual diagnostics presented in that section, we saved the estimated residual vectors (computed by the ACME fit) from each gene-SNP pair. We used these to create 25 null-data replications of each of the 40,000 pairs from the preliminary data. For a fixed gene-SNP pair, we went through the following steps:

1. Recalled s the real allele count vector, $\hat{\beta}_0$ the ACME-estimated value of β_0 from the preliminary data, and $\hat{\sigma}$ the ACME-estimated value of σ from the preliminary data

2. For $r = 1, \ldots, 25$, constructed a vector of realistic errors ε_r by

$$\varepsilon_r = \hat{\sigma}\varepsilon_r^* + \mathcal{N}_r(0, \hat{\sigma}/10)$$

   where ε_r^* is a randomly selected stored residual vector from one of the 40,000 gene-SNP pairs, scaled to have variance 1. The addition of $\mathcal{N}_r(0, \hat{\sigma}/10)$ is a "jitter" (indepenent within r and between all 40,000 pairs) to ensure that no two chosen mock-residual vectors are equivalent, while allowing them to retain any inherent non-Normality.

3. Constructed the j-th replication of null gene expression data

$$g_r = \exp\left\{ \log(\hat{\beta}_0) + Z\gamma_r + \varepsilon_r \right\}$$

   where γ_r is $\mathcal{N}_p(0_p, I_p)$ and independent within r and between all 40,000 pairs.

D.4 Tests of normality and homoskedasticity

This section contains the results from tests for normality and homoskedasticity of residuals, for each cis-QTL group, and for all single-pair eQTL models considered in this paper. For each of the 10,000 gene-SNP pairs in each eQTL group (see in Appendix D.2 above), we calculated the p-value for the canonical Shapiro-Wilk test for normality, and the p-value for the canonical Bartlett test for homoskedasticity. Figure D.1 displays boxplots of these p-values on the $-\log_{10}$ scale. We see that residuals for the linear model with raw gene expression ("RAW") are less normal and less homoskedastic than the residuals for the log-based models (ACME-eQTL, log-linear ("LL"), and log-ANCOVA ("ANCOVA")). This is particularly true for weak eQTLs, for which error assumptions are most important for Type I error control. The dark-red lines in the figure represent the typical FDR cut-off applied to each bin of the data.

