Bayesian Modelling and Inference for Multiple Network Data

Anastasia Mantziou, MSci

Department of Mathematics and Statistics

Submitted for the degree of Doctor of Statistics at Lancaster university.

September 2021

Abstract

There is a growing need for analysing network data due to their prevalence in applications arising from various scientific fields. A broad literature has been developed for the statistical analysis of networks as single observations, while the formulation of statistical frameworks for modelling multiple network data has only recently been considered by researchers. This thesis contributes to the statistical analysis of multiple network data sets, where now each observation in the data comprises a network rather than a scalar quantity.

Our first contribution is the development of a Bayesian model-based approach for clustering multiple network data with respect to similarities detected in the connectivity patterns among the networks' nodes. Our model-based approach allows us to interpret the clusters with respect to a parameterisation, notably, through a network representative for each cluster. Our framework can also be formulated to detect networks in a population that are different from a majority group of networks. Extensive simulation studies show our model performs well in both clustering multiple network data and inferring the model parameters. We further apply our model on two real-world multiple network data sets resulting from the fields of Computing (Human Tracking Systems) and Neuroscience.

Our second contribution is twofold. First, we introduce a new network distance metric that measures dissimilarities between networks with respect to their cycles, motivated by an ecological application. Second, we propose a new Markov Chain Monte Carlo (MCMC) scheme for inferring the parameters of the intractable Spherical Network Family (SNF) model for multiple network data. Specifically, we introduce an Importance Sampling (IS) step within a Metropolis-Hastings (MH) algorithm that allows the approximation of the intractable normalising constant of the SNF model within the MH ratio. We explore the behaviour of the newly proposed distance metric and the performance of our MCMC scheme through simulation studies, and apply our algorithm on a real-world ecological application.

Acknowledgements

Firstly, I would like to thank my supervisors Simón Lunagómez and Robin Mitra who have been supporting and guiding me throughout the PhD not only with their valuable knowledge and expertise, but also with their constant kindness. I am very lucky I had them as supervisors. Thank you also to Paul Fearnhead for providing his useful insight on parts of this project.

I am also thankful for all the friends I met in Lancaster during the PhD who have made this experience even more joyful. Particularly, thank you to friends from B18 office. Special thanks to Eva for her limitless understanding and encouragement through the ups and downs of research. Thank you to Apo for being an unstrained company in stressful times. A profound thank you to Ierotheos for always supporting me in all my endeavours, not only in words but also in practice.

I am grateful for the love and encouragement of my family and close friends. Thank you to my cousin Mirto for always being there in good and bad times. Special thanks to my mother Aggeliki for always supporting me in every way, and last but not least, a big thank you and infinite love to my grandparents Vasilis and Maria.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Complex relationships in a system can often be described by a network representation. The advantage of describing such relationships through networks is the ability to exploit the structure of the network to reveal information about the system it describes. Data arising from various scientific fields exhibit such complex relationships, giving rise to diverse research questions. For example, neuroscientists are often interested in discovering abnormal connectivity patterns among brain regions of individuals suffering from a psychiatric disorder (Nelson et al. [2017], Lynall et al. [2010]), while Biologists are interested in uncovering relationships among a set of genes or proteins (Zhu et al. [2007]). The statistical analysis of a network observation involves the formulation of models that best describe the underlying mechanism that generates its edges. In this respect, different network models capture different types of information about a network's structure.

The advancement of the technological means that measure relationships among a set of objects has become prominent in recent years. This has led to the emergence of network populations resulting from multiple measurements taken over a set of nodes. An example arising from computational neuroscience, is the collection of networks representing interconnections of brain regions for a group of individuals (Zuo et al. [2014]). Thus, there is a growing need for analysing such complex data sets, in which the unit of observation is now a network rather than a scalar. The statistical analysis of multiple network data sets will be the key objective of the work presented in this thesis. An extensive review of the frameworks developed for modelling multiple network data is provided in Chapter 2.

In Section 1.1 we present the network notation that will be used through the remainder of this thesis, and in Section 1.2 we provide an overview of the statistical models formulated for the analysis of a network as a single observation. Finally, in Section 1.3 we outline the content and contributions for each Chapter of this thesis.

1.1 Notation

A network can be represented as a graph $\mathcal{G} = (V, E)$, where $V = \{1, \ldots, n\}$ represents the set of n nodes and E represents the set of observed edges in $\mathcal{G}$, with $E \in \mathcal{E}_n$ and $\mathcal{E}_n = \{(i,j)|i,j \in V\}$. Edges characterise different types of networks as follows:

- *Undirected* networks that have unordered edges such that $(i,j) \equiv (j,i)$.
- *Directed* networks that have ordered edges, such that (i,j) is distinct to (j,i).
- *Weighted* networks that have weights $w_{ij} \in \mathbb{R}^+$ assigned to their edges.
- *Simple* networks that have unique edges $(i,j) \in E$ and no self-edges $(i,i) \notin E$.

The analysis performed in this thesis focuses on undirected, simple graphs for the sake of simplicity. However, extensions to the directed case are straightforward. For an undirected network with n nodes, the number of possible edges is equal to $|\mathcal{E}_n| = \binom{n}{2}$.

A common way to mathematically represent a network is through an $n \times n$ *adjacency matrix* $A_{\mathcal{G}}$, with the $A_{\mathcal{G}}(i,j)$ entry of the matrix denoting the edge state of the (i,j) pair of nodes. Thence, the $(i,j)^{th}$ element of the adjacency matrix for a graph with binary edges is,

$$A_{\mathcal{G}}(i,j) = \begin{cases} 1, & \text{if an edge occurs between nodes i and j,} \\ 0, & \text{otherwise.} \end{cases}$$

The adjacency matrix of an undirected, simple graph is symmetric with $A_{\mathcal{G}}(i,j) = A_{\mathcal{G}}(j,i)$ and $A_{\mathcal{G}}(i,i) = 0$, for $i,j \in \{1, \ldots, n\}$.

By $\mathcal{G}_1, \ldots, \mathcal{G}_N$ we represent a population of N graphs, with corresponding adjacency matrices $A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}$. In this thesis, we assume that the networks in the population $\mathcal{G}_1, \ldots, \mathcal{G}_N$ are labelled, exchangeable and share the same set of n nodes. We represent the space of graphs with n nodes by $\{\mathcal{G}_{|n|}\}$, such that $\{\mathcal{G}_{|n|}\} = \{\mathcal{G} = (V, E) : |V| = n\}$. Thus the size of the space of undirected, simple graphs is $|\{\mathcal{G}_{|n|}\}| = 2^{\binom{n}{2}}$.

A way to quantify similarities among networks is through the use of distance metrics (Donnat and Holmes [2018]), which we denote by $d_{\mathcal{G}}(\cdot,\cdot)$. Two well-known distance metrics that we regularly discuss in this thesis are the following:

1. The *Hamming distance*, that counts the not in common edges and non-edges between two graphs $\mathcal{G}_k$ and $\mathcal{G}_l$ for $k, l \in \{1, \dots, N\}$, defined as:

$$d_H(\mathcal{G}_k, \mathcal{G}_l) = \sum_{i,j} \frac{|A_{\mathcal{G}_k}(i,j) - A_{\mathcal{G}_l}(i,j)|}{n \cdot (n-1)}$$

2. The *Jaccard distance*, that counts the not in common edges and non-edges between two graphs, accounting for the number of edges present in the graphs under comparison, using a normalisation with respect to the union of their edges, defined as:

$$d_J(\mathcal{G}_k, \mathcal{G}_l) = \frac{\sum_{i,j} |A_{\mathcal{G}_k}(i,j) - A_{\mathcal{G}_l}(i,j)|}{\sum_{i,j} \max(A_{\mathcal{G}_k}(i,j), A_{\mathcal{G}_l}(i,j))}$$

Various network properties in the literature allow us to evaluate diverse characteristics of a network. In this section we focus on the most popular network properties in the literature, which are also most relevant to the rest of this thesis. First, a well-known network property is the *density* of a graph, which is calculated by the ratio of the observed edges in the graph $\sum_{i,j} A_{\mathcal{G}}(i,j)$ to the number of possible edges $\binom{n}{2}$. Second, a fundamental property for undirected graphs is their *degree distribution*. The *degree* of node i expresses the number of nodes adjacent to node i, thus $Deg_i = \sum_{i \neq j} A_G(i,j)$. In this regard, the probability of a node in the graph to have degree k is equal to the proportion of nodes in the graph with degree k, $p_k = \frac{\{i \in V | Deg_i = k\}}{n}$, representing the *degree distribution* of the graph. Third, another useful network property is *transitivity*. Connections in a graph are called transitive, if a connection from node i to node j and from node j to node k, implies also a connection from node i to node k. In this respect, if there is a connection between nodes i and k we say that nodes i, j and k form a *triangle*, while if there is no edge between i and k we say that nodes i, j and k form a *connected triple*. Thence, the transitivity of a graph is calculated by the the number of triangles in the graph over the number of connected triples.

As shown by Maugis et al. [2017], two network features that are dominant with respect to the information they reveal about the networks' topology are trees and cycles.

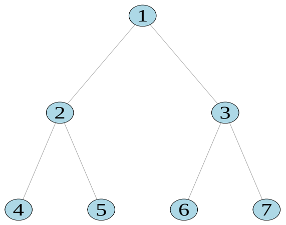

Figure 1.1: Example of tree network.

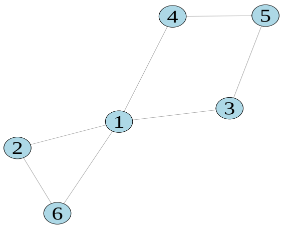

Figure 1.2: Example of graph with cycles $\{1-2-6-1\}$ and $\{1-3-5-4-1\}$.

A tree is a connected, undirected graph with no closed-loops. By the term connected we describe a graph in which every node is accessible through a path from every other node. Trees are often described by a *root node* from which edges to other nodes originate. The nodes branching down from the root node are called *leaves*. An example of such a structure is illustrated in Figure 1.1. Trees are a network motif of interest in the network literature. Another feature of interest about the structure of a network relevant to our study, are the *cycles*. A *cycle* in an undirected network is a sequence of connected nodes in which the only repeated nodes are the first and the last node in the sequence. An illustrative example of an undirected graph with two cycles is presented in Figure 1.2. We note here that $\{1-2-6-1\}$ and $\{1-6-2-1\}$ are considered as the same cycle.

1.2 Statistical models for networks

Networks have caught the interest of statisticians due to the complex structure they can exhibit. The statistical analysis of networks aims to describe the structure of a network by modelling the mechanism that generates a network's edges. Under this framework, there are various approaches that have been developed to model single network observations, which we briefly review in this section. For a more detailed review refer to Yang [2013], Salter-Townshend et al. [2012], Goldenberg et al. [2010] and Hand [2010].

In the simplest set up, a network is modelled under the assumption that edges can occur with the same probability. This model is known as the Erdös-Rényi (ER) model (Erdös and Rényi [1959], Gilbert [1959]). To simulate an undirected network with n nodes under the ER model, we need to specify a probability p of observing an edge which is common across all $\binom{n}{2}$ possible edges of the graph, and generate the edges of the network as

$$A_G(i,j) \sim \text{Bern}(p)$$

for $(i,j) \in \{1,\dots,n|i < j\}$. However, real networks usually exhibit more complex structures than the structure dictated by the ER model. Specifically, the nodes of the graphs generated under the ER model tend to have the same number of neighbours, however in most real-world applications, networks have few nodes with a large number of neighbours while the rest of the nodes have a substantially smaller number of neighbours.

A model that attempts to capture this characteristic of real-world networks, is the preferential attachment model (Barabási and Albert [1999]). Under the preferential attachment model, we assume an initial graph with n_0 nodes, and gradually add new nodes to that graph. Each new node added has a probability of connecting to one of the already existing nodes $i = 1,\dots,n_0$ that depends on the degree of node i. In this regard, nodes with higher degrees tend to gather more and more edges.

Another model that accounts for the degree of the network's nodes is the p_1 model. Under the p_1 model the probability of observing an edge between two nodes depends on some unknown node-specific parameters that are assumed to be constant. An extension to this model is the p_2 model which assumes that the node-specific parameters are random variables instead of constants, thus they are assumed to be drawn from some underlying distribution, and the parameters of that distribution are estimated.

A generalisation of the p_1 and p_2 models, is the Exponential Random Graph model (ERGM), which is a model from the the exponential family. The probability of observing a graph $\mathcal{G}$ under the ERGM, depends on network summary statistics $u(\mathcal{G})$, which are specified according to the information the analyst wants to capture. Common choices of $u(\mathcal{G})$ are the number of edges and the number of triangles in a graph. Thus the probability of observing $\mathcal{G}$ can be written as

$$P(\mathcal{G}) = exp(\theta' u(\mathcal{G}) - \psi(\theta))$$

where θ is the model parameter and $\psi(\theta)$ is the normalising constant.

A popular network model that reveals information about the communities formed by the network's nodes, is the Stochastic Block Model (SBM). The SBM assumes that each node belongs in some unobserved block $k \in \{1,\dots,K\}$, and the probability of observing an edge between two nodes depends on the block membership of the nodes denoted by $\{b_i\}_{i=1}^n$. Thus, Θ denotes the $K \times K$ symmetric probability matrix with elements θ_{kl}, for $k, l \in \{1 \ldots, K\}$, representing the probability of observing an edge between two nodes belonging in blocks k and l respectively. The connection between nodes i, j under the SBM is modelled as

$$A_G(i,j) \sim \text{Bern}(\theta_{b_i b_j}).$$

The classic SBM assumes that the number of communities K are known and prespecified. However, in the studies of Mørup et al. [2011], Tang and Yang [2011] and Schmidt and Morup [2013] the number of blocks K is treated as an unknown variable and is estimated along with the other parameters. An important extension to the classic SBM has been proposed by Karrer and Newman [2011], namely the degree-corrected SBM, that accounts for the heterogeneity of the degree of the nodes which is a commonly observed feature of real-world networks, as previously discussed.

Another important class of network models is the class of Latent Space Models which assume that the nodes of a network lie in an unobserved space and the probability of observing an edge between two nodes depends on the latent positions of the nodes. Hoff et al. [2002] were the first to introduce the latent space model assuming that each node i has a latent position z_i in space, and the distance between two nodes in the space determines the probability of observing an edge connecting them. Thence, the connection between nodes i, j is modelled as

$$\begin{gathered} A_G(i,j) \sim \text{Bern}(p_{ij}) \\ p_{ij} = P(A_G(i,j) = 1) = \frac{1}{1 + e^{-\eta_{ij}}}, \text{ with} \\ \eta_{ij} = \alpha + \beta' x_{ij} - |z_i - z_j| \end{gathered}$$

where $|z_i - z_j|$ denotes the Euclidean distance between z_i and z_j, and x_{ij} represents the observed covariates characterising the dyad.

