Selected topics in network data analysis and phylogenetics

by

Xiaodong Jiang

(Under the Direction of Pengsheng Ji and Liang Liu)

Abstract

Network data analysis is an essential topic in statistical learning field, with ubiquitous applications in social science, physics, biology, etc. In this dissertation, we first propose a set of novel models and algorithms to perform community detection in networks with node attributes and provide theoretical and experimental studies. In the second part, we answered a fundamental question in network data analysis - testing the existence of communities. The Peak dEnsity raTio (PET) statistic is proposed to achieve this goal. An experimental study with simulated networks and real-world benchmark data sets show that our approach can effectively differentiate the presence and absence of communities. A generalized community detection method is applied to phylogenomic data for understanding the evolutionary history of species, often described as a phylogenetic network, under a mixture multispecies coalescence model. The generalized detection method is able to successfully reconstruct the phylogenetic network from phylogenomic data.

INDEX WORDS: Network data analysis, Statistical machine learning, Community detection, Hypothesis testing, Phylogenetic analysis, Mixture coalescence model

Selected topics in network data analysis and phylogenetics

by

Xiaodong Jiang

B.S., Beijing University of Technology, 2014

M.S., University of Georgia, 2016

A Dissertation Submitted to the Graduate Faculty

of The University of Georgia in Partial Fulfillment

of the

Requirements for the Degree

Doctor of philosophy

Athens, Georgia

2019

Selected topics in network data analysis and phylogenetics

by

Xiaodong Jiang

Approved:

Major Professor:	Pengsheng Ji Liang Liu
Committee:	Lynne Billard T.N. Sriram Cheolwoo Park Shuyang Bai

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
May 2019

Selected topics in network data analysis and phylogenetics

Xiaodong Jiang

May 2019

For Mom and Dad

Acknowledgments

The past five years has been the best time in my life. I would like to thank my advisors, Dr. Pengsheng Ji and Dr. Liang Liu, for their continuous, selfless, and endless help, support, and guidance. Dr. Ji introduced me to the area of network data analysis, taught me how to be a statistician and researcher. Dr. Liu is a remarkable researcher, a good friend, and true role model in science, his rigorous and wise thinking guided me forward.

There is never an easy path that leads to the destination. I would like to thank my parents for their love, encouragement, support, and providing me the best education in my life, without whom I will never enjoy so many opportunities.

Finally, I want to thank my wife, Xuan Zhang, who gives me the best time in my life. Without her accompany, I will never achieve so many goals in my life.

Contents

List of Figures

List of Tables

Chapter 1

Overview

Network-structured data is ubiquitous, from social network platforms to citation networks and co-authorship relations, from protein-protein interactions to chemical molecules. Network, as a complex data structure, is very useful in describing the relationships (edges) of objects (nodes). Traditional statistical methods encounter substantial challenges in analyzing such complex, dependent, and dynamic systems; thus more flexible and sophisticated methodologies are on demand. From a statistical learning perspective, network data analysis can be categorized into two major classes, supervised and unsupervised learning. Community detection, one type of unsupervised learning algorithms, attempts to identify a group of vertices that have stronger connections compared to the nodes in other groups. Most existing community detection algorithms focus on the structural connectivity, while neglecting the importance of node covariate. We address this problem in Chapter 2 by proposing a set of novel spectral algorithms to perform community detection

in networks with node variables, i.e., attributed networks.

While communities or clusters can be found in many real-world networks, it is also typical to observe a network, or at least a subset of a network, that does not contain any clusters. Thus, a more fundamental question is, does community exist in a given network? We answered this question in Chapter 3 by proposing the Peak dEnsity raTio (PET) statistic. We justify the framework by presenting extensive experiments with simulated and real-world network data sets.

Network analysis with large scale, complex, and accessible biology data opened up a new era for computational biology. The recent technological and technical advances in biology provided great challenges and opportunities. In Chapter 4, we build a bridge between network analysis and phylogenetics. The phylogenetic analysis aims to understand the evolutionary relationships among biological entities, such as species and genes. The evolutionary history of species is often represented by a phylogenetic tree, or more precisely a species tree, in which the internal nodes are most recent common ancestors and branch lengths are species divergence times. However, the evolution process of species may not always be characterized by a tree structure, and some common biology processes may introduce additional pathes to connect the tree edges, resulting in a so-called phylogenetic network. Estimation and inference for phylogenetic networks suffer from computational challenges, which cannot be solved within polynomial time. We instead argue that a phylogenetic network can be described by a mixture coalescent model for a set of species trees decomposed from the phylogenetic network. We will present our proposed models and algorithms in Chapter 4.

Chapter 2

Collaborative Spectral Clustering in Attributed Networks

2.1. Introduction

Community detection is a fundamental question in network data analysis, with various applications in biology, social science, physics, and computer science, which aims to cluster the nodes into groups, i.e., communities. A large number of classical community detection algorithms have been proposed in the past decades, and many of them are built under the umbrella of Stochastic Block Model by Holland et al. [1983] and its Degree-Corrected variant by Karrer and Newman [2011]. Such methods include but not limited to the following different approaches, (1). Newman [2006b]; Bickel and Chen [2009a]; Zhao et al. [2012b]; Chen et al. [2015] proposed methods based on Modularity maximization, (2). Newman [2006b]; Arash A

et al. [2013] developed likelihood-based algorithms, (3). Rohe et al. [2011]; Qin and Rohe [2013]; Jin [2015]; Ji and Jin [2016] established various spectral clustering methods with certain normalization and regularizations.

However, most of the above methods merely model the adjacency matrix or graph Laplacian with structural information, while ignoring the node attributes or node covariates. In many real-world scenarios, where there are more than one layers of information available from different sources, then statistical inference purely dependent on one single layer, such as structural adjacency matrix, would lead to information loss. For example, Ji and Jin [2016] analyzed the social network for statisticians and identified several meaningful communities or research groups with the structural adjacency matrix. Beyond this, we observe that spatial information actually plays an important role in forming research communities in coauthorship networks, such as the Bayesian Statistics community in the Research Triangle area in North Carolina, since geographical short distance creates a convenient and efficient environment for researchers' collaborations. Ji and Jin [2016] also identified the influential papers and communities from a paper-paper citation network, where the edges are purely constructed by the actual citation relations, and the results are challenging to interpret in some cases.

Some recent works tried to combine the structural and covariate information simultaneously to perform community detection. Yang et al. [2013] introduced a Bayesian framework for clustering networks with binary-coded categorical attributes; Zhou et al. [2009] used various graph clustering methods and a tuning parameter to control the weights between the node and its attributes; Binkiewicz

et al. [2017] proposed a spectral framework to include the attribute information with Gram matrix transformation. These methods can effectively detect the communities by incorporating both structural and attribute information, but the empirical performances still have room to improve, especially in unbalanced and heterogeneous networks.

In this chapter, we first generalize classical Stochastic Block Model (SBM) and its Degree-Corrected variant (DCBM) to Node Attributed versions, NSBM and NDCBM respectively. We then develop novel spectral clustering algorithms under these two models, where each node has a p-dimensional meta covariate from various formats such as text, image, speech, etc. The connectivity matrix $W_{n\times n}$ is constructed with the adjacency matrix $A_{n\times n}$ and covariate matrix $X_{n\times p}$, and $W = (1-\alpha)A + \alpha K(X, X')$, where $\alpha \in [0, 1]$ and K is a kernel to measure the covariate similarities. We then perform the eigen-decomposition on the aggregated connectivity matrix, and run a classical k-means algorithm with the element-wise ratio on leading eigenvectors.

Our approach is different from the current literature. For example, we note that Binkiewicz et al. [2017] used a simple Gram matrix to organize the covariate information and then aggregate with graph Laplacian via a tuning parameter, whereas we use a more general kernel method on the node covariates, combine with adjacency matrix, then perform k-means on the element-wise ratio of the leading eigenvectors, demonstrate the advantages of our proposed method in both theoretical and practical point of views in later sections. Furthermore, our approach is distinct from the popular kernelized clustering in Langone et al. [2015] and Yan

and Sarkar [2016] in the way we perform eigen-decomposition. These methods conduct the eigen-decomposition directly on a kernel matrix while we perform this on aggregated matrix with both adjacency matrix and kernels combined. These two approaches have large differences in analyzing the corresponding eigenspace, and we also demonstrate in section 2.5 that our approach enjoys better performance.

To conclude this section, we establish a new theoretical framework for community detection in networks with node attributes, which could combine information from different sources, and use tuning parameters to balance between them, then perform spectral clustering under certain regularizations, including taking element-wise ratio or row-normalization on leading eigenvectors. The remaining part of the paper is organized as follows. In section 2.2, we give the necessary preliminaries and notations and propose two new stochastic block models with node attributes. In section 2.3, we analyze two classes of algorithms for community detection under different model settings, with the theoretical analysis presented in section 2.4, while all related proofs have been included in the supplementary materials. We give real data examples and extensive simulations in section 2.5.

2.2. Preliminaries

We start by giving some notations and preliminaries for networks with node attributes. Let $G = (V, E, X)$ be an unweighted undirected network with n nodes and R communities, where V is the node or vertex set, E is the edge set, and X is the node attribute matrix. Please note that we will use vertex and node

exchangeably throughout the article.

Notations and Definitions

We organize the necessary notation as follows.

- Adjacent matrix. Let $A_{n\times n}$ be the empirical adjacency matrix, for unweighted undirected network, $A_{ij} = 1$ if vertex i and vertex j are connected, otherwise 0. Let $\Omega_{n\times n}$ be the noiseless version of adjacency matrix, where $\Omega = E[A]$.

- Membership matrix. Let $M_{n\times R}$ be the membership matrix such that $M(i, r) = 1$ if the vertex i is in the rth community. Each vertex belongs to only one community, thus the row sum is always 1.

- Degree intensities of vertex. Let Θ be an $n \times n$ diagonal matrix with all positive diagonal entries; it models additional variabilities of the edge probability at the vertex level.

- Degree intensities of communities. Let D be an $R \times R$ diagonal matrix with all positive diagonal entries with $tr(D^2) = 1$. Assume

$$M'\Theta^2 M = D^2.$$

- Structural connectivity matrix between communities. Suppose P is a $R \times R$ non-negative and symmetric matrix, and $P = (p_{i,j})$ with $\max_{i,j} p_{i,j} = 1$.

- Attribute design matrix. Let $X_{n\times p}$ be attribute design matrix, with p random variables, without loss of generality, we assume $X_{i\cdot} \in N(\mu_{C_i}, \sigma^2_{C_i})$, where C_i

is the membership of vertex i, $C_i = 1, ..., R$; $i = 1, ..., n$; $X_{i\cdot}$ is the ith row in X.

- Kernel matrix. Let $K_{n\times n}$ be kernel matrix, $K := (k(x_i, x_j))_{n\times n}$, where k is a kernel function and $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$, for $i, j = 1, ..., n$, Φ is a feature map.

- Attribute connectivity matrix between communities. Suppose P' is a $R \times R$ non-negative and symmetric matrix, then $P' = (k(\mathcal{X}_{C_i}, \mathcal{X}_{C_j}))_{R\times R}$.

We now review the popular Stochastic Block Model (SBM) and its Degree-Corrected variant (DCBM) models for network with community structure, then give a formal definition of kernel in statistics and machine learning communities.

Definition 1 *(Stochastic Block Model) A stochastic block model (SBM) with n nodes and R communities is parameterized with the membership matrix M and structural connectivity matrix P. The adjacency matrix $A = (a_{ij})_{n\times n}$ is generated as* Bernoulli$(P_{g_i g_j})$ *independently for $i \neq j$, otherwise 0, where g_i is the membership of node i.*

Definition 2 *(Degree-Corrected Block Model) Degree-corrected block model (DCBM) is an extension of SBM to allow for the node heterogeneities. A DCBM is parameterized with the membership matrix M, structural connectivity matrix P, and diagonal degree intensity matrix Θ. The adjacency matrix $A = (a_{ij})_{n\times n}$ is generated as* Bernoulli$(\theta_i \theta_j P_{g_i g_j})$ *independently for $i \neq j$, otherwise 0.*

The SBM model implies the node degrees follow a Poisson distribution, while people often observe that the degrees often follow power-law distributions on real world networks; see, for example, Newman [2006b]. It is known that the restrictions of SMB have negative effects on clustering, so DCBM is proposed by Karrer and Newman [2011] to allow more flexibility on the parameters and fit the real world networks better.

Definition 3 *(Kernel) Given a kernel k and inputs $x_1, ..., x_n \in \mathcal{X}$, then the $n \times n$ kernel matrix*

$$K := (k(x_i, x_j))_{ij}$$

with respect to $x_1, ..., x_n$.

We conclude this section with some examples of kernel functions, where some of them are used in our theoretical and real data analysis. Let x and y denote some sample vectors of the same length, then a set of classical kernel functions are defined as follows.

Gaussian kernel $k(x, y) = \exp\{-||x - y||^2/(2\sigma^2)\}$.

Polynomial kernel $k(x, y) = (x'y + 1)^d$, with d as a constant.

Cosine kernel $k(x, y) = \phi(x)'\phi(y)/||\phi(x)||||\phi(y)||$, with certain mapping function ϕ.

Other interesting kernels are Spline kernel, ANOVA kernel, Graph kernel, etc. We refer readers to Hofmann et al. [2008] as a comprehensive review on kernel method in machine learning and statistics.

Stochastic Block Models for Network with Node Attributes. We pro-

pose two classes of stochastic block models for network with node attributes, NSBM and NDCBM. The structural adjacency matrix follows the strategies of SBM and DCBM, and we assume the covariate follows a mixture of some distributions, while they share the same membership matrix. In other words, the nodes in the same community have same attribute distribution, which is determined by its membership.

Node attributed Stochastic Block Model (NSBM) with n nodes and R communities is parameterized with (M, P, f), where M and P are defined as for SBM, and f is a mixture of R p-dimensional distributions, such as multivariate Gaussian, Binomial, Poisson or others, with the following form

$$f(x) = \sum_{r=1}^{R} 1_r f_r(x; \psi_r)$$

where 1_r is an indicator parameter, and $f_r(x; \psi_r)$ is the node attribute distribution in the r-th community. Then the adjacency matrix and node attribute matrix are generated as follows,

a. The adjacency matrix $A = (a_{ij})_{n \times n}$ is generated as

$$a_{ij} \sim \text{Bernoulli}(P_{g_i g_j})$$

independently for $i \neq j$, otherwise 0, with $i, j \in \{1, ..., n\}$.

b. The $n \times p$ node attribute matrix X is generated as $X_{i\cdot} \sim f_r$ if $g_i = r$, where $X_{i\cdot}$ denotes the ith row of X, and $i \in \{1, ..., n\}$.

Node attributed Degree-Corrected Block Model (NDCBM) with n nodes and R communities is parameterized with (M, P, Θ, f), where M, P and Θ are defined as same as DCBM, and f is a mixture of R p-dimensional distributions with same settings as NSBM. The adjacency matrix and node attribute matrix are generated as follows:

a. The adjacency matrix $A = (a_{ij})_{n\times n}$ is generated as

$$a_{ij} \sim \text{Bernoulli}(\theta_i \theta_j P_{g_i g_j})$$

independently for $i \neq j$, otherwise 0.

b. The $n \times p$ node attribute matrix X is generated as $X_{i\cdot} \sim f_r$ if $g_i = r$.

Similar to the comparison between SBM and DCBM models, the NDCBM model is more capable to fit the real world network data than NSBM model. We proposed different type of algorithms for these models. In general, the degree-corrected version is always preferred in real world applications.

2.3. Collaborative Spectral Clustering Algorithms

We now present the Collaborative Spectral Clustering (CSC) algorithm for attributed network with R underlying communities, where the input data are empirical adjacency matrix A, attribute covariate matrix X, a pre-specified number of communities R, and a value α to balance between structural and attribute information.

Let's start by showing the algorithm procedures for a network under NSBM. The algorithm procedures are showing in Algorithm 1, where the covariate similarity matrix is established with a Gaussian kernel.

Algorithm 1 Collaborative Spectral Clustering

1: **procedure** CSC(A, X, α, R)
2: Obtain the sum of column variance

$$\hat{\sigma}^2 = \sum_{j=1}^{p} \text{Var}[x_{\cdot j}]$$

3: Calculate $K = (k(x_{i\cdot}, x_{j\cdot}))_{n\times n}$, where

$$k(x_{i\cdot}, x_{j\cdot}) = \exp(-\frac{||x_{i\cdot} - x_{j\cdot}||^2}{2\hat{\sigma}^2})$$

4: Obtain leading eigenvectors $U = \{u_1, ..., u_R\}$ of $W = (1-\alpha)A + \alpha K$
5: Apply k-means to U.
6: **end procedure**

Degree heterogeneity is very common in many real world networks and one of the main interests of this paper. To extend CSC algorithm from NSBM to NDCBM, we normalize or regularize the leading eigenvectors of W to remove the degree heterogeneities. There are two main approaches that have been investigated in current literature. (1). Sarkar and Bickel [2015], Joseph and Yu [2016] established theoretical frameworks to analyze the role and impact of *row normalization* in spectral clustering; (2). Jin [2015], Ji and Jin [2016] demonstrated the advantages of taking *element-wise ratio of eigenvectors*.

Algorithm 1 demonstrates the main idea of collaborative clustering, and our focus in this chapter are the generalized versions which are capable to remove the

impact of degree heterogeneities. Algorithm 2 and 3 are two extentions of CSC found by employing two different regularization techniques, where Algorithm 2 uses row-normalization and Algorithm 3 uses the idea of SCORE in Jin [2015] - perform element-wise ratio on the first R leading eigenvectors to remove the heterogeneity effects. We demonstrate that these two algorithms are consistent in both empirical and theoretical aspects.

Algorithm 2 Collaborative Spectral Clustering with Row Normalization

procedure CSC-RN(A, X, α, R)

Obtain the sum of column variance

$$\hat{\sigma}^2 = \sum_{j=1}^{p} \text{Var}[x_{\cdot j}]$$

Calculate $K = (k(x_{i\cdot}, x_{j\cdot}))_{n\times n}$, where

$$k(x_{i\cdot}, x_{j\cdot}) = \exp(-\frac{||x_{i\cdot} - x_{j\cdot}||^2}{2\hat{\sigma}^2})$$

Obtain leading eigenvectors $U = \{u_1, ..., u_R\}$ of $W = (1-\alpha)A + \alpha K$

Obtain U^* s.t.

$$U^*(i,j) = \frac{U(i,j)}{\sqrt{\sum_{j=1}^{R} U^2(i,j)}}$$

Apply k-means to U^*.

end procedure

Algorithm 3 Collaborative Spectral Clustering with SCORE

procedure CSC-SCORE(A, X, α, R)

Obtain

$$\hat{\sigma}^2 = \sum_{j=1}^{p} \mathrm{Var}[x_{\cdot j}]$$

Calculate $K = (k(x_{i\cdot}, x_{j\cdot}))_{n\times n}$, where

$$k(x_{i\cdot}, x_{j\cdot}) = \exp(-\frac{||x_{i\cdot} - x_{j\cdot}||^2}{2\hat{\sigma}^2})$$

Obtain leading eigenvectors $u_1, ..., u_R \in \mathbb{R}^n$ of $W = (1-\alpha)A + \alpha K$

Obtain $n \times (R-1)$ matrix of wise-ratios

$$\hat{E}(i,r) = \frac{u_{r+1}(i)}{u_1(i)}, 1 \le i \le n, 1 \le r \le R-1$$

Apply k-means to $\hat{E}$.

end procedure

2.4. Main Results

The contents of this part are organized as two folds, where we first investigate the consistency properties of CSC algorithm under NSBM settings, while the second part provides more theoretical guarantees within the framework of NDCBM.

2.4.1 Node attributed Stochastic Block Model (NSBM)

We investigate the perfect clustering property under the noiseless version of NSBM. Lemma 1 shows the first R leading eigenvectors U has exactly R different rows corresponding to R communities, which leads to a perfect clustering with k-means.

Lemma 1 *(Perfect clustering under NSBM) Suppose P and P' are non-singular, non-negative, symmetric and irreducible and all eigenvalues of $D((1-\alpha)P+\alpha P')D$ are simple. Let $D((1-\alpha)P+\alpha P')D = U\Lambda U'$ be its eigen-decomposition where Λ is a diagonal matrix and $U'U = UU' = I_R$, then*

(1) For $\mathcal{W} = (1-\alpha)\Omega + \alpha K = (1-\alpha)MPM' + \alpha MP'M$, we have the decomposition

$$(1-\alpha)MPM' + \alpha MP'M = (MD^{-1}U)\Lambda(MD^{-1}U)',$$

and the R nonzero eigenvalues are the diagonal entries of Λ and the corresponding eigenvectors are the columns of $MD^{-1}U$.

(2) The $n \times R$ matrix $MD^{-1}U$ has R distinct rows, each of which corresponds to a community specified in M, i.e., perfect clustering with Ω and $\mathcal{X}$ under NSBM is achieved with CSC algorithm.

Given the perfect clustering property in Lemma 1, we then hope to derive a probabilistic bound on the stochastic $W = (1-\alpha)A + \alpha K$, which guarantees the algorithm working well with empirical data. To achieve this goal, let's first review the following lemma under standard SBM. It provides a probabilistic bound on

empirical adjacency matrix with its noiseless version under SBM, which has been analyzed in literature such as Lei and Rinaldo [2015] and Arias-Castro [2015].