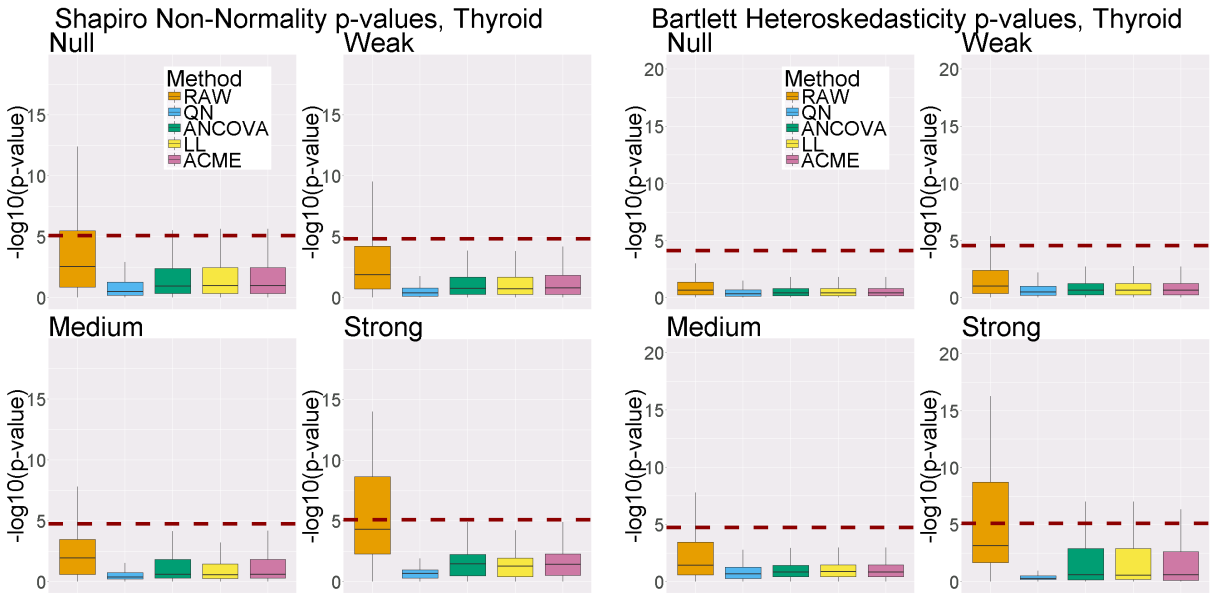

Figure D.1: Boxplots of $-\log_{10}$ Shapiro-Wilk and Bartlett p-values from all models. Above, "AOV" denotes the log-ANCOVA model, "LL" the log-linear model, and "RAW" the standard linear model with un-transformed gene expression. The dark red dashed line indicates the FDR $\alpha = 0.1$ significance cut-off for the particular bin.

D.5 QN-linear regression vs. ACME-eQTL regression

An illustration of the information loss incurred by the QN-linear model approach to eQTL analysis can be made by considering two gene-SNP pairs, each with eQTL evidence that is similar under QN transformation, but disparate on the log scale. We chose one such pair from real data, displayed in Figure D.2. While the estimated ACME-eQTL effect size of pair 2 is ten times greater than that of pair 1, the effect sizes from linear regression with QN-transformed expression are nearly the same. Furthermore, the baseline expression of pair 1 is far greater than that of pair 2, a feature that is obscured by the QN transformation.

D.6 ACME-eQTL fitting algorithm

Here we describe the iterative algorithm used to identify parameters β_0 and β_1 that approximately maximize the likelihood of the ACME-eQTL model. For a particular gene-SNP pair, maximizing the likelihood is equivalent to minimizing the sum of squares

$$\sum_i (y_i - \log(\beta_0 + \beta_1 s_i) - \langle \boldsymbol{Z}_i, \gamma \rangle)^2 \tag{D.1}$$

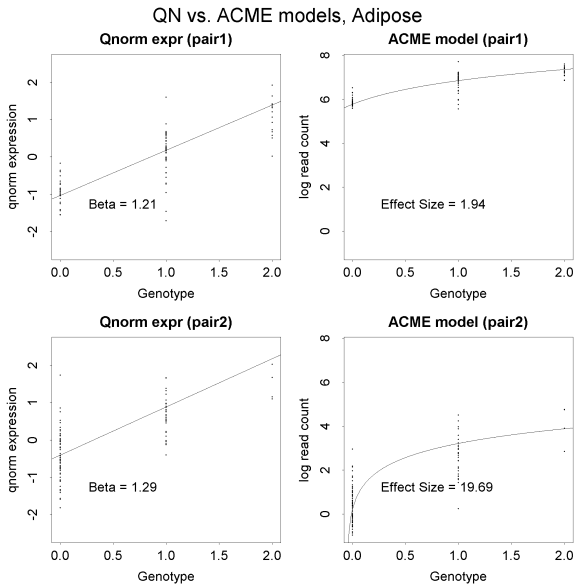

Figure D.2: eQTL data from two selected gene-SNP pairs from Adipose tissue. The fitted lines correspond to the estimated parameters from each model.

over the parameters β_0, β_1 and γ. The iterative algorithm operates by carrying out least-squares regression on an approximating linear model. Denoting the natural effect size β_1/β_0 by η, under the ACME-eQTL model the conditional mean of y_i given $\boldsymbol{Z}_i$ and s_i is given by

$$\mathbb{E}[y_i|\boldsymbol{Z}_i, s_i] = \log(\beta_0) + \log(1 + s_i\eta) + \boldsymbol{Z}_i^T\gamma \tag{D.2}$$

A first-order Taylor approximation of $\log(1 + s_i\eta)$ around η at an estimate $\hat{\eta}^j$ gives

$$\mathbb{E}[y_i|\boldsymbol{Z}_i, s_i] \approx \log(\beta_0) + \log(1 + s_i\,\hat{\eta}^j) + \frac{s_i}{1 + s\,\hat{\eta}^j}(\eta - \hat{\eta}^j) + \boldsymbol{Z}_i^T\gamma \tag{D.3}$$

We use this approximation to motivate a linear model, in which the response variable is $d_i := y_i - \log(1 + s_i\,\hat{\eta}^j)$. Define $\theta_0 := \log(\beta_0)$ and $\theta_1 := \eta - \hat{\eta}^j$. Then subtracting $\log(1 + s_i\,\hat{\eta}^j)$ from each side of Equation D.3 yields the following linear model in the parameters θ_0, θ_1, and γ:

$$d_i = \theta_0 + \frac{s_i}{1 + s_i\,\hat{\eta}^j}\theta_1 + \boldsymbol{Z}_i^T\gamma + \varepsilon_i \tag{D.4}$$