An extension of the latent space model presented in Hoff et al. [2002] is the Random Dot Product graph, which was first introduced by Nickel [2008]. The random dot product graph assumes that the probability of observing an edge between nodes i, j depends on the dot product of the vectors z_i and z_j. This implies that the vectors of nodes that have the same direction in the space are more likely to be connected. Generalisations of the Random dot product graph have been proposed by Young and Scheinerman [2007] and Ng and Murphy [2019].

Another network generative model that expresses the limit of a sequence of finite graphs is the graphon (Lovász [2012], Borgs et al. [2008]). A graphon is a symmetric, measurable function $W : [0, 1]^2 \to [0, 1]$ representing the probability of observing an edge. Thus, to generate a graph with n nodes using a Graphon, we first need to sample n uniform random variables $x_i \sim \mathcal{U}([0, 1])$, for $i \in \{1, \ldots, n\}$, corresponding to each node of the graph, and generate an edge between nodes i, j as

$$A_G(i, j) \sim \text{Bern}(W(x_i, x_j)).$$

Eldridge et al. [2016] perform community detection in graphs generated under a graphon model, while Avella-Medina et al. [2018] develop centrality measures to identify important nodes in graphs which are assumed to be partially observed, and they are generated by a graphon.

1.3 Thesis outline and contributions

This thesis introduces two main contributions to the network literature, each presented in Chapters 3 and 4 respectively. Both contributions involve the statistical analysis of populations of networks. In this section, we present a summary of the content and contributions of each Chapter in this thesis.

In Chapter 2, we provide an analytic review of the diverse frameworks developed in the literature to model the underlying mechanism that generates a population of networks. Notably, we consider three distinct modelling frameworks, namely the latent space models, the distance-based models and the measurement error models. For each study presented, we discuss the key idea, the modelling and inferential framework developed and the potential limitations.

In Chapter 3, we introduce a model-based approach for clustering networks in a population with respect to similarities detected in the connectivity patterns of the networks' nodes, and interpreting the clusters with respect to our model parameterisation. Despite the growing research interest on modelling populations of networks, most studies in the literature do not account for the heterogeneity that can exist in a network population, while the approaches that aim to identify variations between network data entail key limitations. To address these limitations, we extend the work by Le et al. [2018] through the formulation of a mixture of measurement error models for clustering networks, assuming that networks lying within each cluster are noisy realisations of a true underlying cluster representative. We adopt a Bayesian modelling approach that allows us to simultaneously infer the cluster membership of the networks, together with model parameters characterising the distribution of the networks within each cluster as well as those that characterise the structure of the underlying cluster specific representatives. We assess the performance of our algorithm in clustering network populations, and inferring the model parameters through extensive simulation studies. In addition, our framework is flexible enough to answer a diverse range of applied questions with respect to the heterogeneity in a network population. These include being able to detect clusters of networks as well as inferring key different features between clusters through comparisons between the underlying representatives. In addition, when interest is in identifying observations that do not follow the distribution of the majority of the network data, the framework can also be formulated to detect outlying network observations. The approach will be illustrated through two different applied examples, one involving monitoring movement of people across a University Campus and another measuring individuals' connectivity patterns across different regions of the brain.

In Chapter 4, our contribution is twofold. First, we introduce an Importance Sampling (IS) step within a Markov Chain Monte Carlo (MCMC) algorithm, that allows us to make inferences for an intractable distribution for network populations, namely the Spherical Network Family of models introduced by Lunagómez et al. [2021]. Second, we develop a new network distance metric that quantifies similarities among networks with respect to their cycles. As a motivating example we consider a collection of networks representing aggressive interactions among species of fish, for which we are interested in identifying competitive behaviours among them. From an ecological perspective, the formation of cycles describes a type of competitive behaviour. To capture information with regard to cycles in networks, we develop a new network distance metric that measures dissimilarities between networks with respect to their cycles. One way to make inference for the fish network data is to implement the SNF model. The SNF model is a distance-based model that allows the statistician to specify a distance metric of interest to make inferences for network populations. Despite the flexibility that it allows with respect to the distance metric specification, the SNF model typically involves an intractable normalising constant. Lunagómez et al. [2021] implemented an Auxiliary Variable technique (Møller et al. [2006]) to overcome the intractability issue, however, this approach provides poor mixing results for some distance metric specifications such as the new metric we propose. Our proposed IS step within the MCMC algorithm allows us to overcome the mixing issue. We evaluate the performance of our proposed MCMC scheme through simulation studies and explore the behaviour of the new distance metric through synthetic and real data experiments.

In Chapter 5, we discuss potential directions for future work involving the development of approaches for (i) modelling network populations under the assumption that the network observations are dependent, and (ii) performing anomaly detection for the applications presented in this thesis. We conclude with some general remarks on future work directions for the analysis of networks.

Chapter 2

Review of Multiple Network Data Modelling

In the network literature there is a range of diverse studies focusing on the analysis of networks as single observations. The latent space approaches and measurement error models are two well-established frameworks in the network literature for modelling single network observations. Nonetheless, it is only until recently that researchers started focusing on modelling the mechanism that generates network populations. The research questions arising within the context of modelling multiple network data are the following. Would a latent space representation of the networks' nodes be useful to explain the mechanism that generates a network population? Would the assumption of a measurement error process be meaningful in the context of modelling network populations? These are research questions that have recently been considered in the literature for modelling multiple network data. On the other hand, a natural way to explain the variability of networks is through the use of distance metrics. This gives rise to another newly considered framework for modelling network populations based on the notion of a mean network representing a network population, with respect to a distance metric.

In this Chapter we review the studies developed for modelling network populations, under three frameworks. First, the latent space models reviewed in Section 2.1. Second, the distance-based models reviewed in Section 2.2, and third, the measurement error models reviewed in Section 2.3.

2.1 Latent space models for multiple network data

The concept of modelling a single network observation, assuming that the occurrence of an edge between two nodes depends on the positions of the nodes in a latent space, has been extensively considered in the network literature (Hoff et al. [2002], Young and Scheinerman [2007]). Recent studies (Gollini and Murphy [2016], Wang et al. [2017], Nielsen and Witten [2018], Arroyo et al. [2019], Durante et al. [2017]) on modelling multiple network data have borrowed this idea to build models for populations of networks with aligned vertex sets, under the assumption that the nodes lie in a common, unobserved subspace.

Hoff et al. [2002] introduce the Latent Space Model (LSM) for modelling single network observations, assuming that the probability of observing an edge between two nodes depends on the Euclidean distance between the nodes in a latent space. Building on this idea, Gollini and Murphy [2016] model multiple network data which share the same vertex set, assuming that each network observation can be represented by an LSM, while the latent positions of the nodes are unique across all networks. Thus, they obtain the Latent Space Joint Model (LSJM) which has the following form

$$P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} | Z, \alpha_1, \cdots, \alpha_N) = \prod_{k=1}^{N} \prod_{i \neq j}^{n} \frac{\exp(\alpha_k - |z_i - z_j|^2)^{A_{\mathcal{G}_k}(i,j)}}{1 + \exp(\alpha_k - |z_i - z_j|^2)},$$

where Z is an $n \times D$ matrix of the latent positions of the nodes in a D-dimensional space and $\boldsymbol{\alpha}$ is an N-length vector corresponding to the parameter of the k^{th} LSM fitted on the k^{th} network observation.

To infer the nodes' latent positions and the parameters of the LSJM, they implement an Expectation-Maximisation (EM) algorithm using Variational inference, as opposed to Hoff et al. [2002] who apply a Bayesian approach. In the expectation and maximisation steps of the Variational EM algorithm, instead of maximising the log likelihood of the data, they minimise the Kullback-Leibler divergence between the true posterior of their model and the variational posterior. Variational inference provides an alternative to MCMC approaches that can naturally lead to stochastic optimisation, thus significantly reducing the computational cost of implementing an MCMC scheme (Gopalan and Blei [2013]). With this approach, Gollini and Murphy [2016] tackle the problem of the computationally intensive task of sampling from the posterior through MCMC.

Under the assumption that the nodes of a network lie in an unobserved space, another approach for modelling the probabilistic mechanism that generates an edge between two nodes in a network, is proposed by Young and Scheinerman [2007] who introduce the Random Dot Product Graph (RDGP). Under the RDPG model, each vertex is assigned a vector in a predefined space, and the probability of an edge to occur between two nodes depends on the dot product of the vectors associated to the two nodes. Inspired by this idea, Durante et al. [2017], Wang et al. [2017] and Nielsen and Witten [2018] build models for multiple network data to infer their distribution.

Durante et al. [2017] model the unknown distribution of a network population $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$, assuming that each observed network is a realisation of a random variable $\mathcal{A}$. Specifically, they consider the case of undirected networks, thus instead of the adjacency matrix representation they use a vectorised version of the lower triangular elements of the adjacency matrix of $\mathcal{A}$, denoted as $\mathcal{L}(\mathcal{A})$. Under this specification, the goal is to infer the probability mass function of the random variable $\mathcal{L}(\mathcal{A})$.

To achieve this goal, the authors assume that the distribution of $\mathcal{L}(\mathcal{A})$ is a mixture of Bernoulli distributions, with probabilities that vary among the mixture components. Notably, the probability mass function of $\mathcal{L}(\mathcal{A})$ is:

$$P(\mathcal{L}(\mathcal{A}) = \alpha) = \sum_{h=1}^{H} v_h \prod_{l=1}^{n\cdot(n-1)/2} (\pi_l^{(h)})^{\alpha_l}(1-\pi_l^{(h)})^{1-\alpha_l}, \tag{2.1}$$

where α is a realisation of the random variable $\mathcal{L}(\mathcal{A})$, v_h is the probability of the realisation α to be assigned to the $h \in \{1, \cdots, H\}$ mixture component, and $\pi_l^{(h)}$ is the probability of observing an edge between the l^{th} pair of nodes in mixture component $h \in \{1, \cdots, H\}$.

The component specific probability of observing an edge between two nodes depends on a latent space configuration of the networks' nodes in the corresponding mixture component, through the formulation of the dot product of the vectors associated to that pair of nodes,

$$\boldsymbol{\pi}^{(h)} = \{1 + \exp(-Z - D^{(h)})\}^{-1}, \text{ where } D^{(h)} = \mathcal{L}(X^{(h)}\Lambda^{(h)}X^{(h)T}),$$

with $X^{(h)}$ representing the $n \times R$ matrix of the R latent coordinates for the n nodes,

$\Lambda^{(h)}$ denoting the $R \times R$ diagonal matrix with diagonal elements that determine the importance of each R dimension, and Z denoting a similarity vector of the edges shared by all mixture components $h \in \{1, \cdots, H\}$.

To infer the parameters $\boldsymbol{v}, \boldsymbol{\pi}^{(h)}$ of the model, the authors adapt a non-parametric Bayesian approach by implementing a Gibbs sampler algorithm that draws values for $\boldsymbol{v}, \boldsymbol{\pi}^{(h)}$ from their full conditional posteriors. For each posterior sample of $\boldsymbol{v}$ and $\boldsymbol{\pi}^{(h)}$, the authors simulate a large number of networks from model (2.1), and calculate for each network a summary measure of interest (e.g. density, degree mean, triangle frequency). Hence, they can also infer the distribution and the expectation of different network summary measures with respect to the distribution of the network data.

The model presented by Durante et al. [2017] can potentially address the problem of clustering network populations. However, a limitation of this formulation would be that each mixture component would be characterised by a specific network property and topological structure. This can be quite restrictive as under this construction only specific network characteristics would be potentially captured.

Wang et al. [2017] adapt a similar formulation to that of Durante et al. [2017], with some fundamental differences nevertheless. The ultimate idea behind the model of Wang et al. [2017] is still based on the RDPG model, however they do not assume a mixture model for the distribution of the network data. Instead, they consider a generalisation of the RDPG model as seen in Young and Scheinerman [2007], namely the Multiple Random Eigen Graphs (MREG) model. Under the MREG model, the network population has a common vertex set and the vertices share a common embedding space. Thence the probability of observing the adjacency matrix $A_{\mathcal{G}_k}$ of graph $\mathcal{G}_k$ is described as follows:

$$P(A_{\mathcal{G}_k}|\lambda_k, h_1, \cdots, h_D) = \prod_{i<j}(\sum_{d=1}^{D} \lambda_k(d)h_d(i)h_d(j))^{A_{\mathcal{G}_k}(i,j)}(1-\sum_{d=1}^{D} \lambda_k(d)h_d(i)h_d(j))^{1-A_{\mathcal{G}_k}(i,j)},$$

where $\lambda_k \in \mathbb{R}^D$ is the projection of $A_{\mathcal{G}_k}$ into the common subspace for each graph $\mathcal{G}_k$, and $\{h_d\}_{d=1}^{D}$ are shared across graphs $k = 1, \cdots, N$.

They estimate the parameters of the MREG model by implementing an alternating descent algorithm which optimises the D-dimensional Joint Embedding of graphs given by:

$$(\hat{\lambda_1}, \cdots, \hat{\lambda_N}, \hat{h_1}, \cdots, \hat{h_D}) = argmin_{\lambda_K, \|h_d\|=1} \sum_{k=1}^{N} \|A_{\mathcal{G}_k} - \sum_{d=1}^{D} \lambda_k(d) h_d h_d^T\|_F^2,$$

with $\|\cdot\|_F$ denoting the Frobenius norm.

Their algorithm solves a one-dimensional embedding problem at each iteration, which makes the optimisation task easier. However, this formulation does not guarantee the orthogonality of the vectors of each of the D dimensions. Nielsen and Witten [2018] tackle this limitation of the MREG model by presenting the Multiple Random Dot Product Graph (multi-RDPG) model which is a direct extension of the RDPG, and implement an algorithm which estimates all D dimensions concurrently in each iteration.

Specifically, the probability of observing an edge between two nodes in graph $\mathcal{G}_k$ under the multi-RDPG model is:

$$P(A_{\mathcal{G}_k}(i,j) = 1) = f(W^k(i,j)), \quad W^k = U\Lambda^k U^T, \quad k = 1, \cdots, N$$

The key differences between the multi-RDPG model and the MREG model presented in Wang et al. [2017] are:

- For the multi-RDPG, the columns of the $n \times D$ matrix U are constrained to be orthogonal, and the $D \times D$ diagonal matrices $\Lambda^1, \cdots, \Lambda^N$ are constrained to have non-negative entries. Thus, the $n \times D$ matrix W^k is positive semi-definite, which allows the interpretation of its rows as the coordinates of the n nodes in a D-dimensional space.

- For N=1, the multi-RDPG model breaks down into the RDPG model for a single network observation.

Nielsen and Witten [2018] fit the multi-RDPG model by solving the following optimisation problem, for a population of N networks:

$$\min_{U \in \mathbb{R}^{n \times D}, U^T U = I, \Lambda^1, \cdots, \Lambda^N \in \Delta_+^D} \sum_{k=1}^{N} \|A_{\mathcal{G}_k,+} - U\Lambda^k U^T\|_F^2, \tag{2.2}$$

However, the optimisation problem (2.2) does not have a closed-form solution, thus [Nielsen and Witten, 2018] implement an alternating minimisation approach which alternates between updating U and $\Lambda^1, \cdots, \Lambda^N$, at each iteration. The key difference to the algorithm applied in Wang et al. [2017], is that Nielsen and Witten [2018] estimate simultaneously all D dimensions of matrix U at each iteration.