We are now ready to present the first theorem, which guarantees that the empirical W and its noiseless counterpart $E(W)$ are bounded in infinity norm.

Theorem 4 *(Concentration bound on connectivity matrix under NSBM) Let* $W = (1-\alpha)A + \alpha K$, *then* $||W - E[W]||_\infty \leq (1-\alpha)C\sqrt{n}\sqrt{d} + \alpha c\sqrt{\frac{\log p}{p}}$ *with probability at least* $1 - \max\{n^{-r}, n^2 p^{-\rho c^2}\}$

The $||\cdot||_\infty$ represents the infinity norm, where $||x||_\infty = \max(|x_1|, |x_2|, ..., |x_n|)$ if $x = (x_1, x_2, ..., x_n)$. Theorem 4 implies that the absolute value of $W - E[W]$ is bounded with probability at least $1 - \max\{n^{-r}, n^2 p^{-\rho c^2}\}$. We leave the proof to supplementary materials.

Lemma 2 *(Principal subspace perturbation bound) Let* W *and* $E(W)$ *have eigenvalues* $\hat{\lambda}_1, ..., \hat{\lambda}_n$ *and* $\lambda_1, ..., \lambda_n$ *respectively. Let the first* R *leading eigenvectors corresponding to the* R *largest leading eigenvalues be* $\hat{U}$ *and* U *for* W *and* $E[W]$. *Assuming* $\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\} > 0$ *then there exists an orthogonal matrix* $\hat{O}$, *such that,*

$$||\hat{U}\hat{O} - U||_F \leq \frac{2^{3/2}\sqrt{nr}\max\{C\sqrt{n}\sqrt{d}, c\sqrt{\frac{\log p}{p}}\}}{\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}}$$

The CSC algorithm performs k-means on the first R leading eigenvectors of $\hat{W}$, where each row is a point in $\mathbb{R}^K$. Each node is assigned to one cluster where each cluster has a centroid from k-means. Let's define $\hat{C}_1$, ..., $\hat{C}_n$ as the centroid

of the ith node, and let C_1, ..., C_n be the noiseless versions. We conclude that node i is correctly clustered if $\hat{C}_i$ is closer to C_i than any other C_j for any $j \neq i$.

Definition 5 *(Mis-classification) Similar to Qin and Rohe [2013], we define the set of mis-clustering nodes as*

$$Z = \{i : \exists j \neq i, s.t. ||\hat{C}_i O^T - C_i||_2 > ||\hat{C}_i O^T - C_j||_2\}$$

where O^T is a rotation matrix.

Now we present our main result under NSBM, where we build an error bound for the k-means algorithm using first R leading eigenvectors.

Theorem 6 *(Error bound of k-means on leading eigenvectors) Under NSBM with Gaussian distributions in $\mathcal{F}$, assuming $\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\} > 0$, the error bound of k-means on the first R leading eigenvectors is*

$$\frac{||Z||}{N} \leq \frac{64mnr \max\{C\sqrt{n}\sqrt{d}, c\sqrt{\frac{\log p}{p}}\}^2}{N \min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}^2}$$

where $m = \max (M^T M)_{ii}$.

2.4.2 Node attributed Degree-Corrected Block Model (NDCBM)

The theoretical analysis for CSC under NDCBM is more challenging than NSBM, and there are limited theoretical tools in random matrix. To derive the perfect clustering property in this scenario, we employ matrix perturbation techniques.

As in previous part, we also prove a perfect clustering property in lemma 3 and concentration bound on W in lemma 4 under NDCBM.

Lemma 3 *(Perfect clustering with CSC-Row-Normalization under NDCBM). CSC algorithm achieves perfect clustering under NDCBM settings.*

Lemma 4 *(Concentration bound on W) Let* $W = (1-\alpha)A + \alpha K$*, then* $||W - E[W]||_\infty \leq (1-\alpha)\sqrt{4\max_{i=1,...,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)} + \alpha c\sqrt{\frac{\log p}{p}}$ *with probability at least* $1 - \max\{1-\epsilon, n^2 p^{-\rho c^2}\}$*, for a constant c.*

The following two theorems provide theoretical guarantees on principal subspace perturbation bound on W and error bound on k-means with normalized leading eigenvectors in Algorithm 2. Besides, we also prove a perfect clustering property of Algorithm 3.

Theorem 7 *(Principal subspace perturbation bound) Under NDCBM settings, let* W *and* $E(W)$ *have eigenvalues* $\hat{\lambda}_1, ..., \hat{\lambda}_n$ *and* $\lambda_1, ..., \lambda_n$*, respectively. Let the first* R *leading eigenvectors corresponding to the* R *largest leading eigenvalues be* $\hat{U}$ *and* U *for* W *and* $E[W]$ *respectively, and assuming* $\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\} > 0$*, then there exists an orthogonal matrix* $\hat{O}$*, such that,*

$$||\hat{U}\hat{O} - U||_F \leq \frac{2^{3/2}\sqrt{nr}\max\{\sqrt{4\max_{i=1,...,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)}, c\sqrt{\frac{\log p}{p}}\}}{\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}}$$

Let U^* *and* $\hat{U}^*$ *be the row-normalized version of* U *and* $\hat{U}$*, then we have*

$$||\hat{U}^*\hat{O} - U^*||_F \leq \frac{2^{3/2}\sqrt{nr}\max\{\sqrt{4\max_{i=1,\ldots,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)}, c\sqrt{\frac{\log p}{p}}\}}{\Delta\min\{\lambda_{R-1}-\lambda_R, \lambda_R - \lambda_{R+1}\}}$$

where $\Delta = \min_i\{\min\{||U_{i\cdot}||_2, ||\hat{U}_{i\cdot}||_2\}\}$ *is the length of the shortest row in* U *and* $\hat{U}$*.*

Theorem 8 *(Error bound of k-means on normalized leading eigenvectors) Under NDCBM and mixture of Gaussian node attributes, assuming* $\min\{\lambda_{R-1}-\lambda_R, \lambda_R - \lambda_{R+1}\} > 0$*, the CSC algorithm gives a bound on mis-classification error as follows*

$$\frac{||Z||}{N} \leq \frac{64mnr\max\{\sqrt{4\max_{i=1,\ldots,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)}, c\sqrt{\frac{\log p}{p}}\}^2}{\Delta N\min\{\lambda_{R-1}-\lambda_R, \lambda_R - \lambda_{R+1}\}^2}$$

We now add additional theoretical analysis for CSC with SCORE under NDCBM.

Lemma 5 *(Perfect clustering with CSC with SCORE under NDCBM). CSC algorithm achieves perfect clustering under DCBM settings.*

To conclude this section, we have provided a new theoretical framework for community detection under newly developed NSBM and NDCBM models, where we use the recent random matrix theory and matrix perturbation techniques to build a series of concentration bounds, and guarantee the algorithmic performances.

2.5. Simulations and Real World Data Analysis

In this part, we have provided interesting real data examples and extensive simulation studies, showing the empirical consistent performance of our proposed methods.

2.5.1 Simulations

We have conducted extensive simulation studies to investigate the performances of competitive spectral clustering algorithms in different cases, where we consider the following algorithms

(1) CASC algorithm in Binkiewicz et al. [2017].

(2) CSC-RN in Algorithm 2.

(3) CSC-SCORE in Algorithm 3.

Each simulation experiment contains the following steps:

1. *Structure Information.* We first fix $P_{K\times K}$, the community connectivity matrix, and generate $\Theta(i,i) = \theta_i$ with certain strategies for $1 \leq i \leq n$. Generate $A_{ij} \sim Bernouli(\theta_i\theta_j P_{kl})$, where k and l are community labels for i and j respectively. Let $N_0 = (V_0, E_0)$ be the giant component of $N = (V, E)$, and A_0 be the adjacency matrix of N_0.

2. *Attribute Information.* We sample the $n \times p$ design matrix, X, from mixture Gaussian distribution, with the same block/membership structure as the adjacency matrix.

4. *Clustering.* Apply the candidate algorithms with 500 replicates, and measure the oracle performance - the best Hamming Error (HE) from extensive parameter grid search.

Experiment 1. In this experiment, we will investigate how these methods perform in 2-community DCBM samples as the community size ratio $N_1 : N_2 \in \{9 : 10, 8 : 10, 7 : 10, 6 : 10, 5 : 10\}$, where N_k is the number of nodes in the k-th community, and total number of nodes $n = 1200$, and let P be a symmetric matrix where diagonal 0.75 and off-diagonal 0.25. The experiment has three parts, $1(a)$-$1(c)$.

Experiment 1(a). In this experiment, we generate 2-community networks with low heterogeneity effect, $\theta_i \sim U(0, 1)$. This experiment setting is similar to NSBM and our two proposed algorithms perform almost uniformly well as the network becomes unbalanced, see details in Figure 2.1. On the other hand, CASC algorithm could achieve small Hamming Error with balanced networks but gradually gets worse as network becomes unbalanced.

Experiment 1(b). In this experiment, we generate 2-community networks with higher-heterogeneity effects, $\theta_i = 0.02 + 0.48 \times (i/n)^2$, which is more common and realistic in real networks. This scenario is much harder and we can tell that all algorithms become worse as the network shifts from balance to unbalanced, see the second plot in Figure 2.1.

Experiment 1(c). In this experiment, we take binary θ_i with equal probability, $(\theta_1, \theta_2) = (0.02, 0.5)$, which indicates that there are a number of low-degree nodes as well as high-degree nodes in the 2-community networks. This simulation set-

ting is very similar to Experiment 1(b), and our algorithms still maintain a clear advantage over CASC.

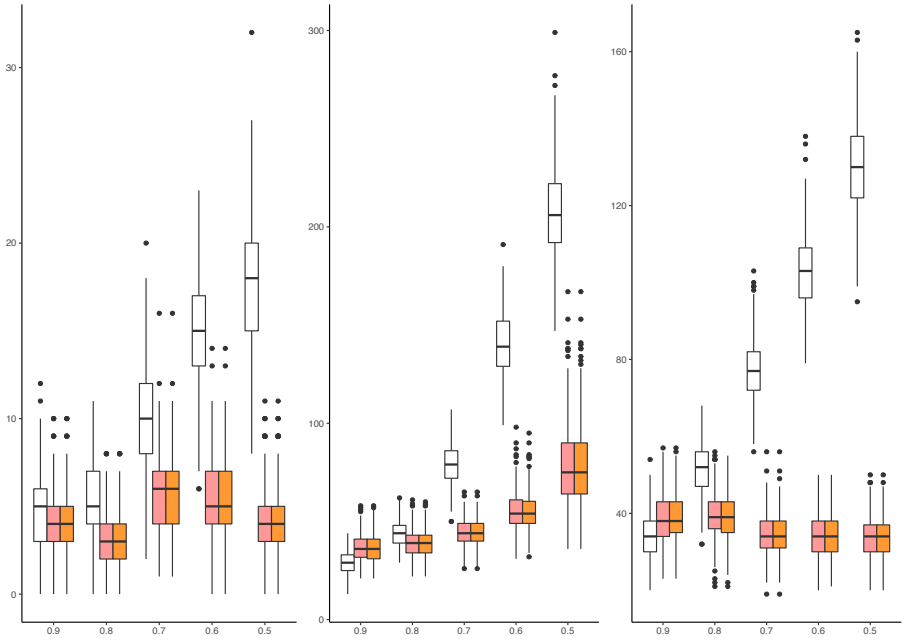

Figure 2.1: Comparison of CSC algorithm with CASC in Experiment 1. (a), (b) and (c) from left to right. White colored box refers to CASC, pink color refers to CSC, and orange color for CSC with SCORE. Under all simulation settings, both CSC and CSC with SCORE are very stable, and uniformly better than CASC in terms of the Hamming Errors.

Experiment 2. In this part, we investigate their performances in unbalanced DCBM samples with different connecting probability ratios taking values between 0 and 1, i.e., $\frac{P_{kl}}{P_{kk}} \in \{0.9, 0.8, 0.7, 0.6, 0.5, 1/3\}$ for $i \neq j$, with $P_{kk} = 1$. We generate $\theta_i = 0.02 + 0.48 \times (i/n)^2$ with heterogeneity effects. In this experiment, we investigate their performances in balanced 2-community NDCBM samples with different connecting probability ratios. The community structure becomes blurred or blended as the connecting probability ratio becomes near to 1, thus it is harder to recover the community labels. The simulation results in Figure 2.2 shows that

our algorithms are still very stable and consistent, performing uniformly better than CASC method.

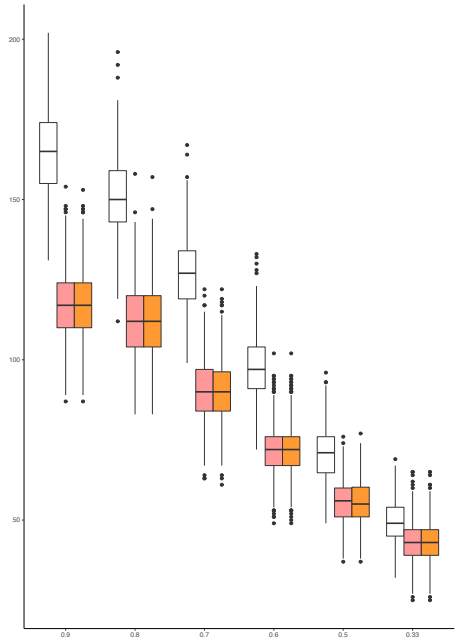

Figure 2.2: Comparison of CSC algorithm with CASC in Experiment 2. Community detection becomes easier from left to right, and our algorithms (pink and orange colored boxes) are uniformly better than CASC (white box).

Experiment 3. In this part, we investigate the methods' performances in NDCBM samples with different heterogeneous θ_i's. Following the Experiment 3 in Jin [2015], we have three parts $3(a) - 3(c)$.

Experiment 3(a). In this experiment, we investigate their performances in 2-community NDCBM samples with $\theta_i = 0.02 + 0.48 \times (i/n)^\gamma$, for $\gamma \in \{0.5, 1, 1.5, 2, 2.5\}$. The results in Figure 2.3 show our algorithms always perform better than CASC under all scenerios.

Experiment 3(b). In this experiment, we are interested in the performances

in 2-community NDCBM samples with $\log(\theta_i) \sim N(0, \sigma^2)$, $1 \leq i \leq n$, where $\sigma = 0.2 \times [1, \sqrt{2}, \sqrt{3}, 2, \sqrt{5}]$. The results in Figure 2.3 demonstrates that these three algorithms have similar performance but ours still maintain clear advantages over CASC.

Experiment 3(c). In this experiment, we assume the balanced 2-community NDCBM samples with binary θ_i with equal probabilities. The number of low-degree and high-degree nodes would become larger as the binary values become more separated, and we set these values vary in this way

$$(\theta_1, \theta_2) \in \{(0.02, 0.5), (0.05, 0.5), (0.1, 0.5), (0.2, 0.5), (0.3, 0.5)\}$$

Figure 2.3 shows a similar results as Experiment 3 (a) and (b).

2.5.2 Real World Data Analysis

We analyze the Paper-Paper citation network data in Statistics community from Ji and Jin [2016]. The data set is based on all published papers from 2003 to the first half of 2012 in four of the top statistical journals: Annals of Statistics (AoS), Biometrika, Journal of American Statistical Association (JASA), and Journal of Royal Statistical Society (Series B) (JRSS-B). For more details of this data set, we refer the readers to the website of Ji and Jin [2016].

Data Preprocessing. We first build the adjacency matrix A for Paper-Paper citation network with $n = 3248$ journal articles, where $A_{ij} = 1$ if paper i cites paper j or vice versa, and $A_{ij} = 0$ otherwise. To establish an attribute matrix

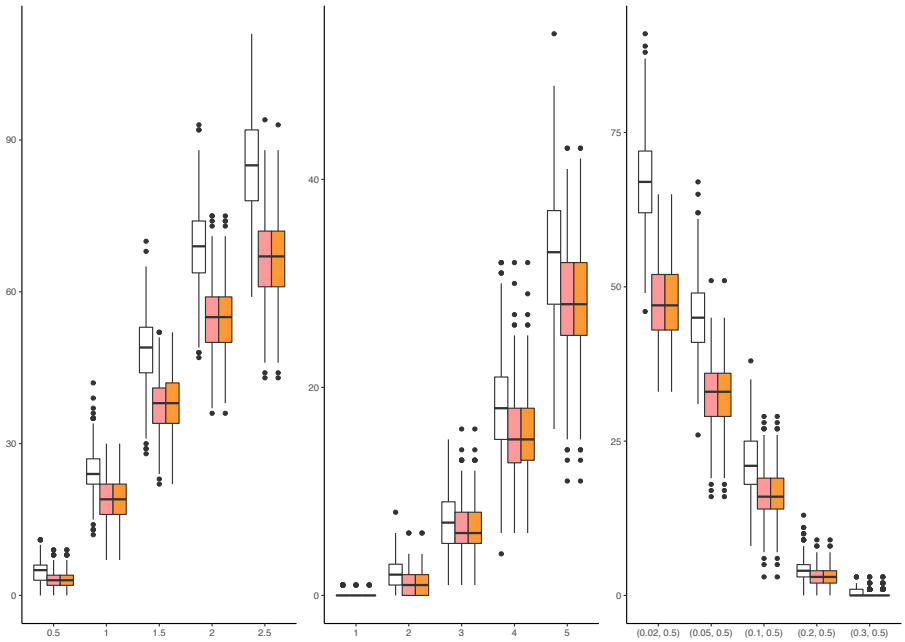

Figure 2.3: Comparison of CSC algorithm with CASC in Experiment 3
(a), (b) and (c) from left to right. Both CSC (pink box) and CSC with SCORE (orange box) are very stable, and uniformly better than CASC (white box) in terms of the Hamming Errors.

X, we first obtain the abstracts for each paper, then use a *bag-of-words* model to retrieve the frequency of each word. Prior to constructing the *paper-word* matrix, a set of extensive data cleaning steps such as removing numbers, punctuations, stop words, white space, etc., was conducted. We then use tf-*idf* algorithm, one of the most popular term-weighting schemes in the context of information retrieval and text mining, to formalize the ultimate attribute matrix X, where we intend to reflect how important a word is to an abstract in a collection of papers.

Results. We comprehensively run our algorithms with different pre-specified community numbers, from 2 to 6, then list the top 5 papers by degree in each community. The results are more interpretable compared to simply performing community detection with adjacency matrix. Due to the space limit, we show and compare only two results with community number $R = 3$ with $\alpha = 0$ and $\alpha = 0.8$. For the first case, the results using CSC with SCORE are shown in Table 2.1. It is easy to tell that only the second group has a clear interpretation - the *lasso* related community, and the other two are somehow mixed together with communities such as *Dimension Reduction*, *Functional Data Analysis*, *Bayesian Statistics*, etc.

ID	Title	Author	Year
1	Covariance regularization by thresholding	Bickel and Levina	2008
1	On properties of functional principal components analysis	Hall and Hosseini-Nasab	2006
1	Marginal nonparametric kernel regression accounting for within-subject correlation	Wang	2003
1	Covariance matrix selection and estimation via penalised normal likelihood	Huang et al	2006
1	Functional data analysis for sparse longitudinal data	Yao et al	2004
2	Regularization and variable selection via the elastic net	Zou and Hastie	2005
2	Nonconcave penalized likelihood with a diverging number of parameters	Fan and Peng	2004
2	The Dantzig selector: statistical estimation when p is much larger than n	Candes and Tao	2005
2	High-dimensional graphs and variable selection with the lasso	Meinshausen and Bühlmann	2006
2	The adaptive lasso and its oracle properties	Zou	2006
3	Contour regression: a general approach to dimension reduction	Li et al	2005
3	Order-based dependent Dirichlet processes	Griffin and Steel	2006
3	Sufficient dimension reduction via inverse regression: a minimum discrepancy approach	Cook and Ni	2005
3	The positive false discovery rate: a Bayesian interpretation and the q-value	Storey	2003
3	Strong control, conservative point estimation and simultaneous conservative consistency of FDR	Storey et al	2004
3	A stochastic process approach to false discovery control	Genovese and Wasserman	2004

Table 2.1: Community detection results from *CSC-SCORE* with $\alpha = 0$

Community 1 consists of three different small groups, *Functional Data Analysis*, *Nonparametrics* and *Covariance Estimations*. Community 2 is a well built group with lasso-related papers. Community 3 is a mix of *Dimension Reduction*, *Multiple Testing*, and *Stochastic Process*.

However, when we shift our direction a little to the attribute information, an interesting result appears when using $\alpha = 0.8$. Table 2.2 shows three clear communities from our algorithm, *Bayesian Statistics*, *Nonparametrics* and *lasso* related topics. The result is more interpretable and understandable than the case with $\alpha = 0$. Readers might also be interested in another extreme case: what are the results when $\alpha = 1$? In fact, this scenario cannot help us to identify the hub papers because we do not use any actual connectivity relations - no intellectual but all plain literal level information has been employed, which conflicts with our original intentions. Thus, we do not recommend to use this extreme case when dealing with scientific collaboration networks.