After fitting this model at the j^{th} iteration, we set $\hat{\eta}^{j+1}$ to $\hat{\theta}_1 + \hat{\eta}^j$, and repeat the procedure. This is repeated until $|\hat{\eta}^j - \hat{\eta}^{j+1}|$ is close to machine precision. As the likelihood (D.1) is convex, we can

expect convergence, since the algorithm is similar to a Gauss-Newton procedure. The last estimates of θ_0 and θ_1 are then used to obtain estimates of β_0 and β_1 via the equations $\hat{\beta}_0 = \exp\{\hat{\theta}_0\}$ and $\hat{\beta}_1 = \hat{\beta}_0\hat{\eta}$.

We set the initial estimate of η to 0. If for any j, $1 + s_i\hat{\eta}^j$ is negative for any index i, we divide $\hat{\eta}^j$ by 2 and restart the j-th iteration.

D.7 Derivation of effect size standard error

Defining $\theta_0 := \log(\beta_0)$ and $\eta := \beta_1/\beta_0$, the ACME-eQTL model may be expressed

$$y_i = \theta_0 + \log(1 + \eta s_i) + \boldsymbol{Z}_i^T\gamma + \varepsilon_i \tag{D.5}$$

for samples $i = 1, \ldots, n$. As discussed in Section 5.1.3.2, we use "effect size" to refer to η. Let $\hat{\varepsilon}_i$ be the estimated residual for patient i from (D.5). Let C be an orthonormalization of the matrix $(\mathbf{1}_n, \boldsymbol{Z}^T)$, and define $P := (I_n - CC^T)$. Letting y, $\log(1+\eta s)$, and $\hat{\varepsilon}$ be $n \times 1$ vectors corresponding to the full set of sample data, we have

$$\hat{\varepsilon} = P\left[y - \log(1 + \eta s)\right], \tag{D.6}$$

as the matrix P residualizes the effect of θ_0 and γ. Thus, the log-likelihood for the full model may be expressed in terms of η, P, σ^2 only:

$$\begin{aligned} -\log L(y; s, \eta, P, \sigma^2) &= \frac{n}{2}\log(2\pi\sigma^2) + \\ \sigma^{-2}&[y - \log(1 + \eta s)]^T P^T P[y_i - \log(1 + \eta s)] \end{aligned} \tag{D.7}$$

We now derive an approximate observed Fisher information for η, using (D.6). Note that $\frac{d}{d\eta}\hat{\varepsilon} = -P\frac{s}{1+\eta s}$ and $\frac{d^2}{(d\eta)^2}\hat{\varepsilon} = P\frac{s^2}{(1+\eta s)^2}$ (where all operations to vectors are component-wise). Define

$d := y - \log(1 + \eta s)$. Then

$$
\begin{aligned}
-I(\eta) &= \frac{d^2}{(d\eta)^2}\left[\frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\hat{\varepsilon}^T\hat{\varepsilon}\right] \\
&= \frac{1}{2\sigma^2}2\left[\hat{\varepsilon}''^T\hat{\varepsilon} + \hat{\varepsilon}'^T\hat{\varepsilon}'\right] \\
&= \frac{1}{\sigma^2}\left[\left(\frac{s^2}{(1+\eta s)^2}\right)^T Pd + \left(\frac{s}{1+\eta s}\right)^T P\frac{s}{1+\eta s}\right],
\end{aligned}
$$

since P is idempotent. The asymptotic standard error for the model can then be estimated by $\sqrt{-I(\hat{\eta})^{-1}}$ with σ^2 replaced by $\hat{\sigma}^2$. Uncertainty in the remaining parameters and their effect on η is propagated through P. To check the accuracy of the standard error, we computed a purely numerical Hessian matrix for the log-likelihood and the full Fisher observed information matrix, verifying a close numerical match to the derivation above.

BIBLIOGRAPHY

L. A. Adamic and N. Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

C. Aicher, A. Z. Jacobs, and A. Clauset. Learning latent block structure in weighted networks. *Journal of Complex Networks*, page cnu026, 2014.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems*, pages 33–40, 2009.