In a similar set up, Arroyo et al. [2019] model a population of networks which share the same set of nodes, assuming that the nodes lie in a common subspace for all networks, while allowing heterogeneity to exist within and across the networks. The probability of observing the $A_{\mathcal{G}_k}$ adjacency matrix for the k^{th} network under the Common Subspace Independent Edge graphs (COSIE) model is:

$$P(A_{\mathcal{G}_k}|V, R_k) = \prod_{i<j} (V(i)^T R_k V(j))^{A_{\mathcal{G}_k}(i,j)} (1 - V(i)^T R_k V(j))^{1-A_{\mathcal{G}_k}(i,j)},$$

where V represents the nodes' common subspace and R_k is the score matrix of network k, depicting the heterogeneity among the $k = \{1, \cdots, N\}$ networks.

To estimate the common subspace V and the score matrices $\{R_k\}_{k=1}^N$, the authors apply a Multiple Adjacency Spectral Embedding (MASE) algorithm. The first step of the MASE algorithm is to obtain the Adjacency Spectral Embedding (ASE) of the adjacency matrix $A_{\mathcal{G}_k}$ by decomposing $A_{\mathcal{G}_k}$, which gives an estimator for V_k for $k = \{1, \cdots, N\}$. Finally, they infer the V common subspace and the score matrices $\{R_k\}_{k=1}^N$, by concatenating the estimators $\hat{V}_k$ and applying a Singular Value Decomposition.

In the studies of Arroyo et al. [2019] and Wang et al. [2017], the authors also test the performance of their models in the inference task of graph classification. However, both studies use non model-based approaches for inferring underlying groups of networks. On the one hand, Wang et al. [2017] use a 1-nearest neighbour classifier, given the estimators they obtain for the loadings $\{\hat{\lambda}_i\}_{i=1}^N$. On the other hand, Arroyo et al. [2019] obtain the Frobenious distance of the estimated parameters of their model, and conduct a Multi Dimensional Scaling (MDS) to infer underlying groups of networks. In addition, their approach is constrained in detecting similar community structures among networks.

In Chapter 3, we provide a model-based framework for clustering multiple network data, that has the advantage of inferring the clusters with respect to the structure of the networks, as well as interpreting each cluster via a parametrisation.

2.2 Modelling multiple network data using metrics

Distance functions are used to measure the distance between two elements in a metric space. Such elements may not only be real numbers, but also vectors or matrices. Networks fall into the category of elements that can be described via a matrix, namely the adjacency matrix which contains information about the interconnections of the graph's nodes.

In the network literature, distance metrics have been used to measure similarities among networks, capturing either local (structural distances) or global (spectral distances) characteristics of the networks (Donnat and Holmes [2018]). Specifically, spectral distances assess the overall structure of the graphs using some representation of the adjacency matrix of the graph. One approach to modelling multiple network data, relies on the notion of an average network that represents a network population with respect to a specified distance metric.

Ginestet et al. [2017] build their theoretical framework on the idea of a representative network, motivated by a population of network data that represents measurements of individuals' brain connections through neuroimaging. Notably, the authors seek to investigate whether there are brain connectivity differences between populations of networks that are grouped by the individuals' sexes or ages, considering global network characteristics.

To fulfil this aim, they obtain analogues of the one and two-sample t-tests for network data, under the assumption of an average network that summarises the information contained in a network population. To obtain an average network, they consider the notion of the sample Fréchet mean which can be defined as follows:

$$\hat{\mu}_n(Y) := \text{argmin}_{x \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} \rho^2(x, Y_i),$$

for a $(\mathcal{X}, \rho)$ metric space and for any sample of realisations denoted by $Y := \{Y_1, \cdots, Y_n\}$.

Thence, to exploit the notion of the Fréchet mean, they first need to identify a metric space where the network realisations lie, as well as a distance function ρ with respect to that space. They define a metric space for networks utilising the Laplacian representation of the networks, which can be defined as $L_{\mathcal{G}_k} = D_{\mathcal{G}_k} - A_{\mathcal{G}_k}$, where $A_{\mathcal{G}_k}$ denotes the $n \times n$ adjacency matrix of graph $\mathcal{G}_k$ and $D_{\mathcal{G}_k}$ denotes a diagonal matrix with diagonal entries the vertex degrees. Thus, they obtain a space of Laplacians, with corresponding distance function ρ the Frobenius distance between the Laplacians, which takes the form

$$\rho(L_{\mathcal{G}_k}, L_{\mathcal{G}_m}) := \sum_{i,j=1}^{n} (L_{\mathcal{G}_k}(i,j) - L_{\mathcal{G}_m}(i,j))^2,$$

for any pair of the Laplacian matrices $L_{\mathcal{G}_k}, L_{\mathcal{G}_m}$ of graphs $\mathcal{G}_k, \mathcal{G}_m$. In this regard, the authors can specify the sample Fréchet mean with respect to the Frobenius distance function ρ.

Having specified a metric space and a distance function with respect to that space, they derive an analogue to the central limit theorem for sequences of network data. This allows the definition of a t-statistic for a one-sample test under the null hypothesis that the true Fréchet mean is equal to some value Λ_0. Thus, under the null hypothesis $H_0 : \mathbb{E}(L) = \Lambda_0$ the test statistic

$$T_1 := N(\phi(\hat{L}) - \phi(\Lambda_0))'\hat{\Sigma}^{-1}(\phi(\hat{L}) - \phi(\Lambda_0)),$$

converges to a χ^2-distribution with $p := \binom{d}{2}$ degrees of freedom, $\phi(L)$ denoting the half-vectorisation of L and $\hat{\Sigma} := 1/(N-1)\sum_{i=1}^{N}(\phi(L_i) - \phi(\hat{L}))(\phi(L_i) - \phi(\hat{L}))'$ denoting the sample covariance. Respectively, they obtain a T_k statistic to test the null hypothesis $H_0 : \Lambda_1 = \cdots = \Lambda_k$, for k independent samples of network populations.

In a similar set up, Kolaczyk et al. [2017] construct a theoretical framework in order to define the notion of an average network, for a population of unlabelled networks. By the term unlabelled network, they characterise a network in which each vertex does not have a unique label. To accomplish this goal, equivalently to Ginestet et al. [2017], they consider the notion of the sample Fréchet mean and specify a space where unlabelled graphs lie, as well as a distance function in that space.

To determine a space for unlabelled graphs, they use equivalence classes that relate the space of labelled graphs to the space of unlabelled graphs. Notably, they consider the group of permutations Σ_n of $\{1, \cdots, n\}$ nodes, and obtain the quotient space

$$\mathcal{U}_n = \mathcal{L}_n/\Sigma_n,$$

which is the space of unlabelled graphs, with $\mathcal{L}_n$ denoting the space of labelled graphs.

They further specify a distance function that measures how close two networks are in the space of unlabelled networks. Notably, the authors consider the Procrustean distance which has the form

$$\rho([\vec{x}],[\vec{y}]) := \min_{\sigma_1,\sigma_2\in\Sigma_n} d_E(\sigma_1\cdot\vec{x},\sigma_2\cdot\vec{y}),$$

where $[\vec{x}],[\vec{y}]\in\mathcal{U}_n$ are the vectorisations of the unlabelled networks and $d_E(\cdot,\cdot)$ is the Euclidean distance between two elements.

Analogously to Ginestet et al. [2017], they obtain a central limit theorem for the sample Fréchet mean in the space of unlabelled graphs, which provides the groundwork for performing statistical inference in that space. Notably, they construct a test statistic for k independent samples of networks, for testing whether all k samples share the same true, unknown mean μ, i.e test the hypothesis $H_0 : \phi(\mu^{(1)}) = \cdots = \phi(\mu^{(k)})$. Under the null hypothesis they show that

$$T_k := \sum_{j=1}^{k} n_j(\phi(\mu_{j,n_j}) - \phi(\mu_n))'\widehat{\Xi}^{-1}(\phi(\mu_{j,n_j}) - \phi(\mu_n)) \longrightarrow \chi^2_{(k-1)D},$$

where n_j denotes the size of the j^{th} sample with $n = \sum_{j=1}^{k} n_j$ for $j\in\{1,\cdots,k\}$, μ_{j,n_j} represents the empirical mean of the j^{th} sample, μ_n represents the empirical mean of the whole sample n, and $\widehat{\Xi} := \sum_{j=1}^{k}\widehat{\Xi}_j/n_j$ is a pooled estimate of covariance, with each $\widehat{\Xi}_j$ denoting the covariance matrices estimates of each subsample.

In the aforementioned studies, the authors approach the problem of interpreting a network population by extending classical statistics methodologies (e.g. t-tests), to datasets where the observational units are networks. However, these approaches introduce computational challenges as the size of the networks and the size of the population of networks increases, especially due to the estimation of the covariance matrices required. A more natural modelling approach for multiple network data, that also utilises the notion of the Fréchet mean, is introduced in the study of Lunagómez et al. [2021] who adopt a Bayesian perspective to make inferences for network populations. In Section 2.2.1, we review the model presented by Lunagómez et al. [2021] which has pivotal role in the work presented in Chapter 4 of this thesis.

A fundamental assumption made in all aforementioned studies, is the uniqueness of the Fréchet mean in a network population, which can be an unrealistic modelling assumption for many real-world data sets. This constraint does not allow the conception of the heterogeneity within a network population, which is common especially in data sets where each observational unit corresponds to an individual.

In the recent work by Arora et al. [2020] the authors investigate the ability of existing network models to capture the variability in a network population, utilising the idea of a dissimilarity space with a set of corresponding distance metrics on that space. Specifically, the authors consider a set of four network generative models (Chung and Lu [2002], the ERGM by Robins et al. [2007], the dk-random graphs by Orsini et al. [2015], the ABNG by Arora and Ventresca [2017]), and fit them using a single network observation from an observed network population. Then, they synthesise a network population using the fitted generative model, and compare the synthesised network population to the observed network population. To achieve this comparison, they assume a dissimilarity space and place the networks in that space with respect to a set of dissimilarity measures that capture node based properties of the networks.

The experimental results obtained, show that all of the four network generative models considered are unable to adequately encode the variability in a network population. They further investigate that using entropy as a measure for evaluating the variability in the properties of the networks under comparison. Specifically, they define the *edge existence entropy* as,

$$H_e(A_{\mathcal{G}}(i,j)) = -(p_1 \log_2 p_1 + p_0 \log_2 p_0),$$

between nodes i, j with $p_1 = P(A_{\mathcal{G}}(i,j) = 1)$ and $p_0 = P(A_{\mathcal{G}}(i,j) = 0)$, which explains the average uncertainty of edge existence in a network population; and the *geodesic entropy* as,

$$H_e(\delta_{ij}) = -\sum_{d \in \Delta} p_d \log_{|\Delta|} p_d,$$

with δ_{ij} denoting the hop distance between nodes i, j, p_d representing the proportion of networks in the population with $\delta_{ij} = d$ and Δ determined empirically.

In their experiments, they evaluate the variability contained in a population of structural brain networks using the edge existence and geodesic entropies. Their exploration verifies their statement that existing network generative models fail to capture the variability of a network population, and highlight the need of the development of network models that explicitly account for that variability in the network data.

2.2.1 The Spherical Network Family of Models (SNF)

In this Section, we describe the Spherical Network Family (SNF) of models presented by Lunagómez et al. [2021], that highly motivated the methods developed in Chapter 4 of this thesis.

Lunagómez et al. [2021] extend the idea of the Normal distribution to the case of network data, assuming an underlying mean network representing the network population and a dispersion parameter denoting the variation of the networks about this mean. They express the mean network in terms of a Fréchet mean, as seen in the studies of Ginestet et al. [2017] and Kolaczyk et al. [2017], and the dispersion parameter in terms of an entropy. Under this construction, they obtain the probabilistic mechanisms that generate data sets of multiple network data.

The functional form of the SNF model is inspired by the symmetry of the density property, which aligns with the assumption of a unique Fréchet mean in Lunagómez et al. [2021]. A model that presents that property is the Ising model, which is a special case of the random field spherical model in the Physics literature (Metz et al. [2014]), which also motivated the name of the SNF model. Thus, the likelihood of observing a population of undirected and unweighted graphs $\mathcal{G}_1, \cdots, \mathcal{G}_N$ under the SNF model is

$$P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} \mid A_{\mathcal{G}^m}, \gamma) \propto \exp\left\{ -\gamma \cdot \sum_{i=1}^{N} \phi(d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m})) \right\},$$

where $\mathcal{G}^m$ is the Frechét mean, $d_G(\cdot, \cdot)$ is a distance metric, γ is the temperature with $\gamma > 0$, $\phi(\cdot) > 0$ is a monotone increasing function and the partition function of this model is the reciprocal of

$$Z(A_{\mathcal{G}^m}, \gamma) = \sum_{A_{\mathcal{G}} \in \{\mathcal{G}_{|n|}\}} \exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}}, A_{\mathcal{G}^m}))\},$$

where $\{\mathcal{G}_{|n|}\}$ is the space of n-node networks.

The term temperature for parameter γ is also inspired by the Physics literature for this class of models. However, we note here that in the rest of this thesis we will refer to γ as the dispersion parameter, which is also another term used by Lunagómez et al. [2021].

A special case of the SNF model with the Hamming distance metric specification for $d_G(\cdot, \cdot)$, is the Centered Erdös-Rényi (CER) model, also presented in Lunagómez et al. [2021]. As introduced in Section 1.1, the Hamming distance is defined as

$$d_H(\mathcal{G}_k, \mathcal{G}_l) = \sum_{i,j} \frac{|A_{\mathcal{G}_k}(i,j) - A_{\mathcal{G}_l}(i,j)|}{n \cdot (n-1)}.$$

Under the Centered Erdös-Rényi (CER) model proposed by Lunagómez et al. [2021], a population of networks is generated by perturbing the edges of a centroid network $\mathcal{G}^m$ using a Bernoulli distribution with probability α, as follows:

$$A_{\mathcal{G}}(i,j) \mid (A_{\mathcal{G}^m}(i,j), \alpha) = |A_{\mathcal{G}^m}(i,j) - Z(i,j)|,$$

where notation $|$ is used to represent dependence, $\mathcal{G}^m$ is the Frechét mean and $Z(i,j)$'s are *iid* Ber(α), with $0 < \alpha < 0.5$. Thence, the likelihood of observing a population of undirected and unweighted $\mathcal{G}_1, \cdots, \mathcal{G}_N$ graphs under the CER model is

$$P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} | A_{\mathcal{G}^m}, \alpha) = \prod_{i=1}^{N} \alpha^{d_H(A_{\mathcal{G}_i}, A_{\mathcal{G}^m})} \cdot (1-\alpha)^{\frac{n(n-1)}{2} - d_H(A_{\mathcal{G}_i}, A_{\mathcal{G}^m})}$$

where $d_H(\cdot, \cdot)$ denotes the Hamming distance metric and n is the number of the networks' vertices.