ID	Title	Author	Year
1	Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models	Omiros Papaspiliopoulos and Roberts	2007
1	An ANOVA model for dependent random measures	Iorio et al	2004
1	Bayesian nonparametric spatial modeling with Dirichlet process mixing	Gelfand et al	2005
1	Hierarchical Dirichlet processes	Teh et al	2005
1	Bayesian density regression	Dunson and Pillai	2007
1	A method for combining inference across related nonparametric Bayesian models	Müller et al	2004
2	Empirical Bayes selection of wavelet thresholds	Johnstone	2005
2	Covariance matrix selection and estimation via penalised normal likelihood	Huang et al	2006
2	New estimation and model selection procedures for semiparametric modeling in longitudinal data analysis	Fan and Li	2004
2	One-step sparse estimates in nonconcave penalized likelihood models	Zou and Li	2008
2	Nonconcave penalized likelihood with a diverging number of parameters	Fan and Peng	2004
3	A stochastic process approach to false discovery control	Genovese and Wasserman	2004
3	Regularization and variable selection via the elastic net	Zou and Hastie	2005
3	The Dantzig selector: statistical estimation when p is much larger than n	Candes and Tao	2005
3	High-dimensional graphs and variable selection with the lasso	Meinshausen and Bühlmann	2006
3	The adaptive lasso and its oracle properties	Zou	2006

Table 2.2: Community detection results from *CSC-SCORE* with $\alpha = 0.8$

Three communities are well built, where Community 1 is *Bayesian Statistics*, Community 2 is *Nonparametrics*, and Community 3 is lasso related group.

2.6. Discussion

In this chapter, we propose a set of novel algorithms for community detection in networks with node attributes, where the attributes or covariates could be generated from different information sources such as text, image, video, etc. A theoretical framework with recent results in random matrix theory and matrix perturbation techniques is developed under NDCBM. We then provide extensive simulation studies to compare our methods with the competing CASC algorithm, showing that ours have great advantages especially in networks with unbalanced communities or degree heterogeneities. In fact, the Gram matrix under CASC setting is actually a special case of our kernel methods, and the main shortcoming of Gram matrix is the error accumulation as network becomes unbalanced with weak signals. The classical SCORE method in Jin [2015] has proved that taking element-wise ratio on leading eigenvectors has great benefits for removing heterogeneity effects, and this advantage is maintained in our method, but we do not observe any significant difference between row-normalization and element-wise ratios approaches based on the simulation results. In practice, both CSC and CSC-SCORE can achieve satisfying accuracy in unbalanced and moderate heterogeneous networks.

A key issue in our algorithms is the choice of tuning parameter, which is also an open question in many unsupervised learning algorithms. Instead of proposing a method to choosing an optimized value of α, we interpret it as a pre-specified belief of the weights for different information sources. For instance, in our real data example, one paper does not necessarily cite all papers in its community, but its

corresponding covariate (paper abstract in this case) would definitely have high connectivity with most papers within the community. Thus, if we use a higher value of α, the result will favor paper abstract information compared to actual citations. Another issue in such algorithms is the choice of community number R, which is also largely an open question in this area. Some researchers suggested to use scree plot-style method in Ji and Jin [2016] or cross validation techniques in Chen and Lei [2017] to choose the optimal number of communities, but we do not focus on this aspect in this work. For the future directions, the above open questions, along with network dynamics, would be worth more attention.

Chapter 3

Testing Existence of Community

3.1. Introduction

Community, one of the most critical topology structure in networks, is usually defined as a group or cluster of nodes that have higher and denser connections compared to between-group relations. The community has different interpretations in different disciplines. In a social network, a community may refer to a cohort of students who study in the same department or a group of players in the same sports team. In biology, a community of proteins may have specific functionalities compared to another collection of proteins. In political science, Adamic and Glance [2005] studies the linking patterns and discussion topics of political bloggers to understand the community structure. Figure 3.1 shows two examples of communities. The Zachary's karate club network data in Zachary [1977] in the left panel is a social network of friendships between 34 members of a karate club

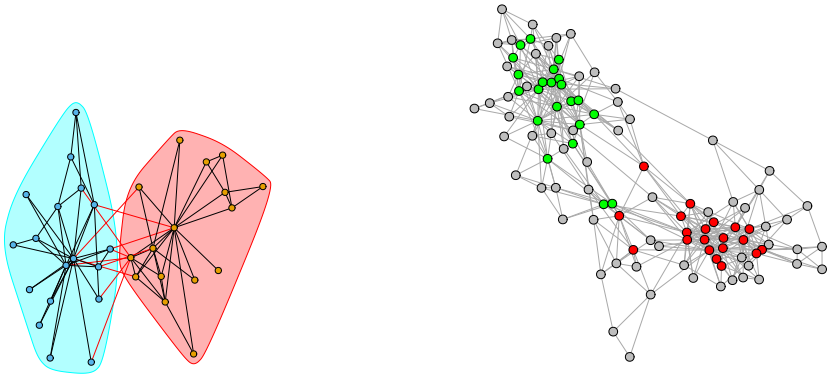

Figure 3.1: Examples of communities. *Left*: two communities in Zachary's karate club. *Right*: two small communities (hubs) surrounding by backgrounds in political books network

at a US university in the 1970s. Two communities are outlined with different colors. The Political Books network data in the right panel shows a network of books about US politics published around the time of the 2004 presidential election on Amazon.com, the edge represents co-purchasing of the same buyers. Zhao et al. [2011] extracted two major communities surrounding by background nodes, where the grey nodes refer to the backgrounds and green/red colored nodes are two communities or hubs.

Investigation of community structures can help researchers to understand the internal connectivity and external association of the units (nodes) in their studies. In the past decades, we witnessed dramatic methods in this field, to understand

the structure, evolution, and dynamics of communities in networks. We usually refer to the methodologies, procedures, or algorithms to discover communities as *community detection* methods.

Community detection is an important research question in machine learning and network data analysis fields. Researchers from different disciplines have proposed various of algorithms, such as spectral clustering methods (Jin [2015]; Ji and Jin [2016]; Bickel and Chen [2009b]), maximum likelihood-based approaches (Newman [2006b]; Zhao et al. [2012a]), deep learning based techniques in Chen et al. [2019], and many others. However, a more fundamental question we may ask is, does community or cluster exist in the observed network? If there is no community in the network, any detection method would not be legitimate. From a statistical perspective, a hypothesis testing framework is desirable to answer this question. There are scattered examples to study the testing problems in networks. Earlier works, such as Radicchi et al. [2004] and Lancichinetti et al. [2011], treat this problem from the definition of community and proposed a set of heuristic algorithms to identify and assess the community patterns. Recently, Wang and Bickel [2017] presented a likelihood ratio test approach to test the heterogeneity in networks with block structure, Banerjee and Ma [2017] derived an optimal test statistic under stochastic block model in a spectral approach. On the other hand, some more recent works try to formulate the problem by counting different types of sub-structures in the network. These sub-structures include triangles, triplets, quadrilateral, k-cliques, etc. For example, Gao and Lafferty [2017] proposed a test statistic with expected densities of edges, vees, and triangles, to test the number

of clusters in a given network, they also provided proof for asymptotical normality under both null and alternative hypotheses. Jin et al. [2018] designed a testing statistic with graphlet counting, where graphlet usually refers to the small connected non-isomorphic induced subgraphs of a large network. Most of the existing works can assess the significance of a given subset of the network, or determine the number of communities, while they are not designed for testing the existence of communities which could appear anywhere in the network. We, on the other hand, proposed a different but effective approach to tackle this problem with local peak hunting and search. Our method operates by investigating local peaks with higher densities in a network, and test the significance of the largest order statistic of these peaks. Kolaczyk [2009] argued that the global clustering coefficient, or transitivity, is a test statistic to evaluate the existence of clusters in networks. A null distribution is simulated by permuting the observed network to random graphs. We will show detailed comparison between this approach (GlobalCC, short for global clustering coefficient) and our new proposed algorithm.

The rest of this chapter is organized as follows. Section 3.2 contains the fundamental definitions and assumptions to be used throughout this chapter. We present the PET test statistic and discussions in Section 3.3. We use extensive experiments to compare the performance of our method and others under different settings in Section 3.4. The results with ten real-world network data sets in different categories are summarized in Section 3.5.

3.2. Preliminaries

We present the necessary definitions and mathematical notations in the following numbered list.

1. Let $G = (V, E)$ be an undirected and unweighted graph with node set V and edge set E. Let $|E|$ and $|V|$ denote the number of edges and nodes respectively. The binary symmetric adjacency matrix is then denoted as A.

2. Let $\deg(v_i)$ denote the degree of node i, i.e., the total number of edges connecting node i. It simply equals the row or column sum of ajacency matrix A if there is no self-connections.

3. We define G_i as a sub-graph of G, which consists of a center node v_i and all its first-order neighbors.

4. **Erdős-Rěnyi Random Graph.** A Erdős-Renyi random graph with n nodes is defined as $G(n, p)$, with $0 \leq p \leq 1$, where each possible edge has probability p of existing independently. For any particular node, the distribution of degree, $\deg v_i$, is binomial.

5. **Clustering Coefficient.** Clustering coefficient, or transitivity, is a global measure to assess the degrees to which nodes that can form a cluster structure. It's usually defined as the ratio of the number of triangles and the number of triplets.

3.3. PET Statistic

The idea of testing the existence of a cluster or community is to identify the signal while removing the noise as much as possible. We observe, from many empirical studies, the networks with community structures usually exhibit extreme local densities, i.e., the distribution of the largest order statistics of density is different from the networks without clusters. We refer to such high-density local sub-networks as peaks, and our goal is to detect the existence of such peaks. The concept of peak hunting is developed and analyzed in Rodriguez and Laio [2014], where they designed a search-and-find density peaks strategy to formulate an algorithm for general purpose clustering analysis. The idea in Rodriguez and Laio [2014] is to iteratively search and find the peaks with higher local densities and assign the cluster assignments by comparing certain distance metrics. Inspired by this approach, we characterize the problem of testing global community existence with a Peak dEnsity raTio (PET) statistic. The principle of PET statistic is to identify the largest signal in a given network and compare with the random counterpart given by Erdős-Renyi random networks. Given a sub-network $G_i = (V_i, E_i)$, which is constructed with a center node v_i and its first-order connected neighbors, the local density d_i, is defined as the ratio of number of edges to the number of nodes, i.e., $d_i = \frac{|E_i|}{|V_i|}$. Other local density measures are also suitable to our framework, examples including (a). graph density, defined as $\frac{|E_i|}{|V_i|\times(|V_i)-1}$, (b). local transitivity, a local version of clustering coefficient that defined in Section 3.2, and (c). higher-order clustering coefficient defined in Yin et al. [2018].

The hypothesis testing problem can be formulated as follows,

H_0: the network contains no community

H_1: the network contains at least one community

The procedure to calculate PET statistic is described as follows

1. For each node v_i in G, calculate density of the subgraph formed with center node v_i and its first order neighbors, denote as $d = (d_1, ..., d_n)$

2. Calculate PET statistic for a network as the ratio of the largest d and the mean of vector d, i.e., $w = \max\{d\}/\bar{d}$, or $d_{(n)}/\bar{d}$, where $d_{(n)}$ denotes the n-th (or largest) order statistic.

3. Estimate the empirical connecting probability $\hat{p} = \frac{\sum A_{ij}}{n(n-1)}$. Simulate N Erdős-Renyi network samples with $G(n, \hat{p})$, and calculate the t statistic for each network sample, denote as a vector W.

4. Calculate the simulated p-value $p = \frac{\sum I_{[W \geq w]}}{N}$

In many real large networks, the signal may not appear in the last order statistic, so a k-rank procedure can be adopted as follows. Instead of examining the distribution of the largest local density, we may investigate the top k largest values and their simulated p-values given a null distribution of the random network. Meanwhile, the cost by adding more order statistics to consider is the trade-off between Type I and Type II errors, depending on the specific strategies. For example, if we looked at the top three peak densities and calculated three corresponding p-values, strategies to determine the significance could be (1). Majority vote, the majority of the above tests being significant leads to a conclusion of significance,

(2). Union, all tests being significant leads to a significance, and (3). Intersect, at least one significant test is enough. Based on our observation with many real-world networks, the last strategy or the 1-rank test is sufficient in balancing Type I and Type II errors.

3.4. Simulation study

We implement an extensive simulation study to show the performance of PET statistic under different scenarios. We consider two major settings of the observed networks, *Degree-Corrected Block Model* (DCBM), and *Backgrounds Model* (BG). For the first setting, we fix the between community connecting probability as $\log(n)/n$ in a network with n nodes. We further define the measure of signal-to-noise ratio (SNR) as the ratio of within-community connection probability to between-community connection probability. Abbe [2017] presented a comprehensive reviews for the exact recovery of classical clustering algorithms, and most of the algorithms can achieve exact recovery when the connecting probability is at least $\Omega(\log(n)/n)$, thus we use $\log(n)/n$ as the baseline probability for between-community connections. For the second setting, the background model, we assume that there is only a fraction of the nodes that form a community while all other nodes are backgrounds, which means the connection within the small group is dense, and all the rest of the connections are relatively weak, i.e., with probability of $\log(n)/n$.

We compare our method to two additional benchmarks, one is the algorithm

proposed in Kolaczyk [2009], GlobalCC, while another is a variant of PET statistic with local clustering coefficient (PET_LCC, short for PET statistic with local clustering coefficient). We present the detailed experiment settings in the following numbered lists.

1. *DCBM.* This is the most frequent and realistic scenario in real life network data. Three different types of heterogeneity distributions are considered. (1). Binary. The θ_i is assumed to have two values with equal probabilities. We test three different cases, $(0.01, 0.5)$, $(0.02, 0.5)$, and $(0.05, 0.5)$. (2). Polynomial. We random sample θ_i from $0.02 + 0.48 \times (i/n)^{1.5}$ and $0.02 + 0.48 \times (i/n)^2$, where $i \in \{1, 2, ..., n\}$. (3). Pareto. Pareto distribution is one type of power-law probability distribution, which is usually used to describe the wealth in society. We consider three different shape and scale parameters, i.e., Pareto$(4, 0.375)$, Pareto$(4, 0.275)$, and Pareto$(4, 0.175)$, where a is the shape parameter and b is the scale parameter in Pareto(a, b). For three scenarios above, the choice number of nodes and SNR is the same as in SBM setting.

2. *BG.* We first simulate the background network with different sizes (same as SBM setting) with the connection probability being $\log(n)/n$. We then simulate a small community of size being 5%, 10%, 15%, and 20% of the total nodes n. Furthermore, we assume the small window follows an SBM setting, and the heterogeneity parameter follows a uniform distribution. SNRs and the total number of nodes follow the same setting as in SBM and DCBM.

We create two sample adjacency matrix in Figure 3.2 to show difference among

three model settings above. The DCBM setting exhibits severe degree diversities. BG is the most challenging scenario where the signals only live in a small window of the network. We present the experiments results in the next three sections. In each setting, we first evaluate the performance of three methods when the null hypothesis is true, i.e., when the SNR equals to 1, and the simulated network has no community structure - we then check the Type I errors. Secondly, we compare the performance by calculating the empirical power of the tests given the alternative hypothesis is true. We generally prefer the methods that have good control over Type I error while having relatively good performance on the Type II error, or empirical power.

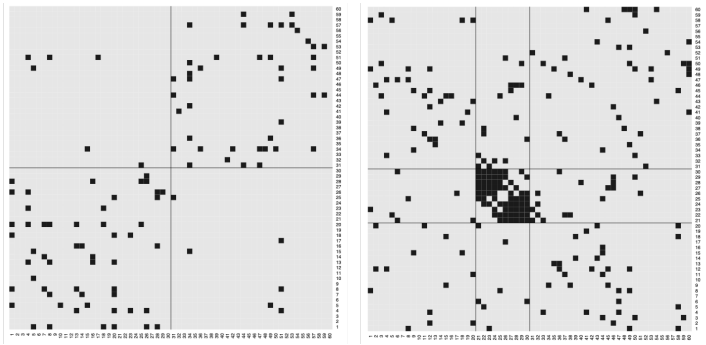

Figure 3.2: From left to right: examples DCBM and Block Model with Backgrounds

3.4.1 Results under DCBM

The DCBM model adds extra heterogeneity parameters, θ_i, to each node in the network; nodes with higher θ have larger probabilities to connect with other nodes. In the social network, the nodes with larger θ value are viewed as opinion leaders. DCBM is more flexible and close to the network data in real life. Such degree incongruity generally introduces difficulty when identify the network topology, see more detailed theoretical analysis of this phenomena in Jin [2015]. We assess these test statistics in DCBM samples with different heterogeneity levels. Three heterogeneity distributions of θ are considered: binary, polynomial, and Pareto distribution. Figure 3.3 shows two density plots with samples from polynomial and Pareto distribution. The Pareto distribution is one type of power-law type distributions, which usually exhibits high heterogeneity and is used to describe the wealth in society. The θ sampled from both polynomial and binary cases is less heterogeneous.

Figure 3.4 and Figure 3.5 shows the results under DCBM with relatively weak heterogeneity effects, i.e., binary and polynomial type of θ. GlobalCC cannot control the Type I error in both cases with different parameter settings, while PET_CC has the best performance over Type I error. Our PET statistic dominates PET_CC uniformly in all sized networks and SNRs. However, if we consider the Type I and II error trade-off, the PET_CC is recommended for large networks with large SNRs.

Figure 3.6 shows a different pattern than the results in previous settings. GlobalCC has better control over the Type I error, compared to previous simulation

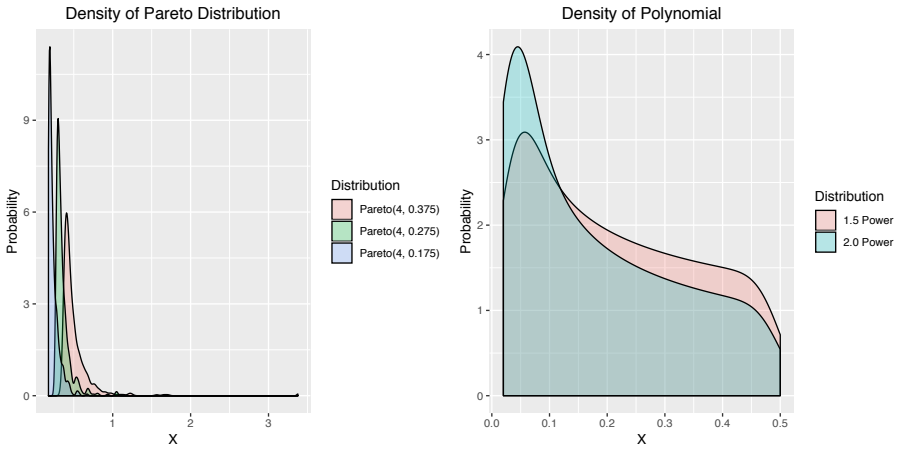

Figure 3.3: Density plots of samples from Pareto (left) and polynomial (right) distributed θ

settings, in large heterogeneity cases, even though it's still the worst one over the other two. The PET_CC is the worst player under this setting - it can control the Type I error as low as zero, in the meanwhile, it has the worst performance in terms of the empirical power in all settings. Our PET statistic has the right balance between Type I error and empirical powers.