R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

E. Almaas, B. Kovacs, T. Vicsek, Z. Oltvai, and A.-L. Barabási. Global organization of metabolic fluxes in the bacterium escherichia coli. *Nature*, 427(6977):839–843, 2004.

R. Andersen, D. F. Gleich, and V. Mirrokni. Overlapping clusters for distributed computation. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 273–282. ACM, 2012.

C. J. Anderson, S. Wasserman, and K. Faust. Building stochastic blockmodels. *Social networks*, 14(1-2):137–161, 1992.

A. Ansari, O. Koenigsberg, and F. Stahl. Modeling multiple relationships in social networks. *Journal of Marketing Research*, 48(4):713–728, 2011.

K. G. Ardlie, D. S. Deluca, A. V. Segrè, T. J. Sullivan, T. R. Young, E. T. Gelfand, C. A. Trowbridge, J. B. Maller, T. Tukiainen, M. Lek, et al. The genotype-tissue expression (gtex) pilot analysis: Multitissue gene regulation in humans. *Science*, 348(6235):648–660, 2015.

L. Bao, X. Xia, and Y. Cui. Expression qtl modules as functional components underlying higher-order phenotypes. *PloS one*, 5(12), 2010.

A.-L. Barabasi and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113, 2004.

M. J. Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76 (6):066102, 2007.

M. Barigozzi, G. Fagiolo, and G. Mangioni. Identifying the community structure of the international-trade multi-network. *Physica A: statistical mechanics and its applications*, 390 (11):2051–2066, 2011.

A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.

T. E. Bartlett. Co-modularity and co-community detection in large networks. *arXiv preprint arXiv:1511.05611*, 2015.

F. Battiston, V. Nicosia, and V. Latora. Structural measures for multiplex networks. *Physical Review E*, 89(3):032804, 2014.

T. M. Beasley, S. Erickson, and D. B. Allison. Rank-based inverse normal transformations are increasingly used, but are they merited? *Behavior genetics*, 39(5):580–595, 2009.

E. A. Bender. The asymptotic number of non-negative integer matrices with given row and column sums. *Discrete Mathematics*, 10(2):217–223, 1974.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

M. Berlingerio, M. Coscia, and F. Giannotti. Finding and characterizing communities in multidimensional networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 490–494. IEEE, 2011.

M. Berlingerio, F. Pinelli, and F. Calabrese. Abacus: frequent pattern mining-based community discovery in multidimensional networks. *Data Mining and Knowledge Discovery*, 27(3):294–320, 2013.

P. J. Bickel and A. Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.

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

K. Bodwin, K. Zhang, and A. Nobel. A testing-based approach to the discovery of differentially correlated variable sets. *arXiv preprint arXiv:1509.08124*, 2015.

B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.

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

I. Cabreros, E. Abbe, and A. Tsirigos. Detecting community structures in hi-c genomic data. *arXiv:1509.05121*, 2015.

A. Cardillo, J. Gómez-Gardenes, M. Zanin, M. Romance, D. Papo, F. del Pozo, and S. Boccaletti. Emergence of network features from multiplexity. *arXiv preprint arXiv:1212.2153*, 2012.

S. E. Castel, A. Levy-Moonshine, P. Mohammadi, E. Banks, and T. Lappalainen. Tools and best practices for data processing in allelic expression analysis. *Genome biology*, 16(1):1, 2015.

A. Celisse, J.-J. Daudin, L. Pierre, et al. Consistency of maximum-likelihood and variational estimators in the stochastic block model. *Electronic Journal of Statistics*, 6:1847–1899, 2012.

B. L. Chamberlain et al. Graph partitioning algorithms for distributing workloads of parallel computations. *University of Washington Technical Report UW-CSE-98-10*, 3, 1998.

J. Chen and B. Yuan. Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, 22(18):2283–2290, 2006.

M. Chen, K. Kuzmin, and B. K. Szymanski. Extension of modularity density for overlapping community structure. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 856–863. IEEE, 2014.

D. S. Choi, P. J. Wolfe, and E. M. Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, page asr053, 2012.

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

F. R. Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

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

A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

A. Coja-Oghlan and A. Lanka. Finding planted partitions in random graphs with general degree distributions. *SIAM Journal on Discrete Mathematics*, 23(4):1682–1714, 2009.

A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

G. Csárdi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

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

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

B. Devlin and K. Roeder. Genomic control for association studies. *Biometrics*, 55(4):997–1004, 1999.

C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 107–114. IEEE, 2001.