To infer the parameters of the model, the authors specify a prior distribution for the centroid network and the dispersion parameter, and implement an MCMC scheme to draw samples from their posterior distributions. However, it is not possible to directly implement a Metropolis-Hastings algorithm for the SNF model, as its normalising constant depends on the parameters of the model. Thus, the normalising constants do not cancel out in the Metropolis-Hastings ratio.

To tackle the problem of the intractable normalising constants, the authors apply the Auxiliary Variable technique presented in Møller et al. [2006]. This method involves the use of an auxiliary variable with proposal density that has the same functional form as the likelihood, thus allowing the normalising constants of the proposal density and the likelihood to cancel out, and the Metropolis-Hastings algorithm to be applied as normal.

However, a limitation with the implementation of the Auxiliary Variable technique (Møller et al. [2006]) for the SNF model, is the poor mixing and the non-convergence of the MCMC chains for some distance metrics specifications. In Chapter 4, we present an alternative method, namely an Importance Sampling step within the MCMC, that allows us to address this limitation.

Respectively to the studies of Ginestet et al. [2017], Kolaczyk et al. [2017] and Arora et al. [2020] presented in this Section, Lunagómez et al. [2021] assume that a network population is generated by a common generative process. However, this assumption may lead to a misconception of the information contained in such data sets, as it does not account for the variability of the network data within a population.

In Chapter 3, we aim to relax the aforementioned assumptions through the use of mixture models for interpreting the process that generates a network population. Our model allows for the existence of clusters of networks within a network population, and represents each cluster through a network representative and some measure of dispersion of the network data about the representative. Thus, we do not only identify subpopulations of networks, but also allow their interpretation through their parameterisation.

2.3 Measurement Error Models for multiple network data

A common issue arising from the data collection process, is that the final data sample may not encompass all the true, underlying information due to the presence of noise. This implies that if the sampling procedure was repeated, the resulting sample might not be identical to the previously collected sample. A fundamental source of noise found in network data originates from the various measurement tools used for the construction of networks, i.e. the processes used to measure an interaction (edge) between two objects (nodes). For example, in the case of biological networks (e.g. protein interaction networks and gene networks), different high-throughput methods used to record interactions between genes or proteins can lead to high rates of falsely recorded edges (Sprinzak et al. [2003]).

Researchers focusing on the statistical analysis of networks as single observations, have developed methods to incorporate the uncertainty of falsely observing edges or non-edges in a network. Such studies involve predicting network topologies accounting for the falsely non-observed edges (Jiang et al. [2011]), estimating the adjacency matrix from a set of noisy entries (Chatterjee et al. [2015]), classifying nodes of networks with errorful edges (Priebe et al. [2015]), developing a regression model for networks assuming that the observed network is a perturbed version of a true unobserved network (Le and Li [2020]) and performing Bayesian inference on the network's structure utilising information from measurements (Young et al. [2020]). Another group of studies focuses on the propagation of the error to network summary statistics (Balachandran et al. [2017], Chang et al. [2020]) and to estimators of average causal effects under network interference (Li et al. [2021]), when the error arises from a measurement process used to construct the network. Recent studies (Le et al. [2018], Newman [2018b], Peixoto [2018]), extend the idea of a measurement error process to the case of multiple network data sets, with ultimate goal to model the probabilistic mechanism that generates a network population under the assumption that the networks therein are noisy realisations of a true unobserved network.

An example of multiple noisy network realisations comes from neuroimaging, where nodes correspond to different regions of an individual's brain, and edges correspond to the connections among the different regions. This type of network data regularly involves measurement errors due to the different preprocessing methods and thresholds used to consider a connection (edge) or not (non-edge) between two regions of the brain. In the study of Buchanan et al. [2020], the authors argue the effect of different thresholding levels and methods on the resulting networks and their associations with external variables, by considering various network measures. Nonetheless, there is no study to our knowledge to date that has looked into the effect of thresholding against network models. Inspired by the noisy nature of brain networks, Le et al. [2018] build an inferential framework to estimate the adjacency matrix of a true underlying network, resulting from multiple noisy network realisations.

Le et al. [2018] assume that the observed network population results from perturbing the edges of a true underlying network, with some probabilities that depend on the existence or non-existence of an edge in the true network. Specifically, they assume that the noise in the data has the form of false positive and false negative edges with respect to the true underlying network. Thence, if there is an edge between two nodes in the true network, the observed network will not have an edge between the same two nodes with some probability called false negative probability, while if there is no edge between two nodes in the true network, the observed network will have an edge between the two nodes with some probability called false positive probability. This can be interpreted mathematically as follows:

$$\text{If } A_{\mathcal{G}}(i,j) = 1 \text{ then } A_{\mathcal{G}_k}(i,j) \sim \text{Bernoulli}(1 - Q_{ij}),$$
$$\text{If } A_{\mathcal{G}}(i,j) = 0 \text{ then } A_{\mathcal{G}_k}(i,j) \sim \text{Bernoulli}(P_{ij}),$$

where $A_{\mathcal{G}}(i,j)$ is the (i,j) entry of the adjacency matrix of the true network, $A_{\mathcal{G}_k}(i,j)$ is the (i,j) entry of the adjacency matrix of the k^{th} observed network, P_{ij} is the (i,j) entry of an $n \times n$ matrix P of false positive probabilities and Q_{ij} is the (i,j) entry of an $n \times n$ matrix Q of false negative probabilities.

Thus, the likelihood of observing $A_{\mathcal{G}}$ given the data $\{A_{\mathcal{G}_k}\}_{k=1}^{N}$, with $W = \mathbb{E}A_{\mathcal{G}}$, is

$$\begin{aligned} L(A_{\mathcal{G}}; A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}) = \prod_{(i,j):i<j} \left[W_{ij} \cdot \prod_{k=1}^{N} (1 - Q_{ij})^{A_{\mathcal{G}_k}(i,j)} Q_{ij}^{1 - A_{\mathcal{G}_k}(i,j)} \right]^{A_{\mathcal{G}}(i,j)} \times \\ \times \left[(1 - W_{ij}) \cdot \prod_{k=1}^{N} (P_{ij})^{A_{\mathcal{G}_k}(i,j)} (1 - P_{ij})^{1 - A_{\mathcal{G}_k}(i,j)} \right]^{1 - A_{\mathcal{G}}(i,j)} . \end{aligned} \quad (2.3)$$

In the case of neuroimaging applications, inferences on the sizes of p and q can potentially be informative for constructing a threshold that indicates the presence or absence of an edge between two brain regions.

Studies have shown that brain imaging networks can reveal underlying clusters of brain regions that are densely connected while an individual either performs a task or rests (Garcia et al. [2018], Power et al. [2011]). Subsequently, Le et al. [2018] impose some community structure on the true latent network motivated by the neuroimaging application. Specifically, they assume that the true network $A_{\mathcal{G}}$ has the structure of a Stochastic Block Model (SBM), which is one of the most commonly used models for community detection in network analysis. They further assume that the matrices of false positive probabilities P, false negative probabilities Q and the expectation of $A_{\mathcal{G}}$, $W = \mathbb{E}A_{\mathcal{G}}$, share the same block structure as that of $A_{\mathcal{G}}$.

To obtain estimates for the true adjacency matrix $A_{\mathcal{G}}$, the false positive/negative probabilities P, Q and the expectation of the true adjacency W, they first apply a Spectral Clustering algorithm to estimate the SBM structure for known number of blocks B, for an initial estimate of the true adjacency $A_{\mathcal{G}}$, defined as $\hat{A}_{\mathcal{G}}(i,j) = \mathbb{1}(S_{ij} \geq N/2)$, with $S = \sum_{k=1}^{N} A_{\mathcal{G}_k}$.

After having specified the block membership $\{c_i\}_{i=1}^{n} \in \{1, \cdots, B\}$ of the n nodes of the initial estimate of $A_{\mathcal{G}}$, they implement an Expectation-Maximisation (EM) algorithm that estimates the parameters of their model by taking the conditional expectation of the log of likelihood (2.3) (E step) and maximising it with respect to W, P, Q (M step). The EM algorithm is iterated until convergence, and the Spectral Clustering algorithm is then applied on the new estimate of $A_{\mathcal{G}}$ obtained from the EM algorithm. The whole process is then repeated for T iterations.

One limitation of the method proposed by Le et al. [2018] is that their algorithm does not simultaneously update the parameters of their model and the block membership of the true network's nodes, as this would require the development of new techniques. Newman [2018b] suggests an alternative method that potentially addresses this limitation, using the Maximum a Posteriori (MAP) estimates for estimating an underlying true network from noisy data, as well as the parameters of his model.

Notably, the method proposed by Newman [2018b] involves the predefinition of two models:

1. A network model, that encodes information about the structure of the true network. The probabilistic mechanism that generates a network of structure A is represented by $P(A|\gamma)$, where γ denotes the parameters of the model.

2. A data model, that explains the measurement error mechanism that generates the noisy data from the true network with structure A. The probability of observing the D data model is represented as $P(D|A;\theta)$, where θ denotes the parameters of the data model.

Thus, the joint posterior distribution of the parameters γ, θ and the true network structure A is,

$$P(A, \gamma, \theta | D) = \frac{P(D|A, \theta) \cdot P(A|\gamma) \cdot P(\gamma) \cdot P(\theta)}{P(D)}.$$

Summing over all the possible network structures A, the author obtains the posterior distribution of the models' parameters,

$$P(\gamma, \theta|D) = \sum_A P(A, \gamma, \theta|D), \tag{2.4}$$

which is then maximised with respect to γ, θ to obtain their MAP estimates.

Specifically, the author takes the log of the posterior (2.4), and applies the Jensen's inequality which gives:

$$\log P(\gamma, \theta|D) = \log \sum_A P(A, \gamma, \theta|D) \geq \sum_A q(A) \log \frac{P(A, \gamma, \theta|D)}{q(A)}, \tag{2.5}$$

where $q(A)$ satisfies the equation $\sum_A q(A) = 1$.

Inequality (2.5) becomes an equality for,

$$q(A) = \frac{P(A, \gamma, \theta|D)}{\sum_A P(A, \gamma, \theta|D)} = P(A|D, \gamma, \theta), \tag{2.6}$$

which can be interpreted as the probability distribution over networks with structure A.

Thus for $q(A)$ given as per (2.6), Newman [2018b] derives the MAP estimates of γ, θ by maximising the right-hand side of (2.5), which is equal to the posterior distribution of these parameters. Under this set up, Newman [2018b] implements an EM algorithm that iterates over the maximisation of the posterior distribution with respect to $q(A), \gamma, \theta$, until convergence is achieved.

An example of a network model that could be used to describe the structure of the true network, is the Stochastic Block Model (SBM), which has also been considered in the study of Le et al. [2018]. Newman [2018a] suggests that the combination of an SBM and a suitably chosen data model, can tackle the limitation of the non-concurrent estimation of the block structure and the models' parameters seen in Le et al. [2018]. However, Newman [2018b] does not explicitly obtain the complete algorithm for the case in which A is an SBM, as the calculations involved would require the use of new techniques as stated in Newman [2018b].

A similar formulation to that presented by Newman [2018b] is adapted in the study of Peixoto [2018], who infers a true underlying network $A_{\mathcal{G}}$ from either a single or multiple noisy network realisations $D = \{A_{\mathcal{G}_k}\}_{k=1}^N$, under the assumption of a known or unknown measurement error process. The framework introduced in Peixoto [2018] consists of the specification of a network generative model denoted by $P(A_{\mathcal{G}}|\theta)$, which encodes information about the structure of the true underlying network, and a measurement error model denoted by $P(D|A_{\mathcal{G}}, \phi)$, that describes the type of measurement error contained in the network realisations.

The fundamental difference between the studies of Peixoto [2018] and Newman [2018b] is in the type of inference adapted in each of these studies. Specifically, Peixoto [2018] performs Bayesian inference in order to decode the uncertainty of the measurement process that is reflected in the unobserved true network. Thence, instead of obtaining a single MAP estimate of the true underlying network as seen in Newman [2018b], Peixoto [2018] obtains draws from the posterior distribution of the true network given the noisy data, described as follows:

$$P(A_{\mathcal{G}}|D) = \frac{P(D|A_{\mathcal{G}}) \cdot P(A_{\mathcal{G}})}{P(D)}, \tag{2.7}$$

where

$$P(D|A_{\mathcal{G}}) = \int P(D|A_{\mathcal{G}}, \phi)P(\phi)d\phi \text{ and } P(A_{\mathcal{G}}) = \int P(A_{\mathcal{G}}|\theta)P(\theta)d\theta,$$

with $P(\theta)$ and $P(\phi)$ representing the prior distributions of the parameters of the network generative model and the measurement error model respectively.

In order to draw values from the posterior distribution of the true network (2.7), one first needs to define a network generative model and a measurement error model. Peixoto [2018] considers the case of a Degree Corrected SBM model (DCSBM) as the network generative process, which has the following form:

$$P(A_{\mathcal{G}}|\boldsymbol{\lambda}, \boldsymbol{\kappa}, \boldsymbol{b}) = \prod_{i<j} \frac{\exp^{-\kappa_i \kappa_j \lambda_{b_i} \lambda_{b_j}} (\kappa_i \kappa_j \lambda_{b_i} \lambda_{b_j})^{A_{\mathcal{G}}(i,j)}}{A_{\mathcal{G}}(i,j)!}, \tag{2.8}$$

where $\kappa_i \kappa_j \lambda_{b_i} \lambda_{b_j}$ denotes the rate of a Poisson process, with $\lambda_{b_j} \lambda_{b_j}$ controlling the number of edges between two groups and κ_i representing the expected degree of node i. An advantage of using the DCSBM over the general SBM, is that the DCSBM can model networks with right-skewed degree distributions, which is a feature encountered in the majority of real-world networks.