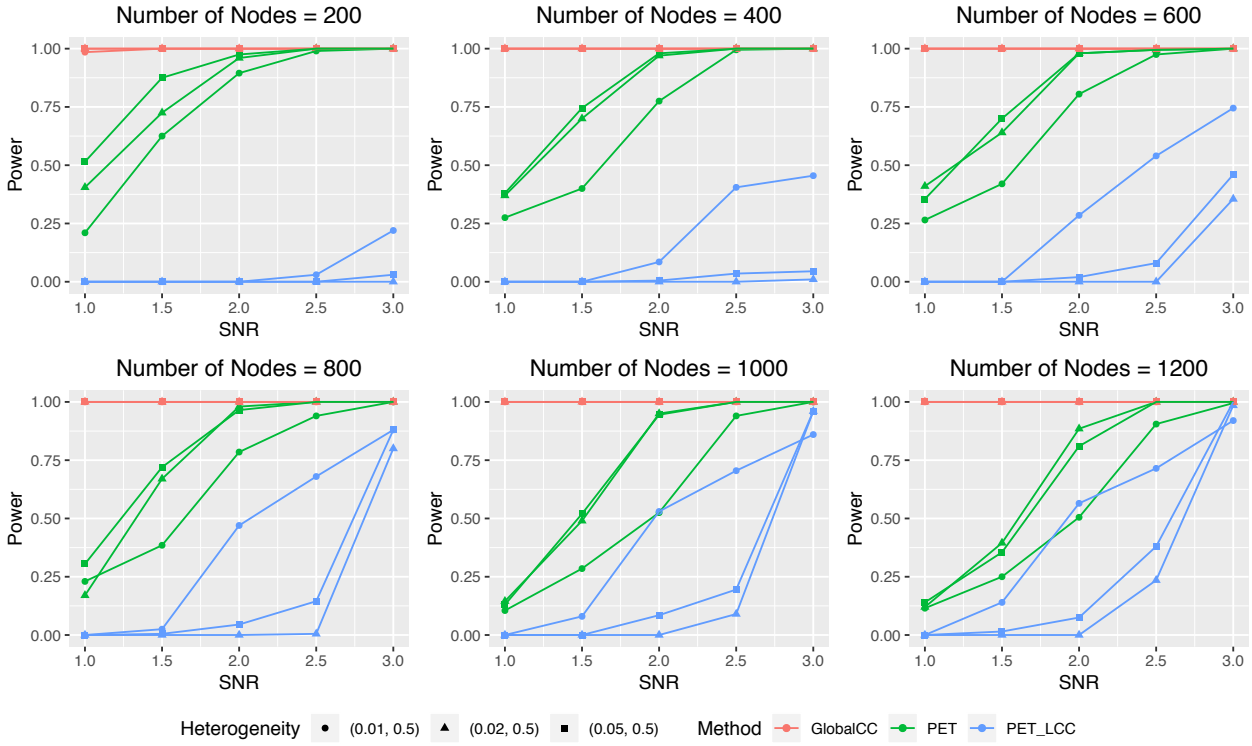

Figure 3.4: Performance comparisons over three test statistics with different SNR under DCBM model with binary θ

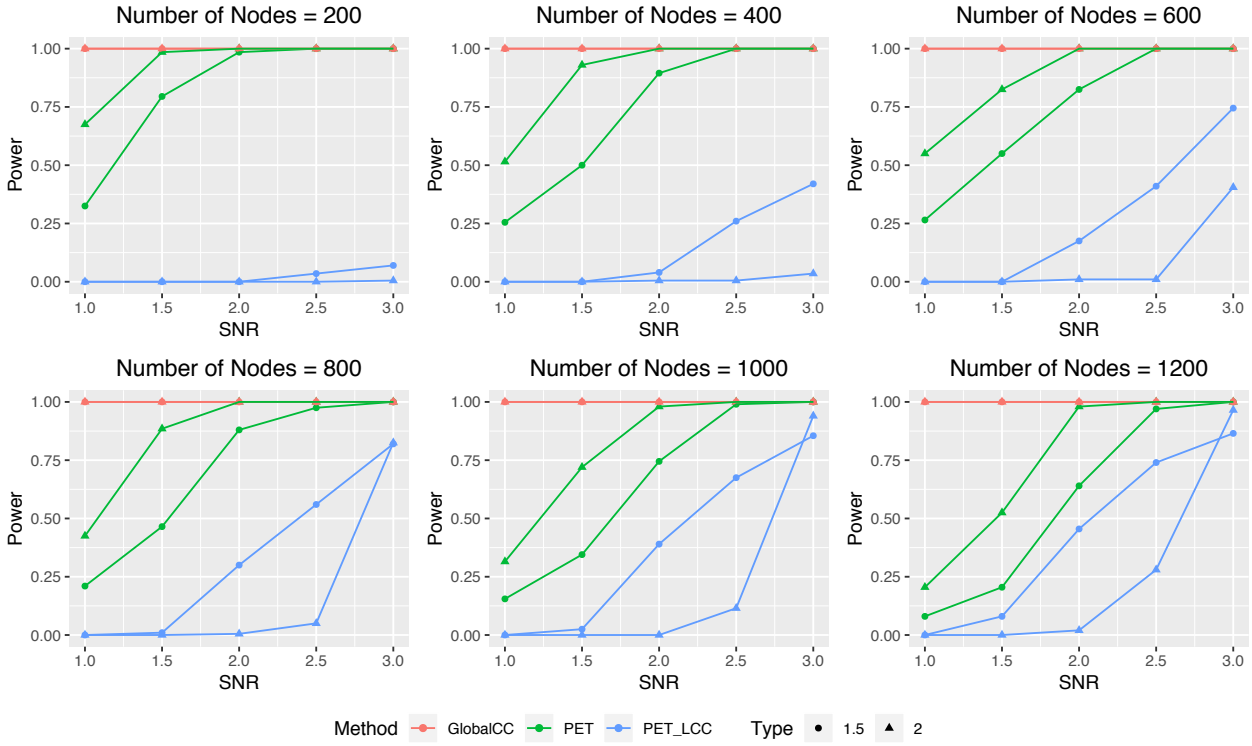

Figure 3.5: Performance comparisons over three test statistics with different SNR under DCBM model with polynomial θ

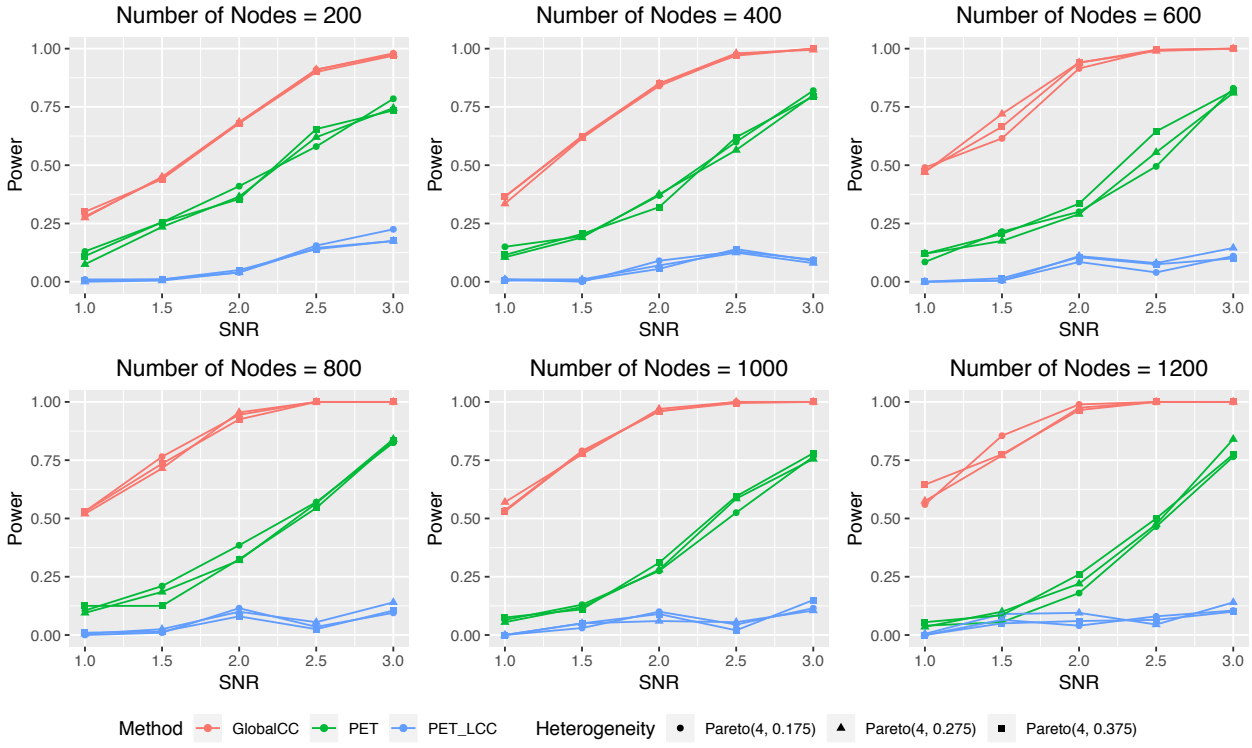

Figure 3.6: Performance comparisons over three test statistics with different SNR under DCBM model with Pareto θ

3.4.2 Results under BG

The performance of GlobalCC, PET, PET_LCC with different window sizes is shown for different network sizes in Figure 3.7. First, none of the methods works well in that the type I error is out of control, except for PET in the first and second panels, suggesting that detection of one community from a noisy background is an extremely challenging problem. It probably should not be surprising that the high-density subgraphs in large networks with no community can be easily taken as a community. Second, combined with our observations for the SBM and DCBM, GlobalCC should not be used at all in any setting. Third, the PET works somehow better than the other two methods, even though it is far from ideal and leaves a lot of space for improvement in future studies.

In some cases of the simulation study, the Type I errors are not controlled well, which is due to the fact that the Erdős-Renyi network samples are approximation but not the true null distribution when θ is not 1, we choose Erdős-Renyi model for its simplicity and computation convenience.

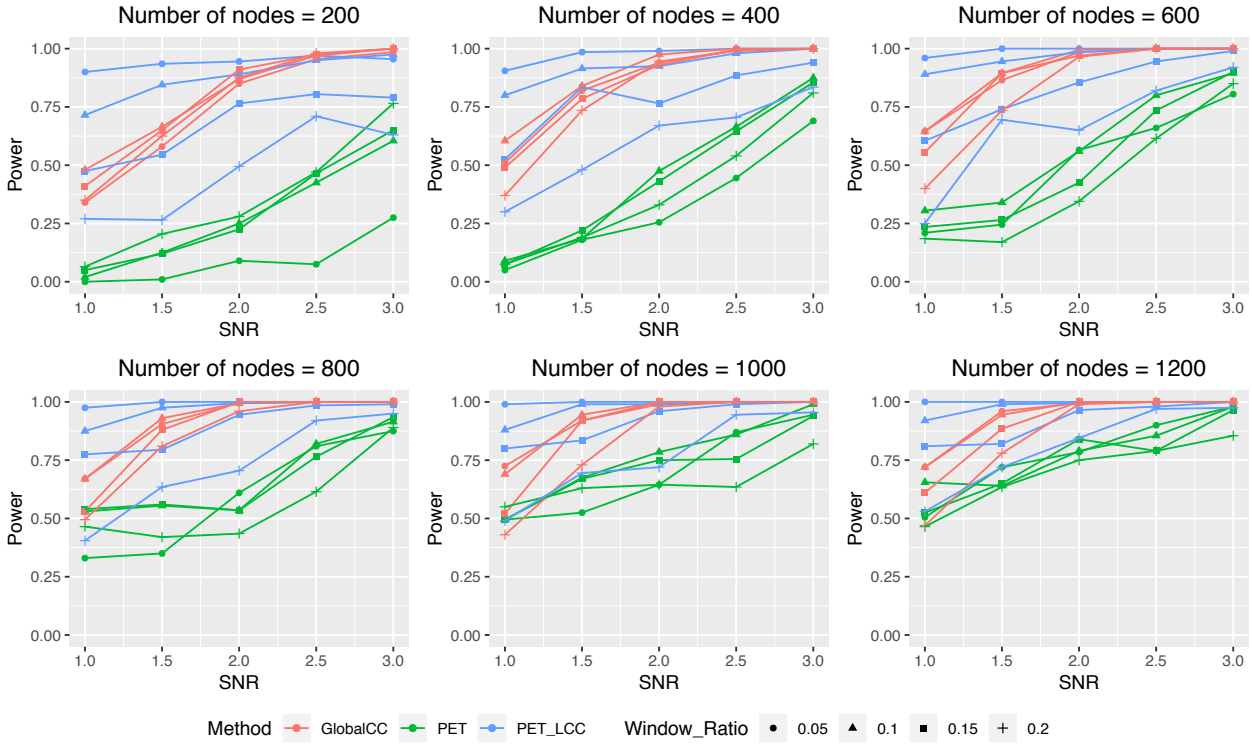

Figure 3.7: Performance comparisons over three test statistics with different SNR under Backgrounds setting

3.5. Real Data Analysis

We implement three test statistics with ten real-world network data sets (which contains real community structures) from seven different disciplines including Social, Transportation, Biology, Politics, Electricity, Academia collaborations, and Novel.

For Social and Academia data sets, UK Faculty in Nepusz et al. [2008] and Network Scientist in Newman [2006a], the node represents the researcher/faculty, and the edge is the corresponding social interaction measures. The three Biology network data sets, Yeast in Von Mering et al. [2002], Dolphin in Lusseau et al. [2003], and C. Elegans in Watts and Strogatz [1998] are protein-protein interaction, New Zealand dolphin living community, and neural network of C. Elegans. The transportation network, US Airports, described the connecting associations between 755 US airports in 2010. The Politics network, Political Blogs in Adamic and Glance [2005], presented the linking relationships among 1,490 bloggers from different political parties. The Electricity network, Power Grid in Watts and Strogatz [1998], is a sparse network representing the topology of the Western States Power Grid of the U.S. We also included two networks from novels. The *Les Miserables* in Knuth [1993] describes the co-appearance network of characters in the novel *Les Miserables*, and *David Copperfield* in Newman [2006a] is a network of common adjectives and nouns in the novel *David Copperfield* by Charles Dickens.

A summary statistics as well as our testing results (p-values) are presented in Table 3.1. The results agreed with the observations in simulation study - in networks with relatively high heterogeneities, the PET_CC cannot discover the signal

in all sized networks; All real-world data sets in our study have distinct communities according to the literature, PET test can perfectly produce the significance, but PET_CC did not.

Data	#Node	#Edge	Type	PET	PET_CC
UK Faculty	81	577	Social	0.02	0.38
Yeast	2,617	11,855	Biology	0.00	1.00
Dolphin	62	159	Biology	0.02	1.00
C. Elegans	297	2,152	Biology	0.00	1.00
US Airports	755	4,677	Transportation	0.00	1.00
Political Blogs	1,490	16,726	Politics	0.00	1.00
Power Grid	4,941	6,594	Electricity	0.00	0.95
Network Scientist	1,589	2,742	Academia	0.00	1.00
Les Miserables	77	254	Novel	0.00	1.00
David Copperfield	112	425	Novel	0.00	0.35

Table 3.1: Results of test statistics with 10 real world network data in 7 different disciplines

Chapter 4

A Mixture Coalescent Model for Phylogenetic Networks

4.1. Preliminaries

Phylogenetics is a discipline studying the evolutionary history and process of biological entities such as species, populations, individuals or genes. Phylogenetic trees are usually constructed to describe the evolutionary history of species. Figure 4.1 shows an example of a rooted species tree for three life domains, Bacteria, Archaea, and Eukaryota. Each node in the species tree with descendants represents the inferred most recent common ancestor of those descendants, and the edge (branch) lengths can be interpreted as time estimates of evolution.

A species tree reflects the pattern of branching of species lineages via the process of speciation. Figure 4.2 shows a simple species tree involving only two species.

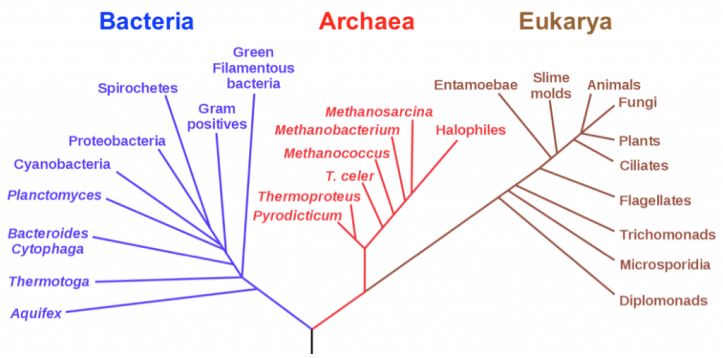

Figure 4.1: A speculatively rooted tree for rRNA genes, showing the three life domains: bacteria, archaea, and eukaryota. Wikipedia [2019]

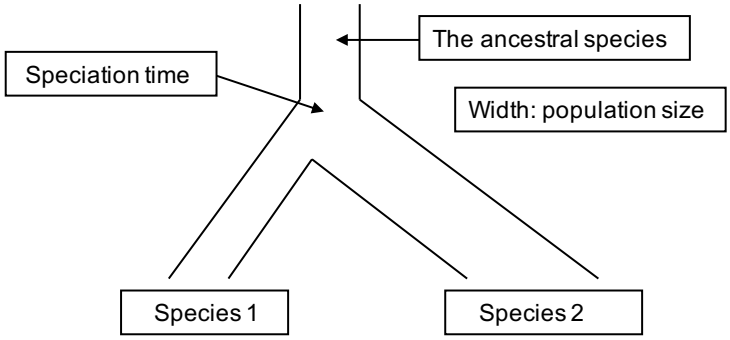

Figure 4.2: Illustrative concept of species tree.

Each species is parameterized by (τ, θ). The branch length τ in a species tree represents the divergence time and the branch width θ is the species population size. The ancestral species evolves to two separated populations (i.e., species 1 and 2) with different sizes in Figure 4.2.

A gene tree is a phylogenetic tree constructed from a single gene from each of the species under study. Figure 4.3 shows an example of gene tree construction from DNA sequences of four species. The terminal nodes (S1, S2, S3, and S4)

represent single genes from four species, and the interior nodes represent genetic divergence events that ultimately go back to a DNA or RNA replication event. Such events may correspond to but typically precede speciation events (Maddison [1997]). The branch length in a gene tree represents the divergence time of gene, while the branch width is not specified. The speciation events in species trees are always later than the corresponding genetic events (which trigger speciation events) in gene trees. Many algorithms can build gene trees from sequence data, such as Distance methods (Li [1981]), maximum likelihood methods (Felsenstein [1983]; Guindon et al. [2005]; Stamatakis [2014]), and Bayesian approaches (Drummond and Rambaut [2007]). Earlier works in Pamilo and Nei [1988] show that a gene tree does not necessarily agree with the actual evolutionary pathway of the species (species tree).

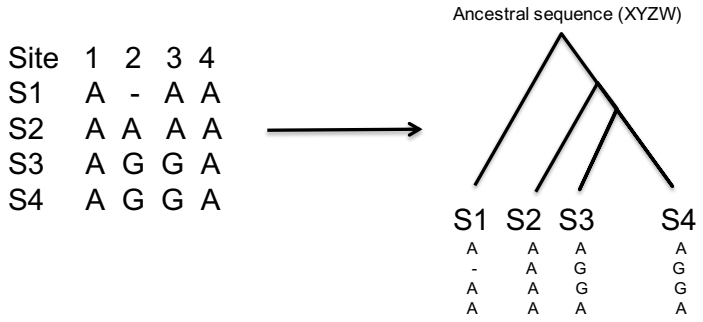

Figure 4.3: Illustrative concept of gene tree.

It is well known that gene trees are estimated from sequence data, while there are mainly two approaches to estimate the species trees, directly compute from DNA sequences or infer from gene trees. The first approach works by first concatenate DNA sequences from different genes to a super gene; then a species tree

is constructed from the given super gene. This approach is problematic for inconsistency when the assumption of homogeneous gene trees is seriously violated (Kubatko and Degnan [2007]). Furthermore, the tree produced by the concatenation method is a tree of sequences, other than a tree of species. The second approach, estimate species tree from gene trees, is preferred in current literature, see Liu and Pearl [2007]; Edwards et al. [2007]. An illustrative example of the estimation process from DNA sequence data to gene trees, then from gene trees to species trees is presented in Figure 4.4, three gene trees with different topology are constructed from different DNA sequence data, then a species tree can be estimated from the given gene trees.

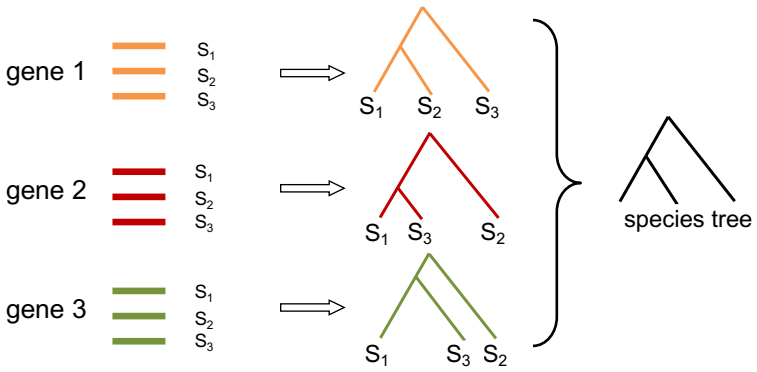

Figure 4.4: Illustrative example of gene tree and species tree estimation

The Bayesian hierarchical model is the most attractive approach to achieve the goal in Figure 4.4, and many efficient algorithms are proposed in recent years, such as MP-EST in Liu et al. [2009] and NJst in Liu and Yu [2011a]. Figure 4.5 shows the overview of such hierarchical structures, $P(G|S)$ represents the probability of gene trees conditional on a given topology of species trees, and $P(D|G)$ represents

the probability of DNA sequences given a set of gene trees. Thus we can fit this structure to a Bayesian hierarchical model, as showing in Figure 4.6. The Bayesian estimation of species trees can be written as

$$f(S|G) = \int_G f(G|D) f(S|G) dG$$

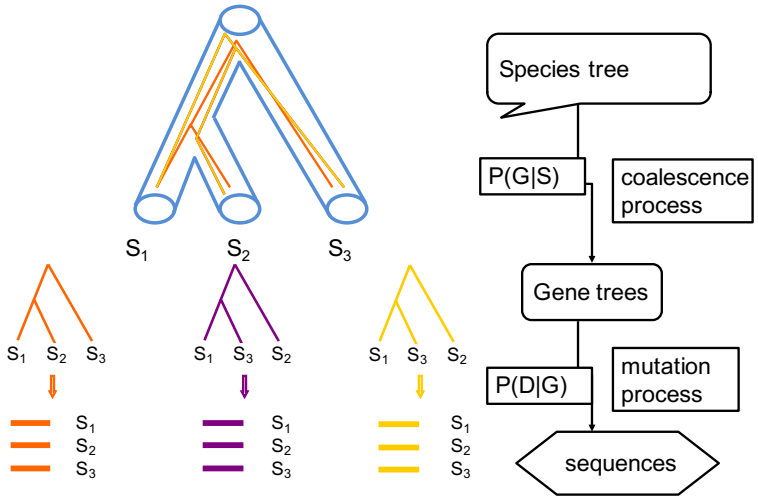

Figure 4.5: Hierarchical structure for gene tree and species tree estimation

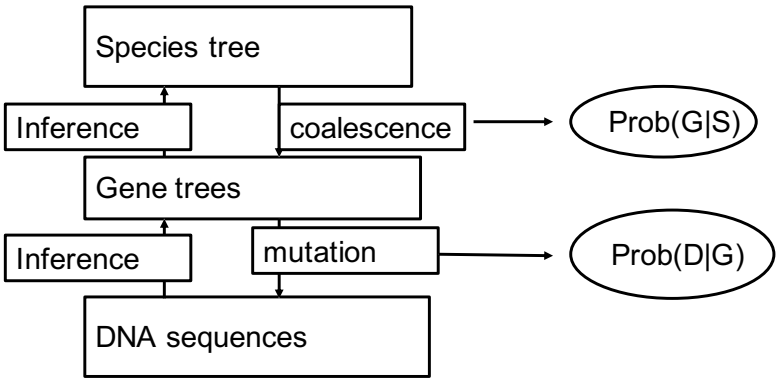

Figure 4.6: Bayesian inference for DNA sequences, gene trees, and species trees

4.2. Introduction

The phylogenomic analysis is emerging as a powerful tool for testing evolutionary hypotheses. The evolution of genetic markers, referring to a gene or DNA sequence with a known location on a chromosome that can be used to identify individuals or species, is a complex stochastic process. This process is primarily driven by two interacting forces - mutation and natural selection. Mutation randomly occurs in each genetic marker and provides novel materials on which selection may act. Meanwhile, individuals within and between populations evolve interactively, forming a complex evolutionary structure that involves vertical and horizontal evolution. Consequently, phylogenomic data are collections of heterogeneous genetic markers evolving interactively on a complex evolutionary structure. Since traditional phylogenetic models using a continuous time Markov chain for nucleotide substitutions are inadequate to reflect the underlying complex evolutionary structure, the past decade has witnessed a growing interest of developing unified evolutionary models by integrating the substitution process with the stochastic processes rooted in population genetics.