A. L. Dixon, L. Liang, M. F. Moffatt, W. Chen, S. Heath, K. C. Wong, J. Taylor, E. Burnett, I. Gut, M. Farrall, et al. A genome-wide association study of global gene expression. *Nature genetics*, 39(10):1202–1207, 2007.

N. Du, B. Wang, B. Wu, and Y. Wang. Overlapping community detection in bipartite networks. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 176–179. IEEE Computer Society, 2008.

N. Durak, T. G. Kolda, A. Pinar, and C. Seshadhri. A scalable null model for directed graphs matching all degree distributions: In, out, and reciprocal. In *Network Science Workshop (NSW), 2013 IEEE 2nd*, pages 23–30. IEEE, 2013.

K. A. Eriksen, I. Simonsen, S. Maslov, and K. Sneppen. Modularity and extreme edges of the internet. *Physical review letters*, 90(14):148701, 2003.

M. Fan, K.-C. Wong, T. Ryu, T. Ravasi, and X. Gao. Secom: A novel hash seed and community detection based-approach for genome-scale protein domain identification. *PLoS ONE*, 7:e39475, 06 2012.

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

S. Ferriani, F. Fonti, and R. Corrado. The social and economic bases of network multiplexity: Exploring the emergence of multiplex ties. *Strategic Organization*, 11(1):7–34, 2013.

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

S. E. Fienberg, M. M. Meyer, and S. S. Wasserman. Analyzing data from multivariate directed graphs: An application to social networks. Technical report, DTIC Document, 1980.

S. E. Fienberg, M. M. Meyer, and S. S. Wasserman. Statistical analysis of multiple sociometric relations. *Journal of the american Statistical association*, 80(389):51–67, 1985.

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

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

E. R. Gamazon, H. E. Wheeler, K. P. Shah, S. V. Mozaffari, K. Aquino-Michaels, R. J. Carroll, A. E. Eyler, J. C. Denny, D. L. Nicolae, N. J. Cox, et al. A gene-based association method for mapping traits using reference transcriptome data. *Nature genetics*, 47(9):1091–1098, 2015.

Y. Gilad, S. A. Rifkin, and J. K. Pritchard. Revealing the architecture of gene regulation: the promise of eqtl studies. *Trends in genetics*, 24(8):408–415, 2008.

M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

D. F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 597–605. ACM, 2012.

M. K. Goldberg, M. Hayvanovych, and M. Magdon-Ismail. Measuring similarity between sets of overlapping clusters. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 303–308. IEEE, 2010.

E. Grundberg, K. S. Small, Å. K. Hedman, A. C. Nica, A. Buil, S. Keildson, J. T. Bell, T.-P. Yang, E. Meduri, A. Barrett, et al. Mapping cis-and trans-regulatory effects across multiple tissues in twins. *Nature genetics*, 44(10):1084–1089, 2012.

R. Guimera and L. A. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Y. Huang, S. Wuchty, M. T. Ferdig, and T. M. Przytycka. Graph theoretical approach to study eqtl: a case study of plasmodium falciparum. *Bioinformatics*, 25(12):i15–i20, 2009.

R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

A. Z. Jacobs and A. Clauset. A unified view of generative models for networks: models, methods, opportunities, and challenges. *arXiv:1411.4070*, 2014.

R. K.-X. Jin, D. C. Parkes, and P. J. Wolfe. Analysis of bidding networks in ebay: aggregate preference identification through community detection. In *Proceedings of AAAI workshop on plan, activity and intent recognition (PAIR)*, 2007.

D. Kahle and H. Wickham. ggmap: Spatial visualization with ggplot2. *The R Journal*, 5(1): 144–161, 2013.

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

D. I. Kim, P. K. Gopalan, D. Blei, and E. Sudderth. Efficient online inference for bayesian nonparametric relational models. In *Advances in Neural Information Processing Systems*, pages 962–970, 2013.

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

A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, 2009.

A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

A. Lancichinetti, F. Radicchi, J. J. Ramasco, S. Fortunato, et al. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011.

R. Langone, C. Alzate, and J. A. Suykens. Modularity-based model selection for kernel spectral clustering. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1849–1856. IEEE, 2011.

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

J. Lei, A. Rinaldo, et al. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.

J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370. ACM, 2010a.