For the calculation of the posterior (2.7), $P(A_{\mathcal{G}})$ is needed which can be derived by

(2.8) as follows:

$$P(A_{\mathcal{G}}) = \sum_{b} P(A_{\mathcal{G}}|\boldsymbol{b})P(\boldsymbol{b}), \tag{2.9}$$

with

$$P(A_{\mathcal{G}}|\boldsymbol{b}) = \int P(A_{\mathcal{G}}|\boldsymbol{\lambda}, \boldsymbol{\kappa}, \boldsymbol{b})P(\boldsymbol{\kappa}|\boldsymbol{b})P(\boldsymbol{\lambda}|\boldsymbol{b})d\kappa d\lambda. \tag{2.10}$$

However, the sum over $\boldsymbol{b}$ is intractable, thus Peixoto [2018] obtains the joint posterior of the true network $A_{\mathcal{G}}$ and block membership $\boldsymbol{b}$

$$P(A_{\mathcal{G}}, \boldsymbol{b}|D) = \frac{P(D|A_{\mathcal{G}})P(A_{\mathcal{G}}|\boldsymbol{b})P(\boldsymbol{b})}{P(D)}, \tag{2.11}$$

from which draws can be obtained using an MCMC algorithm with a Metropolis-Hastings step. By marginalising out the block membership $\boldsymbol{b}$ from the conditional posterior (2.11), Peixoto [2018] finally obtains the posterior distribution of the true network $A_{\mathcal{G}}$,

$$P(A_{\mathcal{G}}|D) = \sum_{b} P(A_{\mathcal{G}}, \boldsymbol{b}|D).$$

What remains is to also define a measurement error model $P(D|A_{\mathcal{G}})$ that describes the measurement error in the observed data. Equivalently to Le et al. [2018] and Newman [2018b], Peixoto [2018] defines the measurement error process as the process by which edges in the network data are observed with some false positive p and false negative q probabilities. Thus, the probability of observing x_{ij} edges between a pair of nodes (i, j) given n_{ij} measurements is

$$P(x_{ij}|n_{ij}, A_{\mathcal{G}}(i,j), p, q) = \binom{n_{ij}}{x_{ij}} \left[(1-q)^{x_{ij}} q^{n_{ij}-x_{ij}}\right]^{A_{\mathcal{G}}(i,j)} \left[p^{x_{ij}}(1-p)^{n_{ij}-x_{ij}}\right]^{1-A_{\mathcal{G}}(i,j)}.$$

The above expression is further simplified by assuming non-informative Beta priors for the false positive and false negative probabilities p, q as follows:

$$P(\boldsymbol{x}|\boldsymbol{n}, A) = \left[\prod_{i<j} \binom{n_{ij}}{x_{ij}}\right] \binom{E}{T}^{-1} \frac{1}{E+1} \binom{M-E}{X-T}^{-1} \frac{1}{M-E+1},$$

where

$$M = \sum_{i<j} n_{ij}, \qquad X = \sum_{i<j} x_{ij},$$

$$E = \sum_{i<j} n_{ij} A_{\mathcal{G}}(i,j), \qquad X = \sum_{i<j} x_{ij} A_{\mathcal{G}}(i,j).$$

Consequently, although Peixoto [2018] uses an MCMC scheme to infer the true underlying network and its block membership, he does not infer the parameters of his model simultaneously in the same MCMC algorithm. Instead, he obtains posterior estimates of the parameters p, q of his model by marginalising their posterior distributions conditioned on $A_{\mathcal{G}}$, with respect to the posterior samples of $A_{\mathcal{G}}$ obtained from the algorithm.

In Chapter 3, we provide an extension of the Measurement Error model presented in Le et al. [2018] through the formulation of a mixture model, that allows us to account for the heterogeneity in a network population in the following two ways: (i) through the detection of multiple clusters of networks in a population, and (ii) through the detection of an outlier cluster of networks in a population. In contrast to the aforementioned studies, we further obtain an MCMC scheme that simultaneously infers all the parameters of our model by drawing samples from their posterior distributions. The benefit of this formulation is that it allows the interpretation of the uncertainty contained in all the model parameters.

In this Chapter we provided a review of the three alternative modelling frameworks considered in the literature, that aim to describe the probabilistic mechanism that generates a network population. However, most of these studies do not account for the potential variability of the network data in a network population. In the recent studies of Mukherjee et al. [2017], Diquigiovanni and Scarpa [2019] and Signorelli and Wit [2020], the authors develop approaches that address this limitation assuming the presence of underlying clusters of networks in a population. In Chapter 3, we provide an extensive review of the studies that consider the topic of clustering multiple network data, along with the limitations that they entail.

Chapter 3

Bayesian model-based clustering for multiple network data

3.1 Introduction

In this chapter we present a mixture model for identifying clusters of networks in a network population, with respect to similarities detected in the connectivity patterns of the networks' nodes. Specifically, we consider the case when the networks in the population share the same set of n nodes and build a mixture model with a predefined number of clusters C. The mixture components of our mixture model are measurement error models, building on the study of Le et al. [2018]. This formulation allows us not only to identify clusters in multiple network data sets, but also to characterise them with respect to a parameterisation.

The availability of multiple network data has risen substantially in recent years due to the advancement of technological means that record this type of data (White et al. [1986], Fields and Song [1989]). This has inspired many researchers to seek for statistical models that most accurately describe the probabilistic mechanism that generates a network population. Despite the growing research interest on modelling multiple network data, only few studies developed to date consider the heterogeneity that can exist in a network population.

Notably, Mukherjee et al. [2017] were the first to consider the problem of clustering multiple network data, under two different settings, (a) when the networks in the population share the same set of nodes, (b) and when the networks in the population do not share the same set of nodes. For case (a), which is relevant to the assumption made in our work, the authors obtain a mixture model of graphons and implement a spectral clustering algorithm to infer the membership allocation of each network observation.

An application driven study on clustering multiple network data is introduced by Diquigiovanni and Scarpa [2019] who aim to cluster a population of networks where each network observation represents the playing style of a football team at a specific match, for a set of different football teams and matches throughout a season. The clustering approach seen in this study involves the specification of an ad hoc measure of similarity between networks, and the implementation of an agglomerative method for clustering the networks according to their similarities.

To the best of our knowledge, the third and last study that examines the problem of clustering network populations is that of Signorelli and Wit [2020]. In this study, the authors deal with the problem of clustering using a mixture model whose components can be any statistical network model, under the restriction that it can be specified as a Generalised Linear Model (GLM). For estimating the parameters of their model, they implement an Expectation Maximization (EM) algorithm for a predefined number of clusters. To determine the network model for their mixture, the authors propose the initial use of a mixture of saturated network models to reveal information about the structure of the data at hand. The saturated network model assumes that each edge in each network in the population is generated with some unique, unconstrained probability.

Another group of studies that accounts for the heterogeneity in a set of network observations, are the studies that perform the task of network classification. Some of these studies consider either specific network summary measures (Prasad et al. [2015]), or vectorise only the important entries (edges) of the adjacency matrix (Richiardi et al. [2011], Zhang et al. [2012]) to classify networks, thus they ignore the overall networks' structure. In contrast to these studies, Relión et al. [2019] perform prediction of the class membership of networks using a linear classifier with predictors the adjacency matrices of the networks. Their approach accounts for the networks' structure by using a penalty to select important nodes and edges.

The key limitations found in the aforementioned studies are the following:

L1 In the studies of Mukherjee et al. [2017], Diquigiovanni and Scarpa [2019] and Relión et al. [2019], the methods proposed are non model-based. Mukherjee et al.

[2017] and Diquigiovanni and Scarpa [2019] propose algorithms that detect underlying network clusters in the data and Relión et al. [2019] predict the class membership of the networks, while in all three studies the groups of networks identified cannot be interpreted with respect to some parametric representation. Interpretability of the different groups of networks in a population is crucial in many applications in order to infer group specific properties and differences.

L2 Even though Signorelli and Wit [2020] provide a model-based approach for clustering multiple network data, the use of already existing network models as the mixture components implies that only specific characteristics of the networks can be inferred, depending on what these model assumptions allow. It would be ideal to have a framework that is flexible enough to incorporate different modelling assumptions as deemed appropriate to application allowing the most scientifically relevant inferences to be made from the data.

L3 Signorelli and Wit [2020] propose to initially obtain a mixture of saturated network models, thus resulting in an overly complex model with a large number of parameters to estimate. This can substantially increase the computational time needed for the EM algorithm to converge as well as increasing the potential for non-convergence due to having to explore a very high dimensional parameter space.

L4 The supervised approach of Relión et al. [2019] requires a training data set to predict the class of a set of network observations. In this regard, the class labels of the networks in the training data set should be pre-specified, which can be restrictive for some network applications for which we do not have a priori information about the networks. Thus, the proposed supervised approach can predict class labels with respect to a specific characterisation of the networks.

In our study we aim to tackle these limitations by providing a model based approach for detecting clusters of networks in a network population, and interpreting the clusters with respect to a parameterisation. Inspired by the approach of Le et al. [2018], we adopt a measurement error formulation, assuming networks lying within each of the C clusters are noisy realisations of a true underlying cluster representative. The attractive feature of this approach is that it decouples the statistical model for the network data from the underlying cluster specific network properties. We are thus able to provide a flexible model based approach for detecting clusters of networks in a network population, as well as interpreting these clusters with respect to our model parameterisation. Our framework is also flexible enough to incorporate, and thus exploit, any underlying assumptions about the structure of the networks within the clusters that are of scientific interest or otherwise supported by the data.

Le et al. [2018] model a network population assuming the existence of a true underlying adjacency matrix, resulting from multiple noisy network-valued observations. The inferential framework built in their study, consists of two steps. First, they use a Spectral Clustering algorithm to infer the community structure formed by the nodes of the true underlying network, and second they implement an EM algorithm to estimate the model parameters. An evident limitation of their inferential framework is that their algorithm does not simultaneously update the parameters of their model and the block membership of the true network's nodes, as this would require the development of new techniques. In addition, the assumption of a sole true underlying adjacency matrix is quite restrictive, especially for large data sets of networks.

The contributions of the work presented in this chapter are the following. First, we extend the work by Le et al. [2018] through the formulation of a mixture model with components the measurement error models. Second, we provide a Bayesian framework for inferring the parameters of the mixture of measurement error models that allows to simultaneously infer all the model parameters, along with the block membership of the nodes. Third, our work adds to the literature on clustering multiple network data as we introduce a model-based approach that is not restricted to describing only specific network characteristics. Fourth, through our model-based approach we are able to interpret the clusters with respect to a network representative and a variability of the network data from that representative, encoded by the false positive and false negative probabilities. Fifth, the framework can also be formulated to detect outlying network observations when interest is in identifying observations that do not follow the distribution of the majority of the network data.

The remainder of this chapter is organised as follows. In Section 3.2, we provide the background to our modelling framework. In Section 3.3, we introduce our proposed Bayesian formulation of a mixture of measurement error models for network data, along with the MCMC scheme to make inferences. In Section 3.4, we present simulation studies to assess the performance of our method for various network sizes and sample sizes. In Section 3.5, we analyse two different multiple network data examples to illustrate the broad applicability of our methods. Lastly, in Section 3.6, we give some concluding remarks.

3.2 Background

In this Section we discuss the relevant background to our modelling framework for clustering network populations introduced in Section 3.3. Specifically, in Section 3.2.1 we provide a review of the measurement error model of Le et al. [2018] which has also been discussed in Chapter 2, and in Section 3.2.2 we provide a detailed review of the studies focusing on the topic of clustering network populations and discuss their limitations.

3.2.1 Measurement Error Model

A natural assumption arising from the objective of modelling a network population, is the assumption of a measurement error that occurs during the construction of the network data. Under this hypothesis, the researcher assumes that the observed network data result from measuring the edges of a true underlying network, which leads to recording some erroneous edges due to the existence of an underlying measurement error process. Le et al. [2018] were the first to introduce this approach for modelling populations of networks, which has inspired the study presented in this chapter.

Let $\mathcal{G}_1, \cdots, \mathcal{G}_N$ denote a population of networks, which share the same vertex set $|V| = n$. Le et al. [2018] assume that the information contained in the network population can be summarised by a representative network $\mathcal{G}^*$ and a measurement error process that does not allow us to accurately observe the representative network. Specifically, the authors assume a false positive probability P_{ij} of observing an edge between nodes i, j in the k^{th} network observation $\mathcal{G}_k$, given that there is no edge between the same two nodes in the representative network $\mathcal{G}^*$; and respectively, a false negative probability Q_{ij} of not observing an edge for the nodes i, j in the k^{th} network observation $\mathcal{G}_k$, while there is an edge for the same two nodes in the representative network $\mathcal{G}^*$. Thus, the entries of the matrices P, Q, are the false positive and false negative probabilities of observing or not and edge between two nodes in the data.

The mathematical formulation of the above set up is the following. Let $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$ denote the adjacency matrices for the network population, and $A_{\mathcal{G}^*}$ the adjacency matrix of the representative network. Thence, the false positive and false negative probabilities P_{ij}, Q_{ij} can be described as follows,

$$\text{if } A_{\mathcal{G}^*}(i,j) = 1, \text{ then } A_{\mathcal{G}_k}(i,j) = \begin{cases} 1, & \text{with prob } 1 - Q_{ij} \\ \\ 0, & \text{with prob } Q_{ij} \end{cases} ;$$

$$\text{if } A_{\mathcal{G}^*}(i,j) = 0, \text{ then } A_{\mathcal{G}_k}(i,j) = \begin{cases} 1, & \text{with prob } P_{ij} \\ \\ 0, & \text{with prob } 1 - P_{ij} \end{cases} . \tag{3.1}$$

From (3.1), it follows that the probability of the occurrence or non-occurrence of an edge between nodes i, j in the k^{th} network observation is,

$$P(A_{\mathcal{G}_k}(i,j) | A_{\mathcal{G}^*}(i,j) = 1, P_{ij}, Q_{ij}) = (1 - Q_{ij})^{A_{\mathcal{G}_k}(i,j)} \cdot Q_{ij}^{1 - A_{\mathcal{G}_k}(i,j)}, \text{ if } A_{\mathcal{G}^*}(i,j) = 1;$$

$$P(A_{\mathcal{G}_k}(i,j) | A_{\mathcal{G}^*}(i,j) = 0, P_{ij}, Q_{ij}) = P_{ij}^{A_{\mathcal{G}_k}(i,j)} \cdot (1 - P_{ij})^{1 - A_{\mathcal{G}_k}(i,j)}, \text{ if } A_{\mathcal{G}^*}(i,j) = 0.$$

In the study of Le et al. [2018], the adjacency matrix of the representative network $A_{\mathcal{G}^*}$ is treated as a latent variable, while the false positive and false negative probabilities P_{ij} and Q_{ij} are the model parameters. Thus, the likelihood of the representative network $A_{\mathcal{G}^*}$ given the network data $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$, as seen in Le et al. [2018], is

$$\mathcal{L}(A_{\mathcal{G}^*}; A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}) = \prod_{k=1}^{N} \prod_{(i,j):i<j} [(1 - Q_{ij})^{A_{\mathcal{G}_k}(i,j)} \cdot Q_{ij}^{1 - A_{\mathcal{G}_k}(i,j)} \cdot W_{ij}]^{A_{\mathcal{G}^*}(i,j)} \cdot$$
$$[P_{ij}^{A_{\mathcal{G}_k}(i,j)} \cdot (1 - P_{ij})^{1 - A_{\mathcal{G}_k}(i,j)} \cdot (1 - W_{ij})]^{1 - A_{\mathcal{G}^*}(i,j)}.$$

where $W_{ij} = \mathbb{E} A_{\mathcal{G}^*}(i,j)$, represents the probability of observing an edge between nodes i, j in the representative network. Le et al. [2018] further assume that the nodes of the underlying true network, form communities that can be described by an SBM, and respectively the corresponding block structure is assumed to be shared among the matrices P, Q and W.