Madison, in their seminal paper (Maddison [1997]), has described several biological scenarios that can lead to conflicts between the genealogies of individual genetic markers and the phylogeny of species, forming a biological foundation for mathematical models unifying vertical and horizontal evolution. Particularly, random drift, which is modeled by a coalescence process, is the driving force of vertical evolution resulting in incongruent genealogies, and other biological factors, including horizontal gene transfer, hybridization, recombination, etc., are the

major contributors to horizontal evolution. Although the Madison paper focuses on the biological causes of incongruent gene trees, the scenarios described in this chapter have a revolutionary implication that genetic markers may evolve from multiple species trees. Figure 4.7 shows an example of hybridization may result in multiple species trees; the blue colored region in the first tree refers to a hybridization between species A and C, and there is 60% of B's genome are from species A and 40% from species B. Consequently, the tree structure contains loops and upgrade to a network. To tackle this problem, researchers have proposed many estimation methods to infer the network structure, such as Jin et al. [2006]; Solís-Lemus and Ané [2016], etc. Given the frequent vertical evolution and large parameter space, this approach suffers from computation and inconsistency problems in practice. We, however, argue that the single species tree is not sufficient to describe the evolution process, the complex network structure is not straightforward to properly estimate, and a mixture model with multiple species trees should be established. In Figure 4.7, the first phylogenetics network with one hybridization can be decomposed into two species trees, and each species tree has a certain probability, 60% and 40% in this case.

Gene flow may also result in multiple species trees, as showing in Figure 4.8. In Case 1, gene flow occurs between two sister species. Some genes evolve along the underlying species tree, while some genes split at a more recent time when gene flow occurs. Thus, gene trees are generated from two species trees with different branch lengths. In Case 2, when gene flow occurs, the corresponding genes evolve on a different species tree, so there are two underlying species trees.

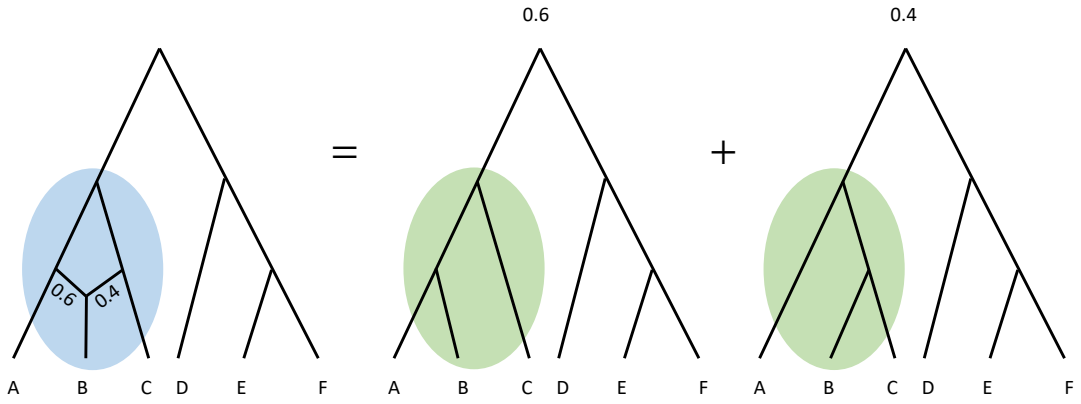

Figure 4.7: Hybridization result in multiple species trees.

Thus, horizontal gene transfer, like gene flow, may result in multiple species trees and different branch lengths.

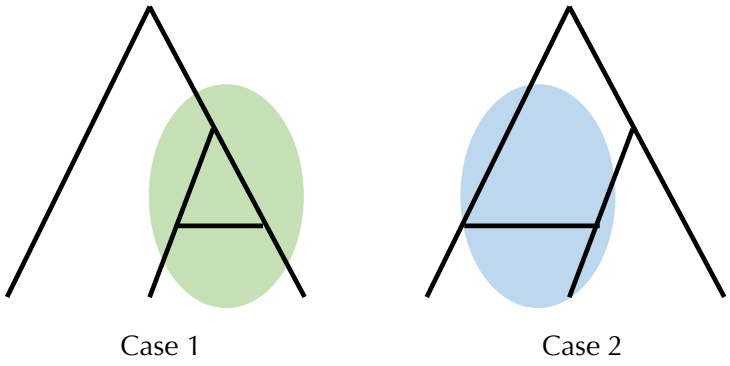

Figure 4.8: Gene flow result in multiple species trees.

Instead of estimating a single species tree or constructing a phylogenetic network, we propose a mixture model framework to estimate multiple species trees. We assume there are k different distributions for k species trees, where the gene trees are sampled. Given the multilocus data (including multiple loci, where a locus refers to a fixed position on a chromosome), we first construct gene trees with the state-of-the-art maximum likelihood-based algorithm, specifically, RAxML (Randomized Axelerated Maximum Likelihood) in Stamatakis [2014]. These gene trees are then clustered to different groups with a pre-specified number of clusters k. We then estimate the species trees with the gene trees from each of these clusters.

A long-standing and open problem in clustering analysis is how to choose the best k, the number of clusters. We further design a sequential testing framework to select the best k.

The rest of this chapter is organized as follows. Section 4.3 present the main algorithms to build multiple species trees and the sequential test to select the best number of clusters. We implemented extensive simulations to justify our methods in Section 4.4, while the findings with real-world data sets from the tree of life are discussed in Section 4.5. The discussion and future works in Section 4.6 close this chapter.

4.3. Methods and Algorithms

We first introduce two main algorithms, SCOT (Spectral Clustering on Trees) and STOCK (Sequential Test on Choosing k), where SCOT can build multiple species

trees given a set of gene trees, while STOCK can select the optimal number of species trees from gene trees.

4.3.1 SCOT: Spectral Clustering on Trees

The spectral based community detection algorithm can be naturally generalized to the current context. We build a network of gene trees by calculating the pairwise gene tree distances, and the distance matrix can be further converted to an adjacency matrix of a network. Figure 4.9 shows an illustrative example of gene trees in different clusters (we name them clouds). We omit the process of building gene trees from sequence data but focus on the procedure from gene trees to build multiple species trees.

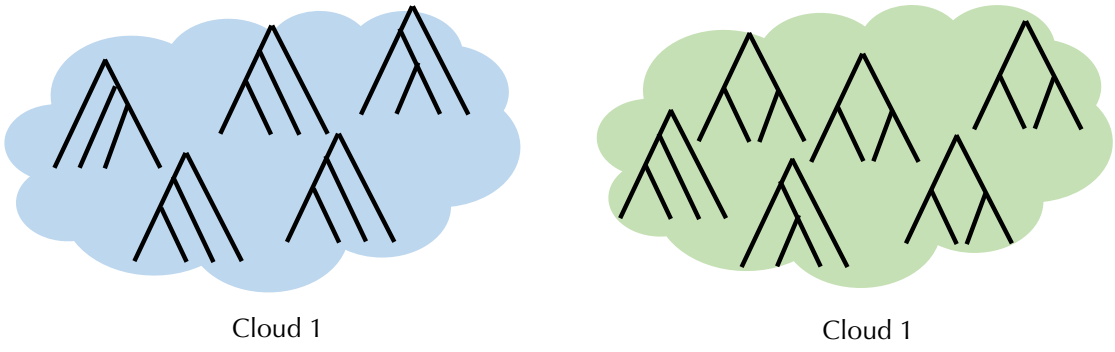

Figure 4.9: Illustrative diagram of gene trees in multiple phylogenetics clouds

Nonparametric distance metrics are widely used in phylogeny analysis, and we use the Robinson-Foulds distance (RF distance for short) in Robinson and Foulds [1981] to calculate the pairwise distance among gene trees. The RF distance is defined as follows. It removes each tree branch to get two seperated partitions,

the distance is defined as the summation of A and B. A refers to the number of partitions of data implied by the first tree but not the second tree, and B refers to the number of partitions of data implied by the second tree but not the first tree. Robinson and Foulds proved that the distance is a metric in Robinson and Foulds [1981].

Algorithm 4 presents the procedure to cluster the gene trees to k groups. After obtaining the similarity matrix S, we calculate the element-wise ratio matrix of the first k eigenvectors of S. A k-medoid algorithm is used to cluster the rows of ratio matrix to k groups. The key idea of taking element-wise ratio is inspired by Jin [2015], and it has been proved to remove all heterogeneity from the underline network - the network with all gene trees.

Algorithm 4 SCOT (Spectral Clustering on Trees)

procedure SCOT($\{gt_1, \ldots, gt_i, \ldots, gt_n\}$, k)
    Input: n gene trees $\{gt_1, \ldots, gt_i, \ldots, gt_n\}$ and the number of clusters k
    Calculate pairwise distance matrix $D_{n\times n}$
    Obtain similarity matrix $S_{n\times n} = \max(D) - D$, and set diagonals to 0.
    Perform Eigen Decomposition on $S_{n\times n} = U\Lambda U^{'}$, then obtain
        the first k leading eigenvector matrix $\hat{U}_{n\times k} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_k)$.
    Obtain the element-wise ratio matrix $\hat{R}_{n\times(k-1)}$ of $\hat{U}_{n\times k}$,
        where $\hat{R}_{n\times(k-1)} = (\frac{\hat{u}_2}{\hat{u}_1}, \frac{\hat{u}_3}{\hat{u}_1}, \ldots, \frac{\hat{u}_k}{\hat{u}_1})$.
    Apply k-medoid on $\hat{R}_{n\times(k-1)}$ and return $C = \{C_1, \ldots C_j, \ldots, C_k\}$,
        where C_j contains the gene trees in the jth cluster.
end procedure

4.3.2 STOCK: Sequential Test on Choosing k

We further propose the sequential algorithm, STOCK, in Algorithm 6 with helper algorithm in Algorithm 5. The helper algorithm presents the procedure to run a single test, where H_0 assumes l clusters while H_1 claims $l+1$ clusters. We first run SCOT algorithm to cluster the observed gene trees to l and $l+1$ groups respectively, denote as C^l and C^{l+1}. We then build species trees with NJst algorithm in Liu and Yu [2011a] for each cluster in C^l and C^{l+1}. The test statistic is defined as the ratio of sum mean distance (SMD), where the SMD is the distance of gene trees to the corresponding species trees in their cluster.

The goal of Algorithm 6 is to determine the optimal number of clusters based on the observed gene trees. We started from $k = 1$ versus $k = 2$ and sequentially move to next test if the preceding test is rejected until a test $k = l$ versus $k = l+1$ is not rejected. Thus, we choose l as the optimal value of the number of clusters.

Algorithm 5 Test (testing H_0: $k = l$ versus H_1: $k = l + 1$)

procedure TEST($\{gt_1, \ldots, gt_n\}$, k)
    Input: n gene trees $\{gt_1, \ldots, gt_n\}$, number of clusters k
    Run SCOT($\{gt_1, \ldots, gt_n\}, l$) to obtain $C^l = \{C_1, \ldots C_j, \ldots, C_l\}$
    Run NJst algorithm in Liu and Yu [2011b] with gene trees in each cluster C_j to construct l species trees, denote as $SP^l = \{SP_1, \ldots SP_l\}$.
    Run SCOT($\{gt_1, \ldots, gt_n\}, l + 1$) to obtain $C^{l+1} = \{C_1, \ldots C_j, \ldots, C_{l+1}\}$
    Run NJst algorithm in Liu and Yu [2011b] with gene trees in each cluster C_j to construct $l+1$ species trees, denote as $SP^{l+1} = \{SP_1, \ldots SP_{l+1}\}$.
    Obtain test statistics t

$$t = \frac{SMD(C^l, SP^l)}{SMD(C^{l+1}, SP^{l+1})}$$

$$SMD\left(C^l, SP^l\right) = \sum_{j=1}^{l} Dist(gt \in C_j, SP_j)$$

$$Dist\left(gt \in C_j, SP_j\right) = \frac{\sum_{i=1}^{n_j} RF\left(gt_i, SP_j\right)}{n_j}$$

    where SMD represents the Sum of Mean Distance, and n_j is the number of gene trees in jth cluster.
    Output: test statistic t, SP^l
end procedure

Algorithm 6 STOCK (Sequential Test on Choosing k)

procedure STOCK
  Input: $\{gt_1, \ldots, gt_n\}$, number of null samples M, p-value threshold p
  for l in 1 to n **do**:
    Obtain t and SP^l with Algorithm 5 with $\{gt_1, \ldots, gt_n\}$ and l
    Simulate null distribution and obtain M-length array of test statistics $T = (t_1, ..., t_m, ..., t_M)$ in the following loop.
    for m in 1 to M **do**
      For each species tree in $SP^l = \{SP_1, \ldots SP_j, \ldots, SP_l\}$, simulate n_j gene trees with coalescence model, denote as gt_{sample}.
      Apply SCOT to gt_{sample} with $k = l + 1$, and obtain

$$C^{l+1}_{sample} = \{C_{s,1}, \ldots C_{s,j}, \ldots, C_{s,l+1}\}$$

      Run NJst algorithm to each $C_{s,j}$ to construct $l + 1$ species trees

$$SP^{l+1}_{sample} = \{SP_{s,1}, \ldots SP_{s,l+1}\}$$

      Obtain test statistics $t_m = \frac{SMD(C^l, SP^l)}{SMD(C^{l+1}_{sample}, SP^{l+1}_{sample})}$
    end for
    Obtain the p-value as percentage of t_m which is greater than t.
  end for
  If p-value $< p$, stop and return $k = l$
  Else: return $k = n$
end procedure

4.4. Simulation Study

We implement extensive simulations to show the robustness and consistency of our methods. Simulation in Section 4.4.1 shows that SCOT algorithm can effectively and precisely cluster the gene trees, and reconstruct the true species trees, with high accuracy. Simulation in 4.4.3 illustrates that, without knowing the true number of clusters or species trees k, the STOCK algorithm can recover the true number. In both simulations, many factors including the distance between true species trees, topology, branch length, population size, sequence generation error, etc., have been considered.

4.4.1 Simulation 1 - SCOT can recover the underlying clusters

The goal of simulation study in this section is to justify the performance of SCOT algorithm in two ways, (1). how accurate the SCOT algorithm cluster the gene trees, and (2) how well the estimated species trees from each clusters compared with the true species trees. For the first goal, we use *Hamming Distance* to measure the clustering error given the true cluster assignment; for the second goal, we use *Minimum Dispersion* to measure how the estimated species trees deviate from the true species trees.

Criteria 1. Hamming Distance

Hamming distance is a metric to measure the clustering quality. In information theory, the Hamming distance between two clustering results, the true assignment, and the clustering assignment, is the number of positions at which the correspond-

ing assignments are different. In other words, it measures the minimum number of substitutions required to change one vector into the other or the minimum number of errors that could have transformed one into the other.

Criteria 2. Minimum Dispersion

Minimum Dispersion measures the deviation between estimated species trees and true species trees. Assume the k true species trees are $SP_1, \ldots, SP_k$, and the estimated k species trees are $\hat{SP}_1, \ldots, \hat{SP}_k$, the Minimum Dispersion is defined as the minimum pairwise RF distance between true and estimated species trees. For example, when k=2, the minimum dispersion is defined as

$$\min\{Dist(SP_1, \hat{SP}_1) + Dist(SP_2, \hat{SP}_2), Dist(SP_1, \hat{SP}_2) + Dist(SP_2, \hat{SP}_1)\}$$

The Minimum Dispersion implies that the ideal clustering results would have zero dispersion, while larger value delivers worse performance.

We consider two scenarios, (a). simulation without sequence generation, and (b). simulation with sequence generation. The scenario (b) is more challenge because of the present of gene tree estimation errors. For each scenario, we use two different population sizes, i.e., $\theta = 0.01$ and 0.05. Larger population size makes the estimation of species tree more difficult.

(a). Simulations without sequence generation. In this simulation study, we first define k true species trees, then simulate n gene trees from each species tree with coalescence model, so the total number of gene trees is nk. The SCOT algorithm with pre-defined number k then apply to the nk gene trees and calculate the corresponding Hamming Distance and Minimum Dispersion. To justify

SCOT's performance on the different number of true species trees, we implement simulations with true $k = 2, 3, 4$ with various of topologies.

1. $k = 2$. We select three different combinations of pair true species trees from the four topologies in Figure 4.10. Species Tree 1 is paired with each of the other three species trees, as the two true species trees.

2. $k = 3$. From Figure 4.10, we use Species Tree 1, 2, and 3 as the first group of true species trees, and Species Tree 1, 2, and 4 as the second group of true species trees.

3. $k = 4$. We choose 4 species trees in Figure 4.10 as the true species trees.

(b). Simulations with sequence generation. We extend simulation (a) to more complex scenarios by adding sequence generation. The estimation error from sequence to gene trees is not neglectable, so instead of applying SCOT directly to gene trees from the coalescence model, we first simulate sequence data from the gene trees with Seq-gen tool, then use RAxML to produce gene trees (RAxML_bestTree) from sequence data, the SCOT algorithm is then applied to the RAxML gene trees. The setting in the section can test the robustness of SCOT algorithm by tuning many factors including site information fraction, sequence length, etc. Similar to simulation (a), we still consider different number of true species trees and various of topologies, thus we simply use the species tree settings in simulation 1(a).

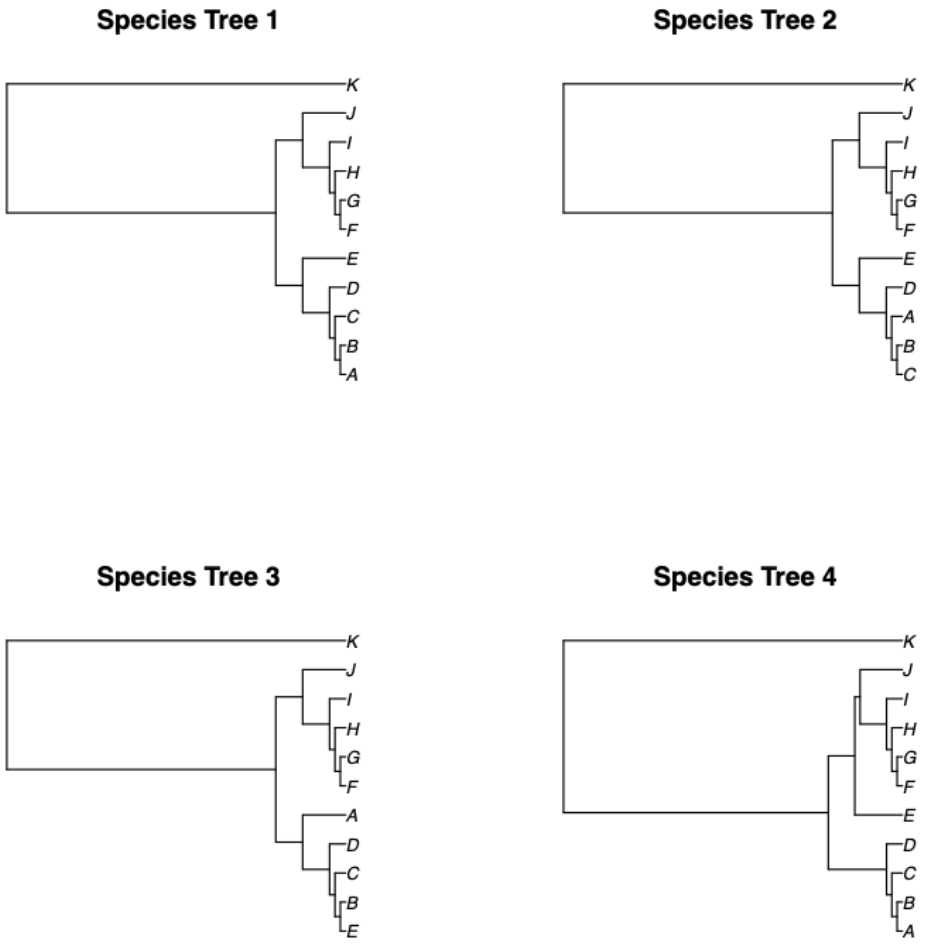

Figure 4.10: True species trees in simulation 1

4.4.2 Simulation Results

We organize the results in this section as follows, Figure 4.11 shows the hamming error and min dispersion plots with $k = 2$ under Simulation (a) and (b), Figure 4.12 displays same type of plots with $k = 3$, and Figure 4.13 reports the results with $k = 4$.