J. Leskovec et al. Stanford network analysis project. *ht tp://snap. stanford. edu*, 2010b.

Y. Li, M. Liang, and Z. Zhang. Regression analysis of combined gene expression regulation in acute myeloid leukemia. *PLoS Comput Biol*, 10(10):e1003908, 2014.

X. Liu and T. Murata. Community detection in large-scale bipartite networks. *Information and Media Technologies*, 5(1):184–192, 2010.

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

M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):1, 2014.

D. Lusseau and M. E. Newman. Identifying the role that animals play in their social networks. *Proceedings of the Royal Society of London B: Biological Sciences*, 271(Suppl 6):S477–S481, 2004.

D. J. McCarthy, Y. Chen, and G. K. Smyth. Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic acids research*, page gks042, 2012.

P. Mohammadi, S. E. Castel, A. A. Brown, and T. Lappalainen. Quantifying the regulatory effect size of *cis*-acting genetic variation using allelic fold change. 2016.

M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

M. Morley, C. M. Molony, T. M. Weber, J. L. Devlin, K. G. Ewens, R. S. Spielman, and V. G. Cheung. Genetic analysis of genome-wide variation in human gene expression. *Nature*, 430(7001): 743–747, 2004.

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

A. J. Myers, J. R. Gibbs, J. A. Webster, K. Rohrer, A. Zhao, L. Marlowe, M. Kaleem, D. Leung, L. Bryden, P. Nath, et al. A survey of genetic human cortical gene expression. *Nature genetics*, 39(12):1494–1499, 2007.

C. G. A. R. Network et al. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *The New England journal of medicine*, 368(22):2059, 2013.

M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003a.

M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003b.

M. E. Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004a.

M. E. Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004b.

M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006a.

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

M. E. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8 (1):25–31, 2012.

M. E. Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822, 2013.

M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

M. E. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03024, 2009.

K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

S. C. Olhede and P. J. Wolfe. Degree-based network models. *arXiv preprint arXiv:1211.6537*, 2012.

T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136): 664–667, 2007.

S. Paul and Y. Chen. Community detection in multi-relational data with restricted multi-layer stochastic blockmodel. *arXiv preprint arXiv:1506.02699*, 2015.

S. Paul and Y. Chen. Null models and modularity based community detection in multi-layer networks. *arXiv preprint arXiv:1608.00623*, 2016.

L. Peel, D. B. Larremore, and A. Clauset. The ground truth about metadata and community detection in networks. *arXiv preprint arXiv:1608.05878*, 2016.

T. P. Peixoto. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Physical Review E*, 92(4):042807, 2015.

J. Platig, P. Castaldi, D. DeMeo, and J. Quackenbush. Bipartite community structure of eqtls. *arXiv preprint arXiv:1509.02816*, 2015.

P. Pons and M. Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

P. Pons and M. Latapy. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):191–218, 2006.

M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the AMS*, 56 (9):1082–1097, 2009.

A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications*, 11(3):430–452, 1990.

J. E. Powell, A. K. Henders, A. F. McRae, J. Kim, G. Hemani, N. G. Martin, E. T. Dermitzakis, G. Gibson, G. W. Montgomery, and P. M. Visscher. Congruence of additive and non-additive effects on gene expression estimated from pedigree and snp data. *PLoS Genet*, 9(5):e1003502, 2013.

U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

M. Rantalainen, C. M. Lindgren, and C. C. Holmes. Robust linear models for cis-eqtl analysis. *PloS one*, 10(5):e0127882, 2015.

J. Reichardt and S. Bornholdt. Clustering of sparse data via network communitiesa prototype study of a large online market. *Journal of Statistical Mechanics: Theory and Experiment*, 2007 (06):P06016, 2007a.

J. Reichardt and S. Bornholdt. Clustering of sparse data via network communitiesa prototype study of a large online market. *Journal of Statistical Mechanics: Theory and Experiment*, 2007 (06):P06016, 2007b.

T. Richardson, P. J. Mucha, and M. A. Porter. Spectral tripartitioning of networks. *Physical Review E*, 80(3):036111, 2009.

M. Rocklin and A. Pinar. On clustering on graphs with multiple edge types. *Internet Mathematics*, 9(1):82–112, 2013.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, pages 1878–1915, 2011.

M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

S. Sahebi and W. W. Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems and the Social Web, RSWEB*, 2011.

P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter*, 7(2):31–40, 2005.

A. A. Shabalin. Matrix eqtl: ultra fast eqtl analysis via large matrix operations. *Bioinformatics*, 28(10):1353–1358, 2012.