The inference of the model parameters and the latent variable is conducted in two stages. First, a Spectral Clustering algorithm is applied to reveal the underlying block membership of the representative's nodes, and second, an EM algorithm is implemented to estimate the model parameters. While this formulation has some appealing features it would be ideal to have a coherent modelling framework that can jointly infer block membership of the representative's nodes together with the parameters characterising the distribution of the network data. In addition, using an EM algorithm to estimate model parameters means that measures of uncertainty such as standard errors for the noise and the SBM parameters, will need to rely on asymptotic approximations that may not be valid in many applications, particularly when involving small samples sizes.

In our work, we extend the model seen in Le et al. [2018] to a mixture of error measurement models, and formulate a Bayesian inferential framework that allows us to jointly infer the parameters of the measurement error model as well as those characterising the underlying network representatives corresponding to each cluster. In addition, the Bayesian formulation is flexible enough to accommodate diverse modelling assumptions for the network representatives.

3.2.2 Clustering multiple network data

Due to the continuous advancements occurring in the field of technology, there is a growing availability of large data sets with network-valued observations. That has triggered the development of new statistical techniques for modelling multiple network data, as seen in Chapter 2. However, most of the studies for modelling multiple network data do not consider the heterogeneity that can potentially exist in a large sample of network observations.

Mukherjee et al. [2017] address this limitation assuming the existence of underlying clusters of networks within a network population, and provide the framework for inferring them under two different conditions:

1. When the network observations share the same set of nodes.

2. When the network observations do not share a common set of nodes.

Under condition (i), the authors assume that a network population is generated by a mixture model of graphons f. A graphon $f : [0,1]^2 \mapsto [0,1]$ is a non negative, bounded, measurable and symmetric function that models the probability of observing an edge between two nodes i, j in a network. Notably, each of the i, j nodes is given a Uniform$(0,1)$ random value ξ_i, ξ_j, and the graphon f determines the probability of a connection to occur between the two nodes as $P_{ij} = f(\xi_i, \xi_j)$.

Thus, the probability of observing a network $A_{\mathcal{G}_k}$ under the graphon mixture model for a fixed number of clusters C is

$$P(A_{\mathcal{G}_k}) = \sum_{c=1}^{C} q_c P_{\Pi_c}(A_{\mathcal{G}_k}), \text{ with } P_{\Pi_c}(A_{\mathcal{G}_k}) = \prod_{u<v} \Pi_{c,uv}^{A_{\mathcal{G}_k}(u,v)} (1 - \Pi_{c,uv})^{1-A_{\mathcal{G}_k}(u,v)}$$

where $q_1, \cdots, q_C$ represent the weights for each mixture component, and $\Pi_1, \cdots, \Pi_C$ denote the edge probability matrices generated by the graphons $f_1, \cdots, f_C$.

Under this model, the authors seek to infer the underlying clusters of networks in a network population $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$ by first estimating for each observational unit $A_{\mathcal{G}_i}$ its link probability matrix Π_i, using already existing methods for graphon estimation. For the estimates of the link probability matrices $\hat{\Pi}_1, \cdots, \hat{\Pi}_C$ obtained, they compute an $N \times N$ distance matrix $\hat{D}$ with $\hat{D}_{ij} = \|\hat{\Pi}_i - \hat{\Pi}_j\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. Finally, the apply a spectral clustering algorithm on the distance matrix estimate $\hat{D}$, in order to detect the cluster membership of the N network observations.

For the case when the network observations do not share a common set of nodes (condition (ii)), the authors adopt the idea of constructing a vector that encodes features of each network observation $A_{\mathcal{G}_i}$, in order to cluster the networks according to their similarities with regard to their feature vectors. Specifically, they consider different network summary statistics which can be calculated by obtaining the trace of powers of the adjacency matrix,

$$m_k(A_{\mathcal{G}_i}) = \text{trace}(A_{\mathcal{G}_i}/n)^k \text{ with } k \text{ integer,}$$

and are called graph moments by the authors. Thus, different powers k give different types of summary statistics (e.g. total number of edges, triangle counts, directed circuits). The advantage of this representation over the subgraph counts is that the former is more computationally efficient.

Therefore, the authors obtain a feature vector for each network observation $A_{\mathcal{G}_i}$, which has the following form

$$g_J(A_{\mathcal{G}_i}) := (\log m_1(A_{\mathcal{G}_i}), \cdots, \log m_J(A_{\mathcal{G}_i})) \in \mathbb{R}^J,$$

for some positive integer $J \geq 2$. Similarly to their approach under condition (i), the authors form an $N \times N$ distance matrix D by calculating the distance between the feature vectors of the networks $d(g_J(A_{\mathcal{G}_i}), g_J(A_{\mathcal{G}_k}))$. Lastly, they cluster the network data using a spectral clustering algorithm to the distance matrix estimate $\hat{D}$ obtained.

Under this framework, the authors achieve to cluster network populations when the networks share or do not share a common vertex set, however the k-means algorithm within the spectral clustering algorithm applied under their construction, is very sensitive to the initialisation of the centroids of the clusters which might lead to unsteady results. Specifically, Spectral clustering is very sensitive to the choice of the similarity graph and its parameters, as different similarity graph and parameter specifications can capture different type of similarities in the data, thus their choice needs to depend on the desired type of information that the researcher aims to capture (Von Luxburg [2007]).

An application oriented study on clustering multiple network data is introduced by Diquigiovanni and Scarpa [2019]. The authors of this study aim to cluster a network population in which each network observation represents the playing style of a football team at a specific match, for a set of different football teams and matches throughout a season. Specifically, the nodes of the networks represent different areas of the pitch, and the edges result from the movements of the ball among these areas; thus, the network population considered in this study shares the same set of nodes. In addition, the edges are directed and weighted, with each weight representing the number of times that the ball consecutively moved from one area of the pitch (node) to another.

The methodological approach presented in Diquigiovanni and Scarpa [2019] involves two stages. At the first stage, the authors pre-process the network data by normalising the weighted edges of the network population. Specifically, for each pair of nodes, they consider the minimum and maximum edge weight observed in the whole population of networks, and use min-max normalisation to normalise the weights corresponding to the specific pair of nodes in the observed network data. In this way, each normalised edge weight accounts for the different sizes of weights observed for the corresponding pair of nodes across the population of networks. The next step of that stage is to detect the communities formed by the nodes of each network observation using the Louvain method (Blondel et al. [2008]). A limitation arising from this method is that it cannot be applied on directed network data. Due to this restriction, the authors consider the undirected case for the network data, which can potentially lead to loss of information.

An issue arising from the normalisation of the edges is that the resulting weights of the edges might vary significantly between two networks, while having the same community structure. To overcome this problem, the authors consider a threshold s for the size of the edge weight to be considered in the following way,

$$w_k(i,j) = \begin{cases} u_k(i,j) & \text{if } u_k(i,j) > s \\ 0 & \text{if } u_k(i,j) \leq s \end{cases};$$

for $k \in \{1, \cdots, N\}$, where N corresponds to the size of the network population and $u_k(i,j)$ is the normalised edge weight between nodes i, j, for the k^{th} network observation.

The value of the threshold s is crucial in order to keep a balance between dealing with the issue arising from the normalisation, and avoiding any loss of information enclosed in the data. The authors propose to obtain a different threshold $s(i,j)$ for each pair of (i,j) nodes in the following way,

$$s(i,j) = q(\boldsymbol{u(i,j)}, \alpha),$$

where $\boldsymbol{u(i,j)}$ is a vector of all the normalised edges found in the N network data for the (i,j) pair of nodes, and $q(x, \alpha)$ is the quantile of order α for x.

At the second stage of Diquigiovanni and Scarpa [2019] method, the authors conduct a clustering process on the network data. To accomplish that, they first define an ad hoc measure of similarity to compare the community structures detected in the network data. Notably, the authors apply the Adjusted Rand Index (ARI) (Hubert and Arabie [1985]) on the community structures of the network data, resulting in a $N \times N$ distance matrix with elements the distances measured under the ARI between the N networks. Lastly, they feed the distance matrix to the UPGMA method (Sokal [1958]) which starts with the assumption that each network belongs to its own cluster, and gradually merges the networks until they all belong to a single cluster.

The identification of the number of clusters existing in the football network population is determined by the size of the difference of the maximum ARI index, between two successive steps of the method. Under this procedure, the authors detected 15 clusters in the network population. In order to characterise each cluster detected, the authors consider network summary statistics (pairs, triads) of the nodes allocated in the same community. The results obtained from the summary statistics, lead to the description of 15 different playing styles in the network population.

Two critical issues that need to be dealt with under the framework proposed by Diquigiovanni and Scarpa [2019] are the determination of the threshold in the pre-processing phase of their method, and the consideration of only undirected network data, which both imply a potentially high loss of information in the data.

A key limitation found in both studies of Mukherjee et al. [2017] and Diquigiovanni and Scarpa [2019] is that the clustering schemes proposed are non model-based. Thus, despite the fact that the algorithms proposed detect underlying clusters of networks in multiple network data sets, the clusters identified cannot be interpreted with respect to some parameterisation.

On the other hand, Signorelli and Wit [2020] tackle this limitation by proposing a model-based framework for clustering multiple network data according to their similarities, with respect to a parameterisation. Notably, the authors assume that a population of networks is generated by a mixture model, having as mixture components any statistical network model that can be specified as a Generalised Linear Model (GLM) or Generalised Mixed Model (GLMM).

Under this framework, the authors predefine the number of clusters C underlying in a network population, and denote by $z_k \in \{1, \cdots, C\}$ the unknown cluster membership of the k^{th} network observation. Thence, the likelihood of their model, given the latent membership $\boldsymbol{z}$ of the networks is

$$L(\boldsymbol{\theta}; A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}, \boldsymbol{z}) = \prod_{k=1}^{N} \pi_{z_k} f(A_{\mathcal{G}_k} | \theta_{z_k}),$$

where π_c denotes the probability of network to belong to cluster c, $f(\cdot)$ corresponds to the network model assumed, $\boldsymbol{\theta}$ denote the parameters of the network model, varying among the C clusters, and $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$ correspond to the adjacency matrices of the N network observations.

The choice of the statistical network model $f(\cdot)$ depends on the network characteristics that the researcher wants to capture, under the restriction that the corresponding model can be defined as a GLM or GLMM. For example, if the researcher wants to identify clusters in a network population according to similarities of the networks' degree distributions, the p_1 and p_2 network models are two candidate models that allow this information to be encoded. Specifically, p_1 and p_2 assume that a probability of observing an edge between two nodes, depends on the nodes' expected degrees.

The authors implement an Expectation-Maximization (EM) algorithm on the likelihood of their model, to estimate the model parameters $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_C)$ and the $N \times C$ probability matrix $\boldsymbol{P}$, whose p_{kc} entries denote the probability of the k^{th} network observation to belong to the c cluster. To determine the number of mixture components C, the authors formulate three model selection criteria, namely the Akaike (AIC), the Bayesian (BIC) and the Generalized (GIC) Information Criteria.

Notwithstanding that Signorelli and Wit [2020] provide a model-based approach for clustering multiple network data, their framework involves some underlying constraints. One limitation is the use of already existing network models as the mixture components of their model, which ultimately implies that only specific, global characteristics of the networks will be explained under their formulation. In addition, a large number of parameters can negatively affect the computational time needed for the EM algorithm to converge.

3.3 Mixture of Measurement Error models

In this section we detail our proposed modelling framework for clustering network populations, that tackles the limitations of the studies discussed in Section 3.2. We first describe the motivation and objectives of our modelling framework in Section 3.3.1, the Bayesian formulation of the measurement error model when there is only one cluster is presented in Section 3.3.2. Thence, we extend this to multiple clusters in Section 3.3.3, and describe how posterior samples can be obtained using MCMC in Section 3.3.4. Finally, we describe a special case of this formulation that can correspond to detecting outlying networks in the data in Section 3.3.5.

3.3.1 Motivation

We now introduce a motivating example that has driven the research objectives of our study. This example comes from the data collected by the Tacita mobile application, created by members of the Computing and Communications department at Lancaster university (Shaw et al. [2018]).

The Tacita application was created to serve as a mean of communication between a display and a viewer, in order for the viewer to be able to see content relevant to his interests. Specifically, the viewer can request what to see on the screen of the display, but also the display can detect when a user is in its proximity in order to show content aligned with his interests. Thus, the application records the consecutive displays visited by the users, along with the time visited and the type of content shown.

Consequently, the Tacita data set serves as an example of multiple network observations, as the movements of each individual can be represented by a network. Thus, for each individual we can obtain a network where nodes represent displays and edges represent the movements of the user among the displays. The questions arising from this data set are the following:

Q1 Can we detect different patterns among the users' movements?

Q2 Can we cluster the users according to their movements?

Q3 How informative can the clustering be for the users in our data?

The aforementioned research questions have motivated the development of a mixture model with mixture components the measurement error model, building upon the work by Le et al. [2018] presented in the previous section. Notably, the research objectives of the mixture model presented in this chapter are:

O1 **The development of a model-based approach for clustering multiple network data**

We aim to capture the multimodality in a network population by formulating a mixture of measurement error models. Under this construction, the cluster membership of the network data will be inferred according to similarities found in the connectivity patterns of the network data.

O2 **The interpretation of the clusters detected with respect to a parameterisation**

The use of the measurement error model as the mixture component of our mixture model not only detects potential clusters formed in the data, but also allows the characterisation of the clusters by a network representative, and a false positive and false negative probability that reflect information about the amount of noise in the data, in a given cluster.

O3 **The formulation of a flexible framework to model the heterogeneity in a network population**

Our framework allows the flexibility to answer diverse research questions with respect to the heterogeneity in a network population. Specifically the mixture model can be formulated to detect network observations that are different from a majority of networks in the sample.

O4 **The construction of a Bayesian framework that allows to simultaneously infer of all the model parameters**

We propose a novel Bayesian framework for the measurement error model that tackles the limitation of having a two-stage inferential scheme for the parameters of the model, as seen in Le et al. [2018]. Specifically, we introduce an MCMC scheme that infers the latent block membership of the nodes and the latent cluster membership of the networks, along with the posterior distribution of the model parameters.

O5 **Allowing varying block structure among the representatives**

Under our formulation, the nodes of each network representative can form a distinct community structure. Our MCMC scheme reveals the unique block structures formed by the nodes of each representative, allowing more information to be decoded about the corresponding cluster.