For all simulations, we observed that adding the sequence generation step can bring to much noise and influence the algorithm performance.

The results in Figure 4.11 are very inspiring. Group 1 is the most challenging case, where we change the order of tips A and C to formulate two true species trees. Even though the hamming error in four subgroups is not perfect, we can still recover the true species trees with very small dispersion. For both Group 2 and 3, the Dispersions are all zero, which means we can perfectly recover the true species trees.

The clustering analysis is more challenging with a larger number of true clusters. Figure 4.12 shows the results with $k = 3$. Both Hamming Error and Min Dispersion become worse than $k = 2$ case, while SCOT algorithm can still recover the true species tree topologies with low error and dispersion.

Figure 4.13 presents the results when $k = 4$. The SCOT algorithm still maintains satisfied performance with relatively low Hamming Errors and Min Dispersions in all cases.

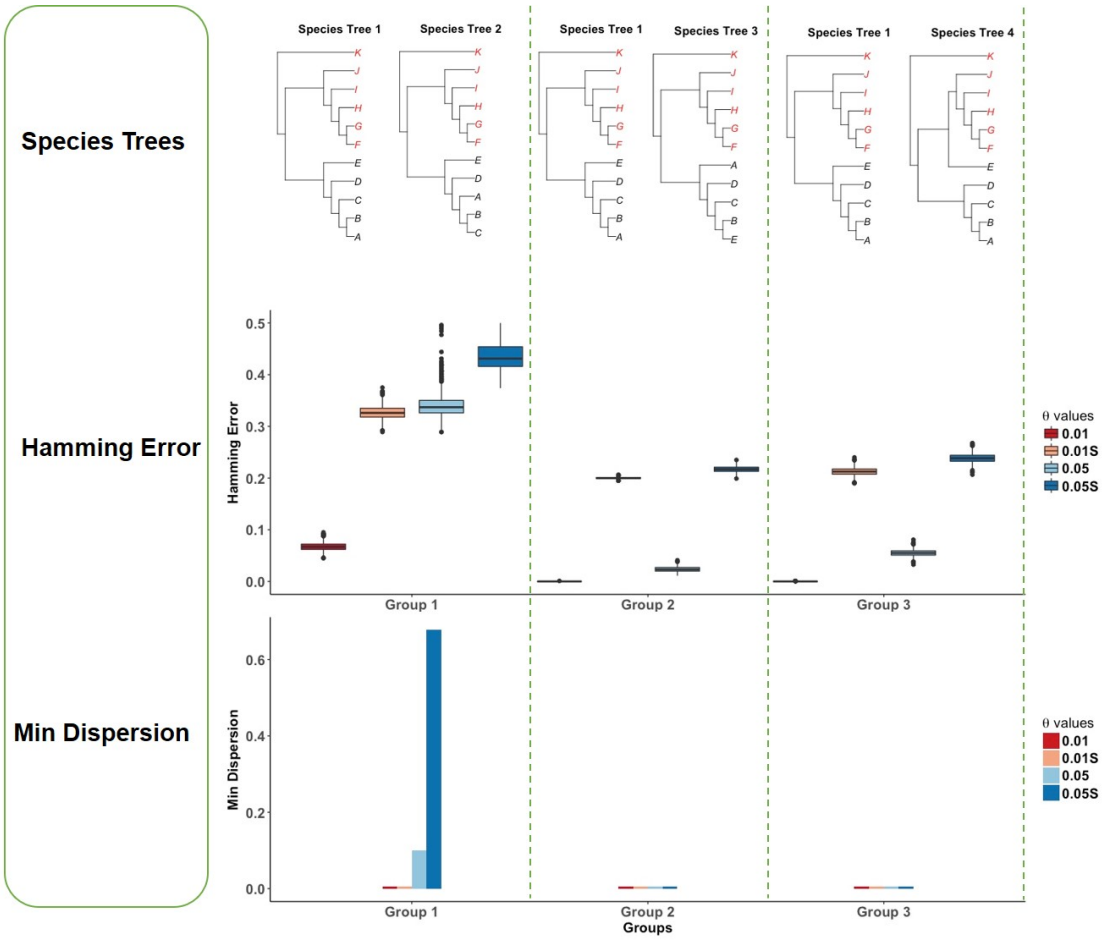

Figure 4.11: Simulation results with $k = 2$ in Simulation 1. We choose θ as 0.01 or 0.05, 0.01S means that we implemented sequence generation in the simulation, 0.05S similarly.

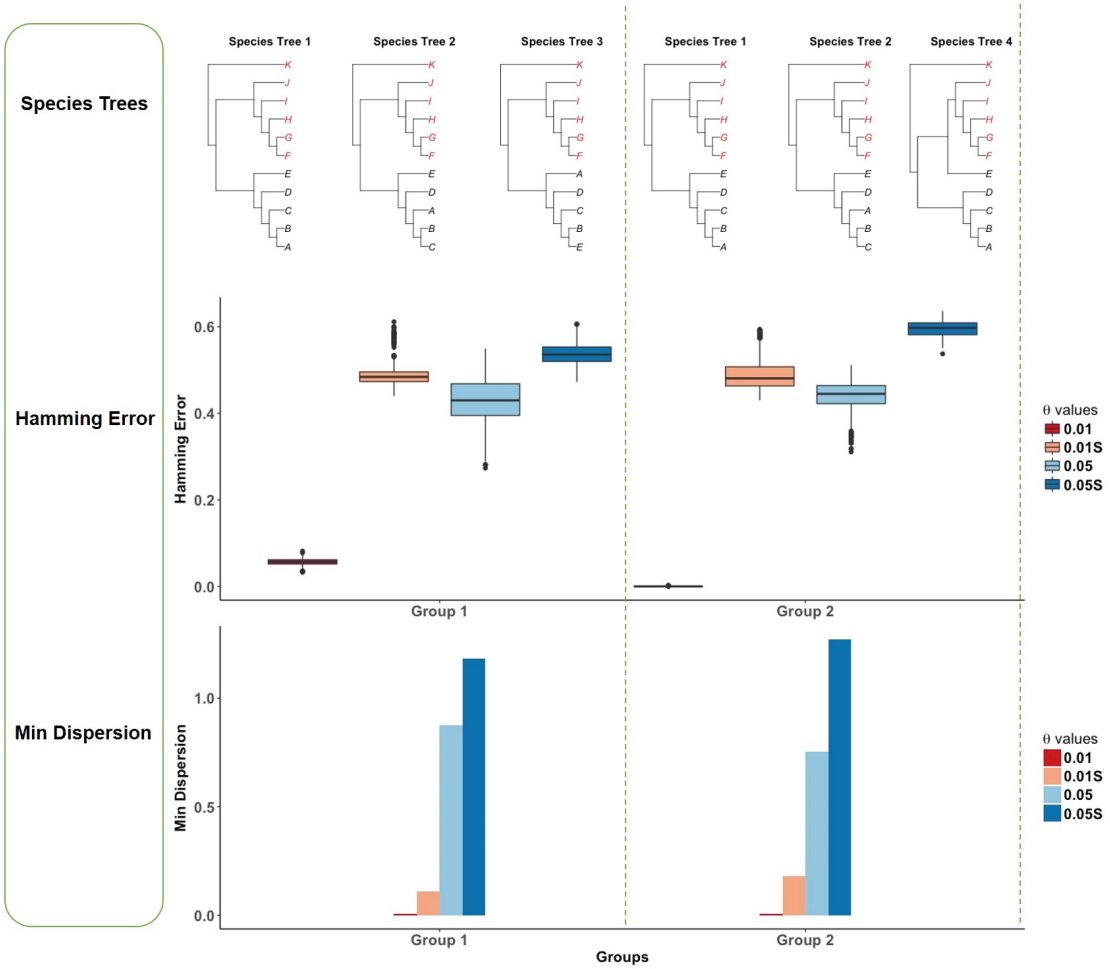

Figure 4.12: Simulation results with $k = 3$ in Simulation 1.

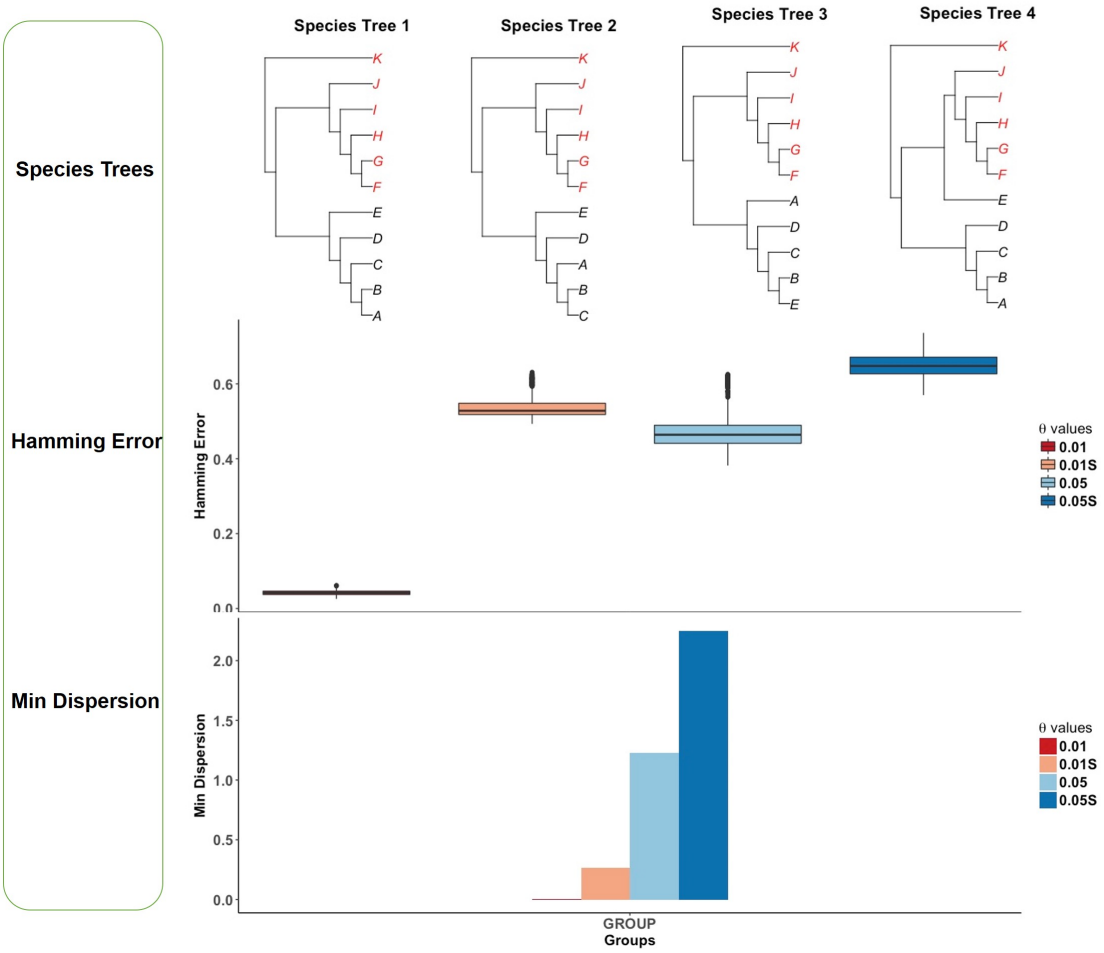

Figure 4.13: Simulation results with $k = 4$ in Simulation 1.

4.4.3 Simulation 2 - STOCK can recover the number of clusters

In this section, we implement a set of simulations to test the robustness to select the number of true species trees. We first define k true species trees, then simulate n gene trees from each species tree with coalescence model, so the total number of gene trees is nk. The STOCK algorithm will be implemented to all gene trees and select the best number of species trees with p-value threshold as 0.05. We replicate this procedure for 20 times and calculate the percentage that we select the true k. We implement these simulations for two scenarios, $k = 2$ and 3, and the results are listed in Table 4.1. For both two and three true species trees cases, without sequence generation, the STOCK algorithm can recover the true number of clusters when $\theta = 0.01$, while can only recover part of true numbers when $\theta = 0.05$. With sequence generation, the performance is also not satisfied for $k = 3$ scenario.

Even though the results are not impressive from Table 4.1 - in many cases we cannot recover the exact number of clusters, our test can recover the fact that the number of clusters is not one in most cases. 4.2 shows the percentage that STOCK algorithm get multiple clusters. In 75% cases, we can correctly recover that the number of clusters is not one, while in more challenging cases, such as the first row in 4.2, the algorithm produces a single species tree when $\theta = 0.05$.

K	True species trees	No Sequence generation		Sequence generation	
		$\theta = 0.01$	$\theta = 0.05$	$\theta = 0.01$	$\theta = 0.05$
2	Tree 1 and 2	1	0.25	1	0.65
	Tree 1 and 3	1	0.90	0.65	1
	Tree 1 and 4	1	1	0.85	1
3	Tree 1, 2 and 3	1	0	0.15	0.20
	Tree 1, 2 and 4	1	0	0.95	0.05

Table 4.1: The percentage of correctly recover the number of true clusters

K	True species trees	No Sequence generation		Sequence generation	
		$\theta = 0.01$	$\theta = 0.05$	$\theta = 0.01$	$\theta = 0.05$
2	Tree 1 and 2	1	0.25	1	0.65
	Tree 1 and 3	1	0.90	1	1
	Tree 1 and 4	1	1	1	1
3	Tree 1, 2 and 3	1	1	1	1
	Tree 1, 2 and 4	1	1	0.40	0.80

Table 4.2: The percentage of correctly recover the multiple clusters

4.5. Multiple Species Trees are Found in the Tree of Life

We further implement our algorithms to seven data sets in the tree of life and detect multiple species trees in three of them. A summary statistics table can be found in Table 4.3. We conclude that our algorithm detected multiple species trees in four data sets, while the other three data sets have single species trees. From a biological point of view, it's reasonable to detect single gene trees in species of bird, mammal, and vertebrate, because the gene flow and hybridization are rare compared to other data sets.

Data	Source	#Loci	#Species	Missing Species	Predicted k
Bird	Jarvis et al. [2014]	9,943	48	Yes	1
Fish	Cui et al. [2013]	1182	27	No	3
Mammal	Liu et al. [2017]	5,162	90	Yes	1
Metazoa	Whelan et al. [2015]	225	21	No	2
Vertebrate	Shen et al. [2017]	1,087	18	No	1
Yeast	Salichos and Rokas [2013]	1,070	23	Yes	3
Plant	Cai et al. [2019]	467	103	Yes	3

Table 4.3: Summary statistics of data sets in the tree of life

To visualize the multiple species trees, we used *Sumtree.py* algorithm in Sukumaran and Holder [2010] to show the evolution process in a single tree structure. Figure 4.14 presents the consensus tree for the Fish data set. Each internal branch has a bracket on it with the proportion of supports from each estimated species trees. In other words, if the bracket only contains a single number - it should be 1, meaning that all species trees support this local topology. If the bracket has more than one numbers, each number represents the corresponding support from certain species trees. We save the consensus trees for the other three data sets that have multiple species trees to Appendix.

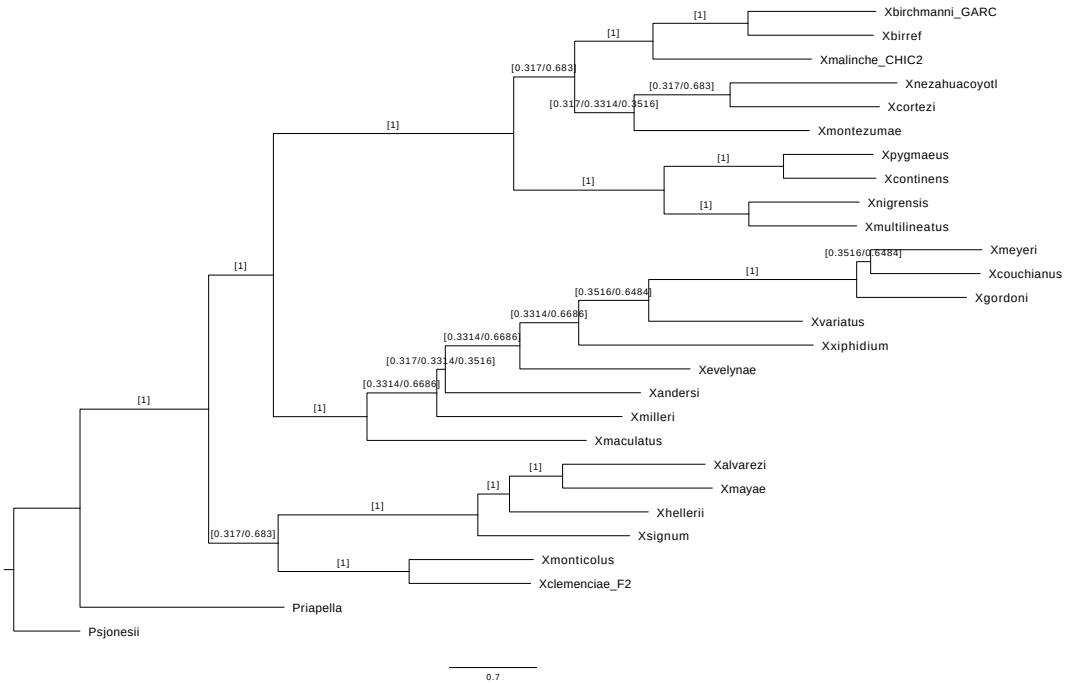

Figure 4.14: Consensus tree for multiple species trees in Fish data in Cui et al. [2013]

4.6. Discussion and Future Works

Species tree estimation is essential to understand the evolution process. The theory of a single species tree is not adequate and sufficient to explain the incongruent trees from many real-world data sets. Given the various vertical and horizontal evolution process, a more sophisticated representation method is necessary and desirable. We claimed the existence of multiple species trees and proposed a set of new algorithms to estimate such trees and present in a new consensus species tree. We implemented extensive simulations to verify our claim, which shows that the newly proposed methods can detect the existence of multiple species trees with high accuracy in different signal-to-noise levels. We also analyzed seven data sets from different species and families, and multiple species trees are detected in four data sets.

We are currently testing with more real-world data sets from a broader range of orders and families, with hope to further explore the existence of multiple species trees in diverse of species. In the meanwhile, a more efficient algorithm is desirable to extend our hypothesis to larger and more complex data sets.

Bibliography

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Lada A Adamic and Natalie 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.

Amini Arash A, Aiyou Chen, Peter J. Bickel, and Elizaveta Levina. Pseudo-likelihood methods for community detection in large sparse networks. *Ann. Statist.*, 41(4):2097–2122, 08 2013. doi: 10.1214/13-AOS1138. URL `https://doi.org/10.1214/13-AOS1138`.

Ery Arias-Castro. A short course on network analysis. *Methodological Advances in Statistics Related to Big Data*, 2015.

Debapratim Banerjee and Zongming Ma. Optimal hypothesis testing for stochastic block models with growing degrees, 2017.

Peter J. Bickel and Aiyou Chen. A nonparametric view of network models and

newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009a. doi: 10.1073/pnas.0907096106.

Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newmangirvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009b. doi: 10.1073/pnas.0907096106. URL `http://www.pnas.org/content/106/50/21068.abstract`.

N. Binkiewicz, J. T. Vogelstein, and K. Rohe. Covariate-assisted spectral clustering. *Biometrika*, 104(2):361–377, 2017. doi: 10.1093/biomet/asx008. URL `http://dx.doi.org/10.1093/biomet/asx008`.

Liming Cai, Zhenxiang Xi, Andr M. Amorim, M. Sugumaran, Joshua S. Rest, Liang Liu, and Charles C. Davis. Widespread ancient whole-genome duplications in malpighiales coincide with eocene global climatic upheaval. *New Phytologist*, 221(1):565–576, 2019. doi: 10.1111/nph.15357. URL `https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/nph.15357`.

Kehui Chen and Jing Lei. Network cross-validation for determining the number of communities in network data. *arXiv:1411.1715*, 2017.

Yudong Chen, Xiaodong Li, and Jiaming Xu. Convexified modularity maximization for degree-corrected stochastic block models. *arXiv:1512.08425v2*, 2015.

Zhengdao Chen, Lisha Li, and Joan Bruna. Supervised community detection with line graph neural networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=H1g0Z3A9Fm`.

Fan Chung and Mary Radcliffe. On the spectra of general random graphs. *the electronic journal of combinatorics*, 18(1):P215, 2011.