A. A. Shabalin, V. J. Weigman, C. M. Perou, and A. B. Nobel. Finding large average submatrices in high dimensional data. *The Annals of Applied Statistics*, pages 985–1012, 2009.

S. Shirinivas, S. Vetrivel, and N. Elango. Applications of graph theory in computer science an overview. *International Journal of Engineering Science and Technology*, 2(9):4610–4621, 2010.

G. K. Smyth. Nonlinear regression. *Encyclopedia of environmetrics*, 2002.

T. A. Snijders and K. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of classification*, 14(1):75–100, 1997.

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

O. Stegle, L. Parts, M. Piipari, J. Winn, and R. Durbin. Using probabilistic estimation of expression residuals (peer) to obtain increased power and interpretability of gene expression analyses. *Nature protocols*, 7(3):500–507, 2012.

J. H. Steiger and A. R. Hakstian. The asymptotic distribution of elements of a correlation matrix: Theory and application. *British Journal of Mathematical and Statistical Psychology*, 1982.

B. E. Stranger, A. C. Nica, M. S. Forrest, A. Dimas, C. P. Bird, C. Beazley, C. E. Ingle, M. Dunning, P. Flicek, D. Koller, et al. Population genomics of human gene expression. *Nature genetics*, 39 (10):1217–1224, 2007.

S. Szymczak, M. O. Scheinhardt, T. Zeller, P. S. Wild, S. Blankenberg, and A. Ziegler. Adaptive linear rank tests for eqtl studies. *Statistics in medicine*, 32(3):524–537, 2013.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

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

H.-J. Westra, M. J. Peters, T. Esko, H. Yaghootkar, C. Schurmann, J. Kettunen, M. W. Christiansen, B. P. Fairfax, K. Schramm, J. E. Powell, et al. Systematic identification of trans eqtls as putative drivers of known disease associations. *Nature genetics*, 45(10):1238–1243, 2013.

J. J. Whang, P. Rai, and I. S. Dhillon. Stochastic blockmodel with cluster overlap, relevance selection, and similarity-based smoothing. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 817–826. IEEE, 2013.

J. D. Wilson, S. Wang, P. J. Mucha, S. Bhamidi, A. B. Nobel, et al. A testing based extraction algorithm for identifying significant communities in networks. *The Annals of Applied Statistics*, 8(3):1853–1891, 2014.

J. D. Wilson, J. Palowitch, S. Bhamidi, and A. B. Nobel. Community extraction in multilayer networks with heterogeneous community structure. *arXiv preprint arXiv:1610.06511*, 2016.

F. A. Wright, A. A. Shabalin, and I. Rusyn. Computational tools for discovery and interpretation of expression quantitative trait loci. *Pharmacogenomics*, 13(3):343–352, 2012.

F. A. Wright, P. F. Sullivan, A. I. Brooks, F. Zou, W. Sun, K. Xia, V. Madar, R. Jansen, W. Chung, Y.-H. Zhou, et al. Heritability and genomics of gene expression in peripheral blood. *Nature genetics*, 46(5):430–437, 2014.

J. Xie, B. K. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 344–349. IEEE, 2011.

J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (csur)*, 45(4):43, 2013.

L. Xin, E. Haihong, J. Song, M. Song, and J. Tong. Book recommendation based on community detection. In *Pervasive Computing and the Networked World*, pages 364–373. Springer, 2014.

Y. Zhao, E. Levina, and J. Zhu. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.

Y. Zhao, E. Levina, J. Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

Y.-H. Zhou and F. A. Wright. Hypothesis testing at the extremes: fast and robust association for high-throughput data. *Biostatistics*, page kxv007, 2015.

Y.-H. Zhou, K. Xia, and F. A. Wright. A powerful and flexible approach to the analysis of rna sequence count data. *Bioinformatics*, 27(19):2672–2678, 2011.

Z. Zhu, F. Zhang, H. Hu, A. Bakshi, M. R. Robinson, J. E. Powell, G. W. Montgomery, M. E. Goddard, N. R. Wray, P. M. Visscher, et al. Integration of summary data from gwas and eqtl studies predicts complex trait gene targets. *Nature genetics*, 2016.

I. Zwiener, B. Frisch, and H. Binder. Transforming rna-seq data to improve the performance of prognostic gene signatures. *PloS one*, 9(1):e85150, 2014.