3.3.2 Bayesian set up for measurement error model with SBM structure for the representatives

In this section, we introduce a Bayesian set up for the measurement error model presented in Le et al. [2018]. To begin with, we assume the presence of a latent representative network with adjacency matrix denoted by $A_{\mathcal{G}^*}$, underlying the observed network data. In addition, we assume that the probability of observing a false positive or false negative edge between two nodes in a the network data, is independent of the pair of nodes considered. Thus, the false positive probability p and false negative probability q are scalars in our case. Under this specification, the probability mass function of the edge state between nodes (i, j) in network observation k, given $A_{\mathcal{G}^*}(i, j)$, p, q, is

$$P(A_{\mathcal{G}_k}(i,j)|A_{\mathcal{G}^*}(i,j),p,q) = [(1-q)^{A_{\mathcal{G}_k}(i,j)} \cdot q^{1-A_{\mathcal{G}_k}(i,j)}]^{A_{\mathcal{G}^*}(i,j)} \cdot \\ [p^{A_{\mathcal{G}_k}(i,j)} \cdot (1-p)^{1-A_{\mathcal{G}_k}(i,j)}]^{1-A_{\mathcal{G}^*}(i,j)}.$$

Hence, the conditional probability of observing a network population $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$ given $A_{\mathcal{G}^*}$, p, q is,

$$P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}|p,q,A_{\mathcal{G}^*}) = \prod_{k=1}^{N} \prod_{(i,j):i<j} P(A_{\mathcal{G}_k}(i,j)|A_{\mathcal{G}^*}(i,j),p,q) =$$

$$\prod_{k=1}^{N} \prod_{(i,j):i<j} ((1-q)^{A_{\mathcal{G}_k}(i,j)} \cdot q^{1-A_{\mathcal{G}_k}(i,j)})^{A_{\mathcal{G}^*}(i,j)} \cdot (p^{A_{\mathcal{G}_k}(i,j)} \cdot (1-p)^{1-A_{\mathcal{G}_k}(i,j)})^{1-A_{\mathcal{G}^*}(i,j)}.$$

An advantage of the measurement error formulation is that the model specification for the representative network $A_{\mathcal{G}^*}$ can vary depending on the type of information the statistician wants to capture for the data at hand. As the SBM model is a widely used model for networks with neat interpretation, and motivated by the work of Le et al. [2018], as a general approach we also assume an SBM structure for the representative $A_{\mathcal{G}^*}$. We note here, that this can be easily modified, e.g. reduced to a simpler model such as the Erdös-Rényi if supported by the data.

Under the SBM assumption, each node of the representative network belongs to an unobserved block $k \in \{1, \cdots, K\}$, and the probability of observing an edge between two nodes depends on their block membership. We represent the block membership of the n nodes by $\{b_i\}_{i=1}^n$, with $b_i \in \{1, \cdots, K\}$, and the probability of observing an edge between nodes (i,j) with $b_i = k,\ b_j = l$ by θ_{kl}. Under this specification, the hierarchical structure of the SBM model is as follows:

$$A_{\mathcal{G}^*}(i,j)|\boldsymbol{\theta}, \boldsymbol{b} \sim \text{Bernoulli}(\theta_{b_i b_j});$$
$$\theta_{kl} \sim \text{Beta}(\epsilon_{kl}, \zeta_{kl});$$
$$\boldsymbol{b}|\boldsymbol{w} \sim \text{Multinomial}(\boldsymbol{w});$$

where w_k represents the probability of a node to belong to block $k \in \{1, \cdots, K\}$. For the vector of probabilities $\boldsymbol{w} = \{w_1, \cdots, w_K\}$ we assume a symmetric Dirichlet prior distribution with hyperparameter $\boldsymbol{\chi}$,

$$\boldsymbol{w} \sim \text{Dirichlet}(\boldsymbol{\chi}).$$

Common choices for the hyperparameter $\boldsymbol{\chi}$ are 0.5 and 1.

Thus the hierarchical structure of the model is,

$$\prod_{k=1}^{N} \prod_{(i,j):i<j} P(A_{\mathcal{G}_k}(i,j)|A_{\mathcal{G}^*}(i,j), p, q) P(A_{\mathcal{G}^*(i,j)}|\boldsymbol{\theta}, \boldsymbol{b})$$

where

$$P(A_{\mathcal{G}^*}(i,j)|\boldsymbol{\theta}, \boldsymbol{b}) = \theta_{b_i b_j}^{A_{\mathcal{G}^*}(i,j)} (1 - \theta_{b_i b_j})^{1 - A_{\mathcal{G}^*}(i,j)}.$$

We further specify a Beta prior distribution for both the false positive p and false negative q probabilities,

$$p \sim \text{Beta}(\alpha_0, \beta_0),\ q \sim \text{Beta}(\gamma_0, \delta_0),$$

for the sake of convenience. A common choice of the hyperparameters for the Beta priors stated in this section is 0.5 corresponding to Jeffreys prior.

As stated previously the Bayesian formulation allows us to jointly infer the SBM parameters in together with the rest of the model parameters when making posterior inferences.

3.3.3 Mixture of measurement error models

We further extend the measurement error model to a mixture of measurement error models, with a predefined number of mixture components C, in order to provide a model-based approach for identifying clusters of networks in a network population $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$. Under this specification, we assume that each cluster c of networks, is described by a unique network representative $A_{\mathcal{G}_c^*}$, a false positive probability p_c and a false negative probability q_c, where $c \in \{1, \dots, C\}$. In this section, we present the Bayesian framework for the mixture of measurement error models, with each cluster-specific representative network being expressed by an SBM. We note here, that the block structure of each representative is allowed to vary.

Let $\boldsymbol{z} = (z_1, \cdots, z_N) \in \{1, \cdots, C\}$ be the latent variables representing the cluster membership of the network data $A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}$. Under this construction, the conditional probability of the data, given the $\boldsymbol{z}$ latent variable, takes the form

$$P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} | \{p_c, q_c, A_{\mathcal{G}_c^*}\}_{c=1}^{C}, z_1, \cdots, z_N) =$$
$$= \prod_{k=1}^{N} \Big(\prod_{(i,j):i<j} \Big((1-q_{z_k})^{A_{\mathcal{G}_k}(i,j)} q_{z_k}^{1-A_{\mathcal{G}_k}(i,j)} \Big)^{A_{\mathcal{G}_{z_k}^*}(i,j)} \cdot \Big(p_{z_k}^{A_{\mathcal{G}_k}(i,j)} (1-p_{z_k})^{1-A_{\mathcal{G}_k}(i,j)} \Big)^{1-A_{\mathcal{G}_{z_k}^*}(i,j)} \Big).$$

Respectively, taking the log of the conditional probability of the data reduces to,

$$\log P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} | \{p_c, q_c, A_{\mathcal{G}_c^*}\}_{c=1}^{C}, z_1, \cdots, z_N)) \propto$$
$$\propto \sum_{k=1}^{N} \Bigg(\sum_{(i,j):i<j} \Big(A_{\mathcal{G}_{z_k}^*}(i,j) \Big(\log(1-q_{z_k}) A_{\mathcal{G}_k}(i,j) + \log(q_{z_k})(1 - A_{\mathcal{G}_k}(i,j)) \Big) +$$
$$+ (1 - A_{\mathcal{G}_{z_k}^*}(i,j)) \Big(\log(p_{z_k}) A_{\mathcal{G}_k}(i,j) + \log(1-p_{z_k})(1 - A_{\mathcal{G}_k}(i,j)) \Big) \Big) \Bigg) =$$
$$= \sum_{k=1}^{N} \Bigg(\sum_{(i,j):i<j} \Big(A_{\mathcal{G}_{z_k}^*}(i,j) \Big(A_{\mathcal{G}_k}(i,j) \log \frac{(1-q_{z_k})(1-p_{z_k})}{q_{z_k} p_{z_k}} + \log \frac{q_{z_k}}{1-p_{z_k}} \Big) +$$
$$+ \Big(A_{\mathcal{G}_k}(i,j) \log \frac{p_{z_k}}{1-p_{z_k}} + \log(1-p_{z_k}) \Big) \Big) \Bigg).$$

We assume that the cluster labels $z_1, \cdots, z_N$ follow a Multinomial distribution with parameter $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_C)$, where τ_c represents the probability that a network observation belongs to class c, and $\sum_{c=1}^{C} \tau_c = 1$. We assume a symmetric Dirichlet prior distribution for the vector of probabilities $\boldsymbol{\tau}$ which has the advantage of being conditionally conjugate with the conditional probability of $\boldsymbol{z}$. As commented in Section 3.3.2, common choices of the Dirichlet hyperparameter are 0.5 or 1.

In Section 3.3.4 we present details on how posterior inferences can be made using MCMC.

3.3.4 MCMC scheme for mixture model

With the modelling framework described above we are able to draw samples from the joint posterior distribution of the parameters using MCMC. The joint posterior distribution is known up to a normalising constant, specifically

$$\begin{aligned} &P(\boldsymbol{A_{\mathcal{G}^*}}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta} | A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) \\ &\propto P(A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N} | \boldsymbol{A_{\mathcal{G}^*}}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}) \cdot P(\boldsymbol{A_{\mathcal{G}^*}} | \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}) \cdot P(\boldsymbol{p} | \boldsymbol{\alpha_0}, \boldsymbol{\beta_0}) \\ &\cdot P(\boldsymbol{q} | \boldsymbol{\gamma_0}, \boldsymbol{\delta_0}) \cdot P(\boldsymbol{z} | \boldsymbol{\tau}) \cdot P(\boldsymbol{\tau} | \boldsymbol{\psi}) \cdot P(\boldsymbol{\theta} | \boldsymbol{\epsilon}, \boldsymbol{\zeta}) \cdot P(\boldsymbol{b} | \boldsymbol{w}) \cdot P(\boldsymbol{w} | \boldsymbol{\chi}), \end{aligned} \tag{3.2}$$

where $P(A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N} | \boldsymbol{A_{\mathcal{G}^*}}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z})$ is the conditional probability of the network data, $P(\boldsymbol{A_{\mathcal{G}^*}} | \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta})$ is the conditional probability of the latent variable $\boldsymbol{A_{\mathcal{G}^*}}$ and the rest of the components of the right hand side of the expression are the prior distributions on the model parameters, as defined in Sections (3.3.2) and (3.3.3).

To obtain posterior samples from the joint posterior in (3.2), we note that many of the full conditional distributions of the unknown quantities (parameters/latent data) are available in closed form, and when these are not available these can be approximated using Metropolis-Hastings. As a result we obtain posterior inferences through a component wise MCMC sampler, also known as a Metropolis-Hastings-within-Gibbs sampler. This closely follows a Gibbs sampler, where all parameters and latent data are updated from their full conditional distributions except for the full conditionals of $\{A_{\mathcal{G}_c^*}, p_c, q_c\}_{c=1}^{C}$, which are approximated using Metropolis-Hastings proposal distributions.

In the Metropolis-Hastings step, we use a mixture of kernels for updating the parameters of the measurement error model $\{A_{\mathcal{G}_c^*}, p_c, q_c\}_{c=1}^{C}$, in analogy to the MCMC scheme seen in Lunagómez et al. [2021]. Specifically, we update the adjacency matrix $A_{\mathcal{G}_c^*}$ of the network representative of cluster c, using either of the following two proposals with some fixed probability, in every iteration of the MCMC:

(I) We perturb the edges of the current network representative $A_{\mathcal{G}_c^*}^{(curr)}$ of cluster c in the following way:

$$A_{\mathcal{G}_c^*}^{(prop)}(i,j) = \begin{cases} 1 - A_{\mathcal{G}_c^*}^{(curr)}(i,j), \text{ with probability } \omega \\ A_{\mathcal{G}_c^*}^{(curr)}(i,j), \text{ with probability } 1-\omega \end{cases}.$$

(II) We propose a new network representative $A_{\mathcal{G}_c^*}^{(prop)}$ for cluster c, with each edge of the proposed representative $A_{\mathcal{G}_c^*}^{(prop)}(i,j)$ being drawn independently, from a Bernoulli distribution with parameter $\frac{1}{N}\sum_{k=1}^{N} A_{\mathcal{G}_k}(i,j)$.

For case (I), we accept the proposed network representative $A_{\mathcal{G}_c^*}^{(prop)}$ with probability

$$\min\left\{1, \frac{P(A_{\mathcal{G}_1},\cdots,A_{\mathcal{G}_N}|A_{\mathcal{G}_c^*}^{(prop)}, p_c^{(curr)}, q_c^{(curr)}, \boldsymbol{z}^{(curr)})P(A_{\mathcal{G}_c^*}^{(prop)}|\boldsymbol{b_c}^{(curr)}, \boldsymbol{\theta_c}^{(curr)})}{P(A_{\mathcal{G}_1},\cdots,A_{\mathcal{G}_N}|A_{\mathcal{G}_c^*}^{(curr)}, p_c^{(curr)}, q_c^{(curr)}, \boldsymbol{z}^{(curr)})P(A_{\mathcal{G}_c^*}^{(curr)}|\boldsymbol{b_c}^{(curr)}, \boldsymbol{\theta_c}^{(curr)})}\right\}, \tag{3.3}$$

where $P(A_{\mathcal{G}_c^*}^{(\cdot)}|\boldsymbol{b_c}, \boldsymbol{\theta_c})$ is the SBM model assumed for the representative, as seen in section (3.3.2). We note that the proposal distribution under case (I) is symmetric, thus, it cancels out from the Metropolis ratio in expression (3.3).

Under case (II), we accept the proposed network representative $A_{\mathcal{G}_c^*}^{(prop)}$ with probability

$$\min\left\{1, \frac{P(A_{\mathcal{G}_1},\cdots,A_{\mathcal{G}_N}|A_{\mathcal{G}_c^*}^{(prop)}, p_c^{(curr)}, q_c^{(curr)}, \boldsymbol{z}^{(curr)})P(A_{\mathcal{G}_c^*}^{(prop)}|\boldsymbol{b_c}^{(curr)}, \boldsymbol{\theta_c}^{(curr)})}{P(A_{\mathcal{G}_1},\cdots,A_{\mathcal{G}_N}|A_{\mathcal{G}_c^*}^{(curr)}, p_c^{(curr)}, q_c^{(curr)}, \boldsymbol{z}^{(curr)})P(A_{\mathcal{G}_c^*}^{(curr)}|\boldsymbol{b_c}^{(curr)}, \boldsymbol{\theta_c}^{(curr)})} \cdot \frac{Q(A_{\mathcal{G}_c^*}^{(curr)}|A_{\mathcal{G}_c^*}^{(prop)})}{Q(A_{\mathcal{G}_c^*}^{(prop)}|A_{\mathcal{G}_c^*}^{(curr)})}\right\}, \tag{3.4}$$

where $Q(A_{\mathcal{G}_c^*}^{(\cdot)}|A_{\mathcal{G}_c^*}^{(\cdot)})$ corresponds to the proposal distribution, which is not cancelling out from the Metropolis ratio in expression (3.4).

To update the false positive probability p_c of cluster c, we use a mixture of random walk proposals indexed by l (Lunagómez et al. [2021]), constrained to lie in the interval (0,0.5) for identifiability reasons, in the following way:

- Draw $v \sim \text{Unif}(-u_l, u_l)$.

- Calculate the candidate proposal value $y = p_c^{(curr)} + v$.