Rongfeng Cui, Molly Schumer, Karla Kruesi, Ronald Walter, Peter Andolfatto, and Gil G Rosenthal. Phylogenomics reveals extensive reticulate evolution in xiphophorus fishes. *Evolution*, 67(8):2166–2179, 2013.

Alexei J Drummond and Andrew Rambaut. Beast: Bayesian evolutionary analysis by sampling trees. *BMC evolutionary biology*, 7(1):214, 2007.

Scott V. Edwards, Liang Liu, and Dennis K. Pearl. High-resolution species trees without concatenation. *Proceedings of the National Academy of Sciences*, 104 (14):5936–5941, 2007. ISSN 0027-8424. doi: 10.1073/pnas.0607004104. URL `https://www.pnas.org/content/104/14/5936`.

J Felsenstein. Parsimony in systematics: Biological and statistical issues. *Annual Review of Ecology and Systematics*, 14(1):313–333, 1983. doi: 10.1146/annurev.es.14.110183.001525. URL `https://doi.org/10.1146/annurev.es.14.110183.001525`.

Chao Gao and John Lafferty. Testing for global network structure using small subgraph statistics, 2017.

Stephane Guindon, Franck Lethiec, Patrice Duroux, and Olivier Gascuel. Phyml onlinea web server for fast maximum likelihood-based phylogenetic inference. *Nucleic acids research*, 33(suppl_2):W557–W559, 2005.

Thomas Hofmann, Bernhard Schlkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008. doi: 10.1214/009053607000000677. URL http://dx.doi.org/10.1214/009053607000000677.

Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(83)90021-7. URL http://www.sciencedirect.com/science/article/pii/0378873383900217.

Erich D Jarvis, Siavash Mirarab, Andre J Aberer, Bo Li, Peter Houde, Cai Li, Simon YW Ho, Brant C Faircloth, Benoit Nabholz, Jason T Howard, et al. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.

Pengsheng Ji and Jiashun Jin. Coauthorship and citation networks for statisticians. *Annals of Applied Statistics*, 2016.

Guohua Jin, Luay Nakhleh, Sagi Snir, and Tamir Tuller. Maximum likelihood of phylogenetic networks. *Bioinformatics*, 22(21):2604–2611, 2006.

Jiashun Jin. Fast community detection by SCORE. *The Annals of Statistics*, 43 (1):57–89, 2015. ISSN 0090-5364. doi: 10.1214/14-AOS1265.

Jiashun Jin, Zheng Ke, and Shengming Luo. Network global testing by counting graphlets. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings*

of Machine Learning Research, pages 2333–2341, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/jin18b.html.

Antony Joseph and Bin Yu. Impact of regularization on spectral clustering. *Annals of Statistics*, 44(4):1765–1791, 08 2016. doi: 10.1214/16-AOS1447. URL http://dx.doi.org/10.1214/16-AOS1447.

Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E. Statistical, Nonlinear, and Soft Matter Physics*, 83(1):016107, 2011. ISSN 1539-3755. doi: 10.1103/PhysRevE.83.016107.

Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. AcM Press New York, 1993.

Eric D. Kolaczyk. *Statistical analysis of network data*. Springer Series in Statistics. Springer, New York, 2009. ISBN 978-0-387-88145-4. doi: 10.1007/978-0-387-88146-1. Methods and models.

Laura Salter Kubatko and James H Degnan. Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Systematic Biology*, 56(1):17–24, 2007.

Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, and Santo Fortunato. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011.

Rocco Langone, Raghvendra Mall, Carlos Alzate, and Johan A. K. Suykens. Kernel spectral clustering and applications. *arXiv:1505.00477*, 2015.

Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. *Annals of Statistics*, 43(1):215–237, 2015. ISSN 0090-5364. doi: 10.1214/14-AOS1274.

Wen-Hsiung Li. Simple method for constructing phylogenetic trees from distance matrices. *Proceedings of the National Academy of Sciences*, 78(2):1085–1089, 1981.

Liang Liu and Dennis K Pearl. Species Trees from Gene Trees: Reconstructing Bayesian Posterior Distributions of a Species Phylogeny Using Estimated Gene Tree Distributions. *Systematic Biology*, 56(3):504–514, 06 2007. ISSN 1063-5157. doi: 10.1080/10635150701429982. URL `https://dx.doi.org/10.1080/10635150701429982`.

Liang Liu and Lili Yu. Estimating Species Trees from Unrooted Gene Trees. *Systematic Biology*, 60(5):661–667, 03 2011a. ISSN 1063-5157. doi: 10.1093/sysbio/syr027. URL `https://dx.doi.org/10.1093/sysbio/syr027`.

Liang Liu and Lili Yu. Estimating Species Trees from Unrooted Gene Trees. *Systematic Biology*, 60(5):661–667, 03 2011b. ISSN 1063-5157. doi: 10.1093/sysbio/syr027. URL `https://dx.doi.org/10.1093/sysbio/syr027`.

Liang Liu, Lili Yu, Dennis K. Pearl, and Scott V Edwards. Estimating species

phylogenies using coalescence times among sequences. *Systematic biology*, 58 5: 468–77, 2009.

Liang Liu, Jin Zhang, Frank E Rheindt, Fumin Lei, Yanhua Qu, Yu Wang, Yu Zhang, Corwin Sullivan, Wenhui Nie, Jinhuan Wang, et al. Genomic evidence reveals a radiation of placental mammals uninterrupted by the kpg boundary. *Proceedings of the National Academy of Sciences*, 114(35):E7282–E7290, 2017.

David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

Wayne P Maddison. Gene trees in species trees. *Systematic biology*, 46(3):523–536, 1997.

Tamás Nepusz, Andrea Petróczi, László Négyessy, and Fülöp Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1):016107, 2008.

M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E. Statistical, Nonlinear, and Soft Matter Physics*, 74(3):036104, 2006a. ISSN 1539-3755. doi: 10.1103/PhysRevE.74.036104.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006b. doi: 10.1073/pnas.0601602103.

P Pamilo and M Nei. Relationships between gene trees and species trees. *Molecular Biology and Evolution*, 5(5):568–583, 09 1988. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040517. URL https://dx.doi.org/10.1093/oxfordjournals.molbev.a040517.

Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3120–3128. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5099-regularized-spectral-clustering-under-the-degree-corrected-stochastic-block pdf.

Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658–2663, 2004.

D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1):131 – 147, 1981. ISSN 0025-5564. doi: https://doi.org/10.1016/0025-5564(81)90043-2. URL http://www.sciencedirect.com/science/article/pii/0025556481900432.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-

dimensional stochastic blockmodel. *Annals of Statistics*, 39(4):1878–1915, 08 2011. doi: 10.1214/11-AOS887. URL `https://doi.org/10.1214/11-AOS887`.

Leonidas Salichos and Antonis Rokas. Inferring ancient divergences requires genes with strong phylogenetic signals. *Nature*, 497(7449):327, 2013.

Purnamrita Sarkar and Peter J. Bickel. Role of normalization in spectral clustering for stochastic blockmodels. *Annals of Statistics*, 2015. doi: 10.1214/14-AOS1285.

Xing-Xing Shen, Chris Todd Hittinger, and Antonis Rokas. Contentious relationships in phylogenomic studies can be driven by a handful of genes. *Nature Ecology & Evolution*, 1(5):0126, 2017.

Claudia Solís-Lemus and Cécile Ané. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS genetics*, 12(3): e1005896, 2016.

Alexandros Stamatakis. Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.

Jeet Sukumaran and Mark T Holder. Dendropy: a python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571, 2010.

Christian Von Mering, Roland Krause, Berend Snel, Michael Cornell, Stephen G Oliver, Stanley Fields, and Peer Bork. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, 417(6887):399, 2002.

Y. X. Rachel Wang and Peter J. Bickel. Likelihood-based model selection for stochastic block models. *Ann. Statist.*, 45(2):500–528, 04 2017. doi: 10.1214/16-AOS1457. URL https://doi.org/10.1214/16-AOS1457.

Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440, 1998.

Nathan V Whelan, Kevin M Kocot, Leonid L Moroz, and Kenneth M Halanych. Error, signal, and the placement of ctenophora sister to all other animals. *Proceedings of the National Academy of Sciences*, 112(18):5773–5778, 2015.

Wikipedia. Phylogenetic tree — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Phylogenetic%20tree&oldid=882389445, 2019. [Online; accessed 08-March-2019].

Bowei Yan and Purnamrita Sarkar. On robustness of kernel clustering. In *Advances in Neural Information Processing Systems*, pages 3090–3098, 2016.

Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156. IEEE, 2013. doi: 10.1109/icdm.2013.167.

Hao Yin, Austin R. Benson, and Jure Leskovec. Higher-order clustering in networks. *Phys. Rev. E*, 97:052306, May 2018. doi: 10.1103/PhysRevE.97.052306. URL https://link.aps.org/doi/10.1103/PhysRevE.97.052306.

Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the daviskahan theo-

rem for statisticians. *Biometrika*, 102(2):315–323, 2015. doi: 10.1093/biomet/asv008. URL +http://dx.doi.org/10.1093/biomet/asv008.

Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012a. ISSN 0090-5364. doi: 10.1214/12-AOS1036.

Yunpeng Zhao, Elizaveta Levina, and Ji Zhu. Consistency of community detection in networks under degree-corrected stochastic block models. *Annals of Statistics*, 40(4):2266–2292, 08 2012b. doi: 10.1214/12-AOS1036. URL https://doi.org/10.1214/12-AOS1036.

Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. *Proc. VLDB Endow.*, 2(1):718–729, August 2009. ISSN 2150-8097. doi: 10.14778/1687627.1687709. URL http://dx.doi.org/10.14778/1687627.1687709.

4.7. Appendix

4.7.1 Proofs for main results under NSBM

Lemma 6 *(Perfect clustering under NSBM) Suppose P and P' is non-singular, non-negative, symmetric and irreducible and all eigenvalues of $D((1-\alpha)P+\alpha P')D$ are simple. Let $D((1-\alpha)P+\alpha P')D = U\Lambda U'$ be its eigendecomposition where Λ is a diagonal matrix and $U'U = UU' = I_R$. Perfect clustering with Ω and $\mathcal{X}$ under NSBM is achieved with CSC algorithm.*

Then

(1) For $\mathcal{W} = (1-\alpha)\Omega + \alpha K = (1-\alpha)MPM' + \alpha MP'M$, we have the decomposition

$$(1-\alpha)MPM' + \alpha MP'M = (MD^{-1}U)\Lambda(MD^{-1}U)',$$

and the R nonzero eigenvalues are the diagonal entries of Λ and the corresponding eigenvectors are the columns of $MD^{-1}U$.

(2) the $n \times R$ matrix $MD^{-1}U$ has R distinct rows, each of which corresponds to a community specified in M.

Proof 1 *Without loss of generality, we start from a K-community undirected and*

unweighted network $G = (E, V)$, *let* $\mathcal{W} = (1-\alpha)\Omega + \alpha K$,

$$\begin{aligned}(1-\alpha)\Omega + \alpha K &= (1-\alpha)MPM + \alpha MP'M \\ &= M((1-\alpha)P + \alpha P')M \\ &= MD^{-1}U\Lambda U'D^{-1}M \\ &= (MD^{-1}U)\Lambda(MD^{-1}U)\end{aligned} \tag{4.1}$$

Proposition 1 *(Concentration bound of binary symmetric random matrices) Let* A *be the adjacency matrix of a random graph on* n *nodes in which edges occur independently. Set* $E[A] = P = (p_{ij})_{i,j=1,\dots,n}$, *and assume that* $n \max_{ij} p_{ij} \leq d$ *for* $d \geq c_0 \log n$ *and* $c_0 > 0$. *Then, for any* $r > 0$ *there exists a constant* $C = C(r, c_0)$ *such that*

$$||A - P||_2 \leq C\sqrt{d}$$

with probability at least $1 - n^{-r}$.

Proof 2 *Please refer to the proof of Theorem 5.2 in Lei and Rinaldo [2015], and Lemma 2 in Arias-Castro [2015].*

Theorem 9 *(Concentration bound on connectivity matrix under* $nSBM$*) Let* $W = (1-\alpha)A + \alpha K$, *then* $||W - E[W]||_\infty \leq (1-\alpha)C\sqrt{n}\sqrt{d} + \alpha c\sqrt{\frac{\log p}{p}}$ *with probability at least* $1 - \max\{n^{-r}, n^2 p^{-\rho c^2}\}$

Proof 3 *Let* $W = (1-\alpha)A + \alpha K(X, X)$, *where* A *is the adjacent matrix,* K *is*

Kernel matrix, and X is the node attributes, then

$$\begin{aligned} ||W - E[W]||_\infty &= ||(1-\alpha)A + \alpha K(X,X) - E[(1-\alpha)A + \alpha K(X,X)]||_\infty \\ &\leq (1-\alpha)||A - E[A]||_\infty + \alpha||K(X,X) - E[K(X,X)]||_\infty \\ &\leq (1-\alpha)\sqrt{n}||A - E[A]||_2 + \alpha||K(X,X) - E[K(X,X)]||_\infty \\ &= T_1 + T_2 \end{aligned} \tag{4.2}$$

For T_1, given the results in Lemma 1, we arrive at

$$\begin{aligned} T_1 &= (1-\alpha)\sqrt{n}||A - E[A]||_2 \\ &\leq (1-\alpha)C\sqrt{n}\sqrt{d} \end{aligned} \tag{4.3}$$

For T_2, Yan and Sarkar [2016] provides a tight bound for the empirical Kernel matrix and its population counterpart, given X distributed in mixture of gaussian, which leads us to

$$\begin{aligned} T_2 &= \alpha||Kr(X,X) - E[Kr(X,X)]||_\infty \\ &\leq \alpha c\sqrt{\frac{\log p}{p}} \end{aligned} \tag{4.4}$$

with probability at least $1 - n^2 p^{-\rho c^2}$, where p is the dimension of X, c and $\rho > 0$ are constants.

Then we combine T_1 and T_2 with triangle inequality to have the follows

$$\begin{aligned} ||W - E[W]||_\infty &\leq T_1 + T_2 \\ &\leq (1-\alpha)C\sqrt{n}\sqrt{d} + \alpha c\sqrt{\frac{\log p}{p}} \end{aligned} \tag{4.5}$$

with probability at least $1 - \max\{n^{-r}, n^2 p^{-\rho c^2}\}$

Lemma 7 *(Principal subspace perturbation bound) Let W and $E(W)$ have eigenvalues $\hat{\lambda}_1, ..., \hat{\lambda}_n$ and $\lambda_1, ..., \lambda_n$. Let the first R leading eigenvectors corresponding the largest R leading eigenvalues being U and $\hat{U}$ for $E[W]$ and W respectively, then there exists an orthogonal matrix $\hat{O}$, such that,*

$$||\hat{U}\hat{O} - U||_F \leq \frac{2^{3/2}\sqrt{nr}\max\{C\sqrt{n}\sqrt{d}, c\sqrt{\frac{\log p}{p}}\}}{\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}}$$

Proof 4 *Simply apply Davis-Khan Theorem.*

$$\begin{aligned} ||\hat{U}\hat{O} - U||_F &\leq \frac{2^{3/2}\sqrt{nr}||W - E[W]||_\infty}{\min\{-\lambda_1, \lambda_R - \lambda_{R+1}\}} \\ &= \frac{2^{3/2}\sqrt{nr}\max\{C\sqrt{n}\sqrt{d}, c\sqrt{\frac{\log p}{p}}\}}{\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}} \end{aligned} \tag{4.6}$$

Theorem 10 *(Error bound of k-means on leading eigenvectors) Under NSBM with Gaussian distributions in $\mathcal{F}$, the error bound of k-means on the first R leading eigenvectors is*

$$||Z||/N \leq \frac{64mnr \max\{C\sqrt{n}\sqrt{d}, c\sqrt{\frac{\log p}{p}}\}^2}{N \min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}^2}$$

where $m = \max (M^T M)_{ii}$.

Proof 5 *Binkiewicz et al. [2017] established a sharp bound on erros of k-means algorithm with principal subspace perturbation bound (as in Lemma 7), then we have the following inequalitiy holds*

$$||Z||/N \leq \frac{8m}{N}||\hat{U}\hat{O} - U||_F^2$$

With the results in Lemma 7, we arrive at the following mis-clustering bound with probability $1 - \alpha$

$$||Z||/N \leq \frac{64mnr \max\{C\sqrt{n}\sqrt{d}, c\sqrt{\frac{\log p}{p}}\}^2}{N \min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}^2}$$

4.7.2 Proofs for main results under NDCBM

The theoretical analysis for CSC under NDCBM is more challenging than NSBM, due to the limitation of available theories in the random matrix. To derivate the perfect clustering property in this scenario, we employ the cutting-edge matrix perturbation techniques.

Lemma 8 *(Perfect clustering with CSC-Row-Normalization under NDCBM). CSC algorithm achieve perfect clustering under NDCBM settings.*

Proof 6 *We apply the matrix perturbation techniques to prove the following statements.*

(1). Perfect clustering when $\alpha\mathcal{K}$ is a perturbation to $(1-\alpha)\Omega$.

Let $\mathcal{W} = (1-\alpha)\Omega + \alpha\mathcal{K} = \mathcal{W}_1 + \mathcal{W}_2$, and $(1-\alpha)\Omega = V\Lambda V'$. We anticipate the eigenvalue of $\mathcal{W}$ changes from λ_i to $\lambda_i + \hat{\lambda}_i$, the eigenvector changes from v_i to $v_i + \hat{v}_i$, then for the eigenvectors corresponding to the non-zero eigenvalues, we have,

$$(\mathcal{W}_1 + \mathcal{W}_2)(v_i + \hat{v}_i) = (\lambda_i + \hat{\lambda}_i)(v_i + \hat{v}_i)$$

Given the fact that $\mathcal{W}_1 v_i = \lambda_i v_i$, and remove the second order terms, we have

$$\mathcal{W}_2 v_i + \mathcal{W}_1 \hat{v}_i = \lambda_i \hat{v}_i + \hat{\lambda}_i v_i$$

We also have

$$\hat{v}_i = \sum_{j=1}^{n} \omega_j v_j$$

since $\mathcal{W}_1$ is a real and symmetric matrix, where ω_j is the coefficient and v_j is the jth eigenvector of $\mathcal{W}_1$. Insert this equation into Eq.(*), we arrive at

$$\mathcal{W}_2 v_i + \sum_{j=1}^{n} \omega_j \lambda_j v_j = \lambda_i \sum_{j=1}^{n} \omega_j v_j + \hat{\lambda}_i v_i$$

We then multiply v_i' to both sides results to

$$v_i' \mathcal{W}_2 v_i + \sum_{j=1}^{n} \omega_j \lambda_j v_i' v_j = \lambda_i \sum_{j=1}^{n} v_i' \omega_j v_j + \hat{\lambda}_i v_i' v_i$$

Notice that in the last equation equals to 1 and the second term in the left-hand side and the first term in the right-hand side cancel each other. Thus we have

$$\hat{\lambda}_i = v_i' \mathcal{W}_2 v_i$$

With the similar technique, multiply v_j' to both sides of Eq. (*), resulting

$$v_j' \mathcal{W}_2 v_i + \sum_{k=1}^{n} \omega_k \lambda_k v_j' v_k = \lambda_i \sum_{k=1}^{n} v_j' \omega_k v_k$$

which evaluates to

$$v_j' \mathcal{W}_2 v_i + \omega_j \lambda_j = \lambda_i \omega_j$$

where we can find that

$$\omega_j = \frac{v_j' \mathcal{W}_2 v_i}{\lambda_i - \lambda_j}$$

then we have

$$\hat{v}_i = \sum_{j \neq i} \frac{v_j' \mathcal{W}_2 v_i}{\lambda_i - \lambda_j} v_j$$

When $R = 2$, i.e., a network with two communities, $i, j = 1, 2$, then we have

$$\hat{v}_1 = \frac{v_2' \mathcal{W}_2 v_1}{\lambda_1 - \lambda_2} v_2$$

and

$$\hat{v}_2 = \frac{v_1' \mathcal{W}_2 v_2}{\lambda_2 - \lambda_1} v_1$$

And we have

$$\begin{aligned} v_1 + \hat{v}_1 &= v_1 + \frac{v_2' \mathcal{W}_1 v_1}{\lambda_1 - \lambda_2} v_2 \\ &= v_1 + m_2 \sum_{i=1}^{n} \theta_i v_2 \\ &= v_1 + m_2 c v_2 \end{aligned} \tag{4.1}$$

And

$$\begin{aligned} v_2 + \hat{v}_2 &= v_2 + \frac{v_1' \mathcal{W}_1 v_2}{\lambda_2 - \lambda_1} v_1 \\ &= v_2 + m_1 \sum_{i=1}^{n} \theta_i v_1 \\ &= v_2 + m_1 c v_1 \end{aligned} \tag{4.2}$$

where $c = \sum_1^n \theta_i$

Let's assume

$$v_1 = (\theta_1 x_1, \theta_2 x_2, ..., \theta_n x_n)'$$

and

$$v_2 = (\theta_1 y_1, \theta_2 y_2, ..., \theta_n y_n)'$$

Then for $i = 1, ..., n$, *we have*

$$U_{i1}^* = \frac{x_i + m_1 c y_i}{\sqrt{(x_i + m_1 c y_i)^2 + (y_i + m_2 c x_i)^2}} \tag{4.3}$$

which is free of θ_i.

(2). Perfect clustering when $(1-\alpha)\Omega$ *is a perturbation to* $\alpha\mathcal{K}$.

With the similar techniques as in (1), let $\mathcal{W} = (1-\alpha)\Omega + \alpha\mathcal{K} = \mathcal{W}_1 + \mathcal{W}_2$, *and* $\mathcal{W}_2 = U\Lambda U'$. *We anticipate the eigenvalue of* $\mathcal{W}$ *changes from* λ_i *to* $\lambda_i + \hat{\lambda}_i$, *the eigenvector changes from* u_i *to* $u_i + \hat{u}_i$, *then for the eigenvectors corresponding to the non-zero eigenvalues.*

Then the perturbated eigenvector $\hat{u}_i$ *is*

$$\hat{u}_i = \sum_{j \neq i} \frac{u_j' \mathcal{W}_1 u_i}{\lambda_i - \lambda_j} u_j$$

When $R = 2$, *i.e., a network with two communities,* $i, j = 1, 2$, *then we have*

$$\hat{u}_1 = \frac{u_2' \mathcal{W}_1 u_1}{\lambda_1 - \lambda_2} u_2$$

and

$$\hat{u}_2 = \frac{u_1' \mathcal{W}_1 u_2}{\lambda_2 - \lambda_1} u_1$$

And we have

$$\begin{aligned} u_1 + \hat{u}_1 &= u_1 + \frac{u_2' \mathcal{W}_1 u_1}{\lambda_1 - \lambda_2} u_2 \\ &= u_1 + m_2 \sum_{i=1}^{n} \theta_i u_2 \\ &= u_1 + m_2 c u_2 \end{aligned} \tag{4.4}$$

And

$$\begin{aligned} u_2 + \hat{u}_2 &= u_2 + \frac{u_1' \mathcal{W}_1 u_2}{\lambda_2 - \lambda_1} u_1 \\ &= u_2 + m_1 \sum_{i=1}^{n} \theta_i u_1 \\ &= u_2 + m_1 c u_1 \end{aligned} \tag{4.5}$$

where $c = \sum_1^n \theta_i$

Let's assume

$$u_1 = (\theta_1 x_1, \theta_2 x_2, ..., \theta_n x_n)'$$

and

$$v_2 = (\theta_1 y_1, \theta_2 y_2, ..., \theta_n y_n)'$$

Then for $i = 1, ..., n$, *we have*

$$U_{i1}^* = \frac{x_i + m_1 c y_i}{\sqrt{(x_i + m_1 c y_i)^2 + (y_i + m_2 c x_i)^2}} \tag{4.6}$$

is free of θ_i.

Proposition 2 *(Spectral bound on adjacent matrix of general random graph) Let* G *be a random graph with* A *being a binary adjacent matrix, and* $E(A)_{ij} = p_{ij}$. *Then*

$$||A - E(A)||_2 \leq \sqrt{4 \max_{i=1,...,n} p_{ij} \log(2n/\epsilon)}$$

with probability at least $1 - \epsilon$.

Proof 7 *Please refer to Theorem 1 in Chung and Radcliffe [2011]*

Lemma 9 *(Concentration bound on W) Let* $W = (1-\alpha)A + \alpha K$*, then* $||W - E[W]||_\infty \leq (1-\alpha)\sqrt{4\max_{i=1,\ldots,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)} + \alpha c\sqrt{\frac{\log p}{p}}$ *with probability at least* $1 - \max\{1-\epsilon, n^2 p^{-\rho c^2}\}$

Proof 8 *Let* $W = (1-\alpha)A + \alpha K(X,X)$*, where* A *is the adjacent matrix,* K *is Kernel matrix, and* X *is the node attributes, then*

$$\begin{aligned} ||W - E[W]||_\infty &= ||(1-\alpha)A + \alpha K(X,X) - E[(1-\alpha)A + \alpha K(X,X)]||_\infty \\ &\leq (1-\alpha)||A - E[A]||_\infty + \alpha||K(X,X) - E[K(X,X)]||_\infty \\ &\leq (1-\alpha)\sqrt{n}||A - E[A]||_2 + \alpha||K(X,X) - E[K(X,X)]||_\infty \\ &= T_1 + T_2 \end{aligned} \tag{4.7}$$

For T_1*, given the results in Lemma 2, we arrive at*

$$\begin{aligned} T_1 &= (1-\alpha)\sqrt{n}||A - E[A]||_2 \\ &\leq (1-\alpha)\sqrt{4\max_{i=1,\ldots,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)} \end{aligned} \tag{4.8}$$

T_2 *has the same probabilistic bound as 1. Thus we arrive at*

$$\begin{aligned} ||W - E[W]||_\infty &\leq T_1 + T_2 \\ &\leq (1-\alpha)\sqrt{4\max_{i=1,\ldots,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)} + \alpha c\sqrt{\frac{\log p}{p}} \end{aligned} \tag{4.9}$$

with probability at least $1 - \max\{1 - \epsilon, n^2 p^{-\rho c^2}\}$

Theorem 11 *(Principal subspace perturbation bound) Under NDCBM settings, let* W *and* $E(W)$ *have eigenvalues* $\hat{\lambda}_1, ..., \hat{\lambda}_n$ *and* $\lambda_1, ..., \lambda_n$*. Let the first* R *leading eigenvectors corresponding the largest* R *leading eigenvalues being* U *and* $\hat{U}$ *for* $E[W]$ *and* W *respectively, then there exists an orthogonal matrix* $\hat{O}$*, such that,*

$$||\hat{U}\hat{O} - U||_F \leq \frac{2^{3/2}\sqrt{nr}\max\{\sqrt{4\max_{i=1,...,n}\theta_i\theta_j P_{g_i g_j}\log(2n/\epsilon)}, c\sqrt{\frac{\log p}{p}}\}}{\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}}$$

Let U^* *and* $\hat{U}^*$ *be the row-normalized version of* U *and* $\hat{U}$*, then we have*

$$||\hat{U}^*\hat{O} - U^*||_F \leq \frac{2^{3/2}\sqrt{nr}\max\{\sqrt{4\max_{i=1,...,n}\theta_i\theta_j P_{g_i g_j}\log(2n/\epsilon)}, c\sqrt{\frac{\log p}{p}}\}}{\Delta\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}}$$

where $\Delta = \min_i\{\min\{||U_{i\cdot}||_2, ||\hat{U}_{i\cdot}||_2\}\}$ *is the length of the shortest row in* U *and* $\hat{U}$*.*

Proof 9 *We will prove the first part and the second part holds naturally. By applying the variant of Davis-Khan theorem in Yu et al. [2015], we have*

$$\begin{aligned} ||\hat{U}\hat{O} - U||_F &\leq \frac{2^{3/2}\sqrt{nr}||W - E[W]||_\infty}{\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}} \\ &= \frac{2^{3/2}\sqrt{nr}\max\{\sqrt{4\max_{i=1,...,n}\theta_i\theta_j P_{g_i g_j}\log(2n/\epsilon)}, c\sqrt{\frac{\log p}{p}}\}}{\min\{\lambda_{R-1} - \lambda_R, \lambda_R - \lambda_{R+1}\}} \end{aligned} \tag{4.10}$$

Theorem 12 *(Error bound of* k*-means on normalized leading eigenvectors) Under*

NDCBM and mixture of Gaussian node attributes, the CSC algorithm gives a bound on mis-classification error as follows

$$||Z||/N \leq \frac{64mnr \max\{\sqrt{4\max_{i=1,\ldots,n}\theta_i\theta_j P_{g_ig_j}\log(2n/\epsilon)}, c\sqrt{\frac{\log p}{p}}\}^2}{\Delta N \min\{\lambda_{R-1}-\lambda_R, \lambda_R-\lambda_{R+1}\}^2}$$

Proof 10 *Similar to the proof in Theorem 2.*

We now add additional theoretical analysis for CSC-SCORE algorithm under NDCBM.

Lemma 10 *(Perfect clustering with CSC-SCORE under NDCBM). CSC algorithm achieve perfect clustering under DCBM settings.*

Proof 11 *We apply the matrix perturbation techniques (similar to lemma 6) to prove the following statements*

(1). Perfect clustering when $\alpha\mathcal{K}$ is a perturbation to $(1-\alpha)\Omega$.

Let $\mathcal{W} = (1-\alpha)\Omega + \alpha\mathcal{K} = \mathcal{W}_1 + \mathcal{W}_2$, and $(1-\alpha)\Omega = V\Lambda V'$. We anticipate the eigenvalue of $\mathcal{W}$ changes from λ_i to $\lambda_i + \hat{\lambda}_i$, the eigenvector changes from v_i to $v_i + \hat{v}_i$, then for the eigenvectors corresponding to the non-zero eigenvalues, we have,

$$(\mathcal{W}_1 + \mathcal{W}_2)(v_i + \hat{v}_i) = (\lambda_i + \hat{\lambda}_i)(v_i + \hat{v}_i)$$

Given the fact that $\mathcal{W}_1 v_i = \lambda_i v_i$, and remove the second order terms, we have

$$\mathcal{W}_2 v_i + \mathcal{W}_1\hat{v}_i = \lambda_i\hat{v}_i + \hat{\lambda}_i v_i$$

We also have

$$\hat{v}_i = \sum_{j=1}^{n} \omega_j v_j$$

since $\mathcal{W}_1$ *is a real and symmetric matrix, where* ω_j *is the coefficient and* v_j *is the* j*th eigenvector of* $\mathcal{W}_1$*. Insert this equation into Eq.(*), we arrive at*

$$\mathcal{W}_2 v_i + \sum_{j=1}^{n} \omega_j \lambda_j v_j = \lambda_i \sum_{j=1}^{n} \omega_j v_j + \hat{\lambda}_i v_i$$

We then multiply v_i' *to both sides results to*

$$v_i' \mathcal{W}_2 v_i + \sum_{j=1}^{n} \omega_j \lambda_j v_i' v_j = \lambda_i \sum_{j=1}^{n} v_i' \omega_j v_j + \hat{\lambda}_i v_i' v_i$$

Notice that in the last equation equals to 1 and the second term in the left-hand side and the first term in the right-hand side cancel each other. Thus we have

$$\hat{\lambda}_i = v_i' \mathcal{W}_2 v_i$$

With the similar technique, multiply v_j' *to both sides of Eq. (*), resulting*

$$v_j' \mathcal{W}_2 v_i + \sum_{k=1}^{n} \omega_k \lambda_k v_j' v_k = \lambda_i \sum_{k=1}^{n} v_j' \omega_k v_k$$

which evaluates to

$$v_j' \mathcal{W}_2 v_i + \omega_j \lambda_j = \lambda_i \omega_j$$

where we can find that

$$\omega_j = \frac{v_j' \mathcal{W}_2 v_i}{\lambda_i - \lambda_j}$$

then we have

$$\hat{v}_i = \sum_{j \neq i} \frac{v_j' \mathcal{W}_2 v_i}{\lambda_i - \lambda_j} v_j$$

When $R = 2$, i.e., a network with two communities, $i, j = 1, 2$, then we have

$$\hat{v}_1 = \frac{v_2' \mathcal{W}_2 v_1}{\lambda_1 - \lambda_2} v_2$$

and

$$\hat{v}_2 = \frac{v_1' \mathcal{W}_2 v_2}{\lambda_2 - \lambda_1} v_1$$

thus the element-wise ratio E results

$$\begin{aligned} E &= \frac{v_2 + \hat{v}_2}{v_1 + \hat{v}_1} \\ &= \frac{v_2 + \frac{v_1' \mathcal{W}_2 v_2}{\lambda_2 - \lambda_1} v_1}{v_1 + \frac{v_2' \mathcal{W}_2 v_1}{\lambda_1 - \lambda_2} v_2} \\ &= \frac{v_2 + m_1 \sum_{i=1}^{n} \theta_i^2 v_1}{v_1 + m_2 \sum_{i=1}^{n} \theta_i^2 v_2} \\ &= \frac{v_2 + m_1 c(\theta) v_1}{v_1 + m_2 c(\theta) v_2} \end{aligned} \tag{4.11}$$

Let's assume

$$v_1 = (\theta_1 x_1, \theta_2 x_2, ..., \theta_n x_n)'$$

and

$$v_2 = (\theta_1 y_1, \theta_2 y_2, ..., \theta_n y_n)'$$

Then for $i = 1, ..., n$, we have

$$\begin{aligned} E_i &= \frac{\theta_i y_i + m_1 c(\theta)\theta_i x_i}{\theta_i x_i + m_2 c(\theta)\theta_i y_i} \\ &= \frac{y_i + m_1 c(\theta) x_i}{x_i + m_2 c(\theta) y_i} \end{aligned} \tag{4.12}$$

which is free of θ_i.

(2). Perfect clustering when $(1-\alpha)\Omega$ is a perturbation to $\alpha\mathcal{K}$.

With the similar techniques as in (1), let $\mathcal{W} = (1-\alpha)\Omega + \alpha\mathcal{K} = \mathcal{W}_1 + \mathcal{W}_2$, and $\mathcal{W}_2 = U\Lambda U'$. We anticipate the eigenvalue of $\mathcal{W}$ changes from λ_i to $\lambda_i + \hat{\lambda}_i$, the eigenvector changes from u_i to $u_i + \hat{u}_i$, then for the eigenvectors corresponding to the non-zero eigenvalues.

Then the perturbated eigenvector $\hat{u}_i$ is

$$\hat{u}_i = \sum_{j \neq i} \frac{u_j' \mathcal{W}_1 u_i}{\lambda_i - \lambda_j} u_j$$

When $R = 2$, i.e., a network with two communities, $i, j = 1, 2$, then we have

$$\hat{u}_1 = \frac{u_2' \mathcal{W}_1 u_1}{\lambda_1 - \lambda_2} u_2$$

and

$$\hat{u}_2 = \frac{u_1' \mathcal{W}_1 u_2}{\lambda_2 - \lambda_1} u_1$$

thus the element-wise ratio E results

$$\begin{aligned} E &= \frac{u_2 + \hat{u}_2}{u_1 + \hat{u}_1} \\ &= \frac{u_2 + \frac{u_1' \mathcal{W}_1 u_2}{\lambda_2 - \lambda_1} u_1}{u_1 + \frac{u_2' \mathcal{W}_1 u_1}{\lambda_1 - \lambda_2} u_2} \\ &= \frac{u_2 + m_1 \sum_{i=1}^{n} \theta_i u_1}{u_1 + m_2 \sum_{i=1}^{n} \theta_i u_2} \\ &= \frac{u_2 + m_1 c(\theta) u_1}{u_1 + m_2 c(\theta) u_2} \end{aligned} \tag{4.13}$$

Let's assume

$$u_1 = (\theta_1 x_1, \theta_2 x_2, ..., \theta_n x_n)'$$

and

$$v_2 = (\theta_1 y_1, \theta_2 y_2, ..., \theta_n y_n)'$$

Then for $i = 1, ..., n$, we have

$$E_i = \frac{y_i + m_1 c(\theta) x_i}{x_i + m_2 c(\theta) y_i} \tag{4.14}$$

is free of θ_i.

4.7.3 Additional results for multiple species trees in the tree of life

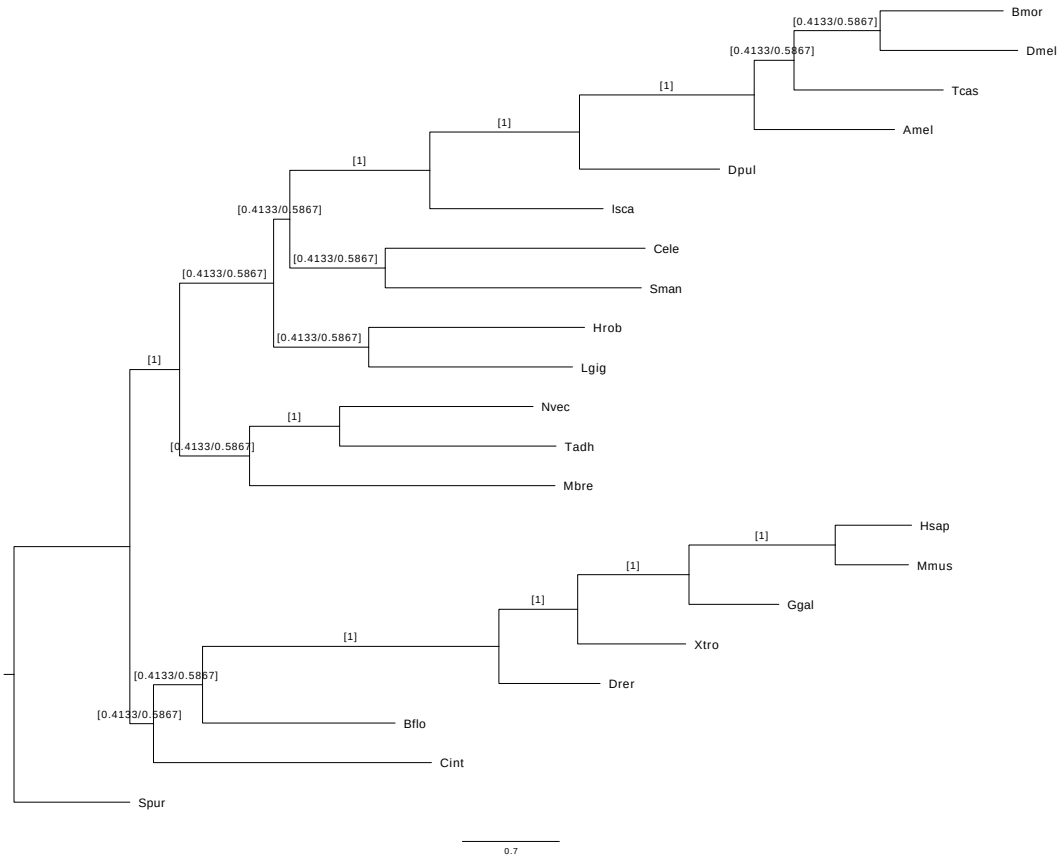

Figure 4.15: Consensus tree for multiple species trees in Metazoan data

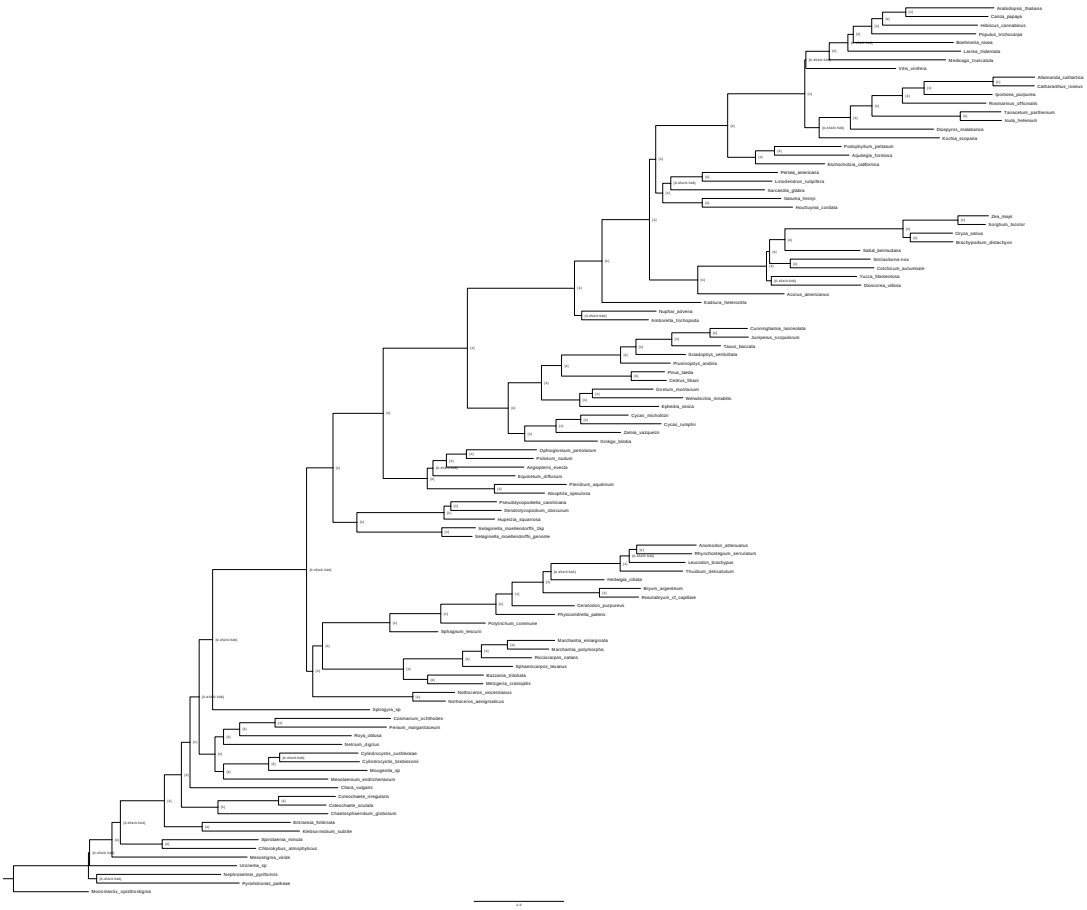

Figure 4.16: Consensus tree for multiple species trees in Plant data