- Propose a new value for p_c as follows,

$$p_c^{(prop)} = \begin{cases} y, & \text{if } 0 < y < 0.5; \\ -y, & \text{if } y < 0; \\ 1 - y, & \text{if } y > 0.5. \end{cases}$$

The mixture is over $\{u_1, \ldots, u_L\}$. Thus, we perturb the current state of the false positive probability $p_c^{(curr)}$ using various sizes of u_l, each imposing a less or more drastic change on $p_c^{(curr)}$. We accept the proposed value $p_c^{(prop)}$ with probability

$$\min\left\{1, \frac{P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} | A_{\mathcal{G}_c^*}^{(curr)}, p_c^{(prop)}, q_c^{(curr)}, \boldsymbol{z}^{(curr)}) P(p_c^{(prop)} | \alpha_{0,c}, \beta_{0,c})}{P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} | A_{\mathcal{G}_c^*}^{(curr)}, p_c^{(curr)}, q_c^{(curr)}, \boldsymbol{z}^{(curr)}) P(p_c^{(curr)} | \alpha_{0,c}, \beta_{0,c})}\right\}, \qquad (3.5)$$

where $P(p_c^{(\cdot)} | \alpha_{0,c}, \beta_{0,c})$ is the Beta prior with hyperparameters $\alpha_{0,c}, \beta_{0,c}$, as stated in section (3.3.2). We note again that the proposal distribution for p_c is symmetric, thus it does not appear in the Metropolis ratio in expression (3.5). In exactly the same manner, we update the false negative probability q_c, for $c \in \{1, \ldots, C\}$.

The rest of the parameters are updated via Gibbs samplers, by drawing values from their full conditional posteriors. The full conditional posterior for $\boldsymbol{\tau}$ is given by

$$P(\boldsymbol{\tau} | \boldsymbol{A}_{\boldsymbol{\mathcal{G}}^*}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) = \text{Dirichlet}(\psi + \eta_1, \ldots, \psi + \eta_C). \qquad (3.6)$$

where $\eta_c = \sum_{j=1}^N \mathbb{1}_c(z_j)$, $c = 1, \ldots, C$, denotes the number of networks that belong to cluster c.

We draw the latent cluster-membership z_k for each network observation k from a Multinomial distribution with unnormalised probabilities specified in the following way:

$$\begin{aligned} P(z_k = c | \boldsymbol{\tau}, \boldsymbol{A}_{\boldsymbol{\mathcal{G}}^*}, \boldsymbol{p}, \boldsymbol{q}, A_{\mathcal{G}_k}) \propto P(A_{\mathcal{G}_k} | z_k = c, \boldsymbol{A}_{\boldsymbol{\mathcal{G}}^*}, \boldsymbol{p}, \boldsymbol{q}) \cdot P(z_k = c | \boldsymbol{\tau}) \\ = \tau_c \cdot \prod_{(i,j): i<j} \left((1 - q_c)^{A_{\mathcal{G}_k}(i,j)} q_c^{1 - A_{\mathcal{G}_k}(i,j)}\right)^{A_{\mathcal{G}_c^*}(i,j)} \cdot \left(p_c^{A_{\mathcal{G}_k}(i,j)} (1 - p_c)^{1 - A_{\mathcal{G}_k}(i,j)}\right)^{1 - A_{\mathcal{G}_c^*}(i,j)} \end{aligned} \qquad (3.7)$$

where $P(A_{\mathcal{G}_k}|z_k = c, \boldsymbol{A_{\mathcal{G}^*}}, \boldsymbol{p}, \boldsymbol{q})$ is the probability we observe network k given cluster membership $z_k = c$, described by a measurement error model. The normalised probabilities are obtained via Bayes Theorem.

The full conditional posterior for $\boldsymbol{w_c}$ is

$$P(\boldsymbol{w_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{\theta_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) = \text{Dirichlet}(\chi + h_1, \ldots, \chi + h_K).$$

where h_k denotes the number of the nodes that belong to block k.

The full conditional posterior for the vector of the block-specific probabilities of an edge occurrence for the network representative of cluster c, $\boldsymbol{\theta_c}$, is

$$P(\boldsymbol{\theta_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{w_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) = \text{Beta}(A_{\mathcal{G}_c^*}[kl] + \epsilon_{kl}, \zeta_{kl} + n_{c,kl} - A_{\mathcal{G}_c^*}[kl]). \tag{3.8}$$

where $A_{\mathcal{G}_c^*}[kl] = \sum_{(i,j):b_{c,i}=k, b_{c,j}=l} A_{\mathcal{G}_c^*}(i,j)$ represents the sum of the entries for the pairs of nodes of the network representative for cluster c that have block membership k, l respectively, and $n_{c,kl} = \sum_{(i,j):i \neq j} \mathbb{I}(b_{c,i} = k, b_{c,j} = l)$ represents the number of the pair of nodes of the representative for cluster c that have membership k, l respectively.

Similarly to the formulation obtained for updating the latent cluster-membership $\boldsymbol{z}$ of the network data, we obtain updates of the latent block-membership $\boldsymbol{b_c}$ for the nodes of the network representative of cluster c from a Multinomial distribution with unnormalised probabilities specified as follows:

$$\begin{aligned} P(b_{c,i} = k|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{\theta_c}, \boldsymbol{w_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) &\propto P(A_{\mathcal{G}_c^*}|\boldsymbol{w_c}, \boldsymbol{\theta_c}, b_{c,i} = k) \cdot P(b_{c,i} = k|\boldsymbol{w_c}) \\ &= w_{c,k} \cdot \prod_{j=1}^{n} \theta_{kb_{c,j}}^{A_{\mathcal{G}_c^*}(i,j)} (1 - \theta_{kb_{c,j}})^{1 - A_{\mathcal{G}_c^*}(i,j)}. \end{aligned} \tag{3.9}$$

where $P(A_{\mathcal{G}_c^*}|\boldsymbol{w_c}, \boldsymbol{\theta_c}, b_{c,i} = k)$ is the probability of observing the representative of cluster c, $A_{\mathcal{G}_c^*}$, described by an SBM, given its i^{th} node belongs to block k. Normalised probabilities are obtained by Bayes Theorem.

For the detailed derivation of the full conditional posterior distributions refer to Appendix A.1. In addition, the MCMC algorithm for clustering is sketched in Appendix

A.4.

3.3.5 Outlier network detection

In this section we present a modification of the mixture model presented in Section 3.3.3, that allows us to explore the heterogeneity in a population of networks under a different perspective. Notably, we modify our mixture model to detect a cluster of outlier networks that are different to the majority of the networks in the population. Under this formulation, we are able to answer different type of research questions for the data at hand, compared to the case of multiple cluster detection.

In contrast to the mixture model formulated for multiple cluster detection, the outlier cluster detection model assumes a single network representative for the whole population of networks. Under this set up, we assume that there are ultimately two clusters of networks formed within the population of networks, one cluster being the majority cluster, while the other cluster enclosing the outlier networks of the population. Thus, while the false positive and false negative probabilities remain component specific for each of the two clusters, the network representative is no more a component specific latent variable.

Similarly to the mixture model formulated in Section 3.3.3, we now specify the number of clusters to $C = 2$, and $\boldsymbol{z} = (z_1, \cdots, z_N) \in \{1, 2\}$ denotes the latent cluster membership of the network data $A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}$. Under the assumption of a single network representative $A_{\mathcal{G}^*}$, the likelihood of the data given the $\boldsymbol{z}$ latent variable takes the form

$$P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} | \{p_c, q_c\}_{c=1}^{C}, A_{\mathcal{G}^*}, z_1, \cdots, z_N) =$$
$$\prod_{k=1}^{N} \Big(\prod_{(i,j):i<j} \Big((1-q_{z_k})^{A_{\mathcal{G}_k}(i,j)} q_{z_k}^{1-A_{\mathcal{G}_k}(i,j)}\Big)^{A_{\mathcal{G}^*}(i,j)} \cdot \Big(p_{z_k}^{A_{\mathcal{G}_k}(i,j)}(1-p_{z_k})^{1-A_{\mathcal{G}_k}(i,j)}\Big)^{1-A_{\mathcal{G}^*}(i,j)} \Big).$$

As in the case of the mixture model for multiple cluster detection, the model specification for the representative network can vary depending on the type of information we want to capture for the data at hand. A common choice is to consider an SBM structure again for the representative. Due to having now a single representative, the SBM model parameters are no more component specific to the cluster.

To sample from the joint posterior of this model, we develop a Metropolis-Hastings-within-Gibbs MCMC scheme, as presented in Section 3.3.4. The full conditional posterior distributions are obtained as seen in the expressions of Section 3.3.4, with sole difference that the parameters/latent variables involving the representative $A_{\mathcal{G}^*}$, namely the block-depending probability of an edge $\boldsymbol{\theta}$, the probability of a node to belong to a block $\boldsymbol{w}$ and the block membership of the nodes $\boldsymbol{b}$, are no more component specific, i.e. not indexed by cluster c.

3.4 Simulation study for Mixture Error Measurement Model

In this Section, we perform simulation studies to assess the performance of our algorithm in inferring the model parameters/latent variables and clustering network data. In Section 3.4.1, we explore the performance of our algorithm for moderate network sizes and various noise levels and SBM models, and in Section 3.4.2, we investigate the algorithm performance for various network and sample sizes.

3.4.1 Moderate network sizes

In this simulation study, we explore and evaluate the performance of our model in inferring the model parameters for network populations with a moderate size of nodes, under different parameterisations of the model. Specifically, we consider the case of networks with $n = 21$ number of nodes, and a network population size of $N = 180$ networks. In addition, we consider $C = 3$ number of clusters where each network belongs and $B = 2$ number of blocks where the nodes of each representative network belong, and allow varying sizes of the parameters of the model, in order to explore performance. To simulate the network population, we first need to simulate the representative network of each cluster. We generate representatives of the clusters under two different SBM structures with the following parameters,

- **SBM 1:** $(w_1^{(1)}, w_2^{(1)}) = (0.5, 0.5)$, $(w_1^{(2)}, w_2^{(2)}) = (0.5, 0.5)$, $(w_1^{(3)}, w_2^{(3)}) = (0.5, 0.5)$, $(\theta_{11}^{(1)}, \theta_{12}^{(1)}, \theta_{22}^{(1)}) = (0.8, 0.2, 0.8)$, $(\theta_{11}^{(2)}, \theta_{12}^{(2)}, \theta_{22}^{(2)}) = (0.8, 0.2, 0.8)$, $(\theta_{11}^{(3)}, \theta_{12}^{(3)}, \theta_{22}^{(3)}) = (0.8, 0.2, 0.8)$;

- **SBM 2:** $(w_1^{(1)}, w_2^{(1)}) = (0.7, 0.3)$, $(w_1^{(2)}, w_2^{(2)}) = (0.5, 0.5)$, $(w_1^{(3)}, w_2^{(3)}) = (0.3, 0.7)$, $(\theta_{11}^{(1)}, \theta_{12}^{(1)}, \theta_{22}^{(1)}) = (0.7, 0.05, 0.8)$, $(\theta_{11}^{(2)}, \theta_{12}^{(2)}, \theta_{22}^{(2)}) = (0.7, 0.05, 0.8)$, $(\theta_{11}^{(3)}, \theta_{12}^{(3)}, \theta_{22}^{(3)}) = (0.7, 0.05, 0.8)$.

In Figure 3.1 we visualise the 21-node representatives for each of the $C = 3$ clusters, under each of the SBM structures (1 and 2) described above.

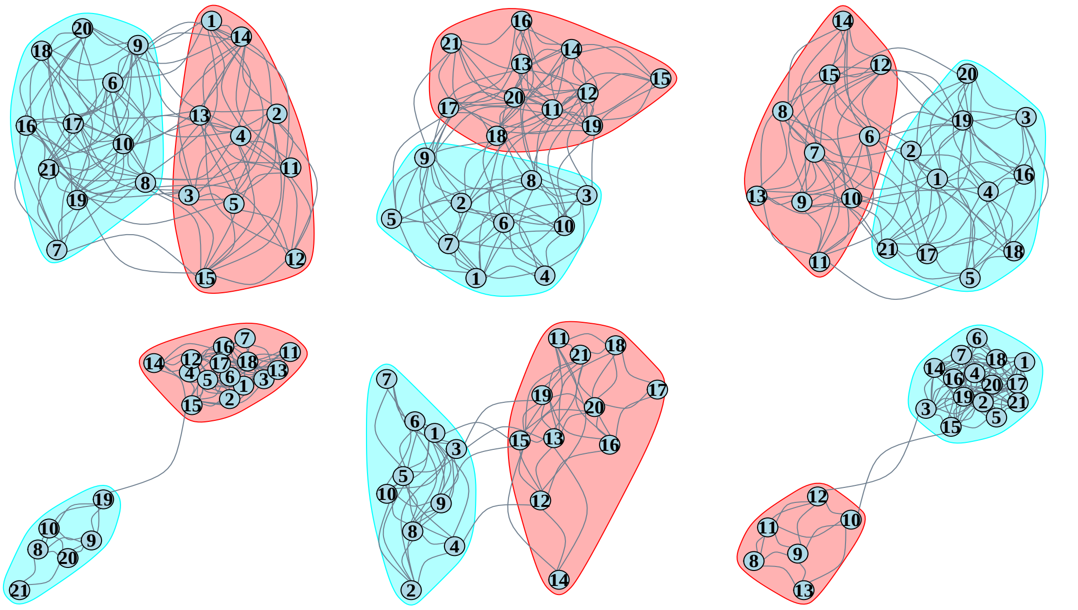

Figure 3.1: Top panel: Each network corresponds to a network representative for cluster $c \in \{1, 2, 3\}$. The network representatives were generated under the parameters specified for "SBM 1". Bottom panel: Each network corresponds to a network representative for cluster $c \in \{1, 2, 3\}$. The network representatives were generated under the parameters specified for "SBM 2". Nodes grouped under the same colour (red or blue) are nodes belonging in the same block. Nodes' layout is random.

Next, we generate a population of 180 networks by perturbing the edges of each representative through a measurement error process. Specifically, we generate edges for 60 networks in each cluster c, depending on the existence or non-existence of an edge in the representative network of the corresponding cluster c, given a false positive p_c and false negative q_c probability. The simulation regimes considered for p_c, q_c and the SBM parameters described above, are gathered in Table 3.1.

For each simulation regime, we run our MCMC for 500,000 iterations with a burn-in of 150,000. We assess the performance of our model in inferring the model parameters by obtaining the posterior means and credible intervals for these parameters. Indicatively, in Tables 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9 we present the posterior means and credible intervals for the false positive probabilities p_c, false negative probabilities q_c, the block probabilities of having an edge $\boldsymbol{\theta}^{\boldsymbol{c}}$. In addition in Table 3.10, 3.11, 3.12 we present the posterior means for the block membership probabilities $\boldsymbol{w}^{\boldsymbol{c}}$, for $c \in \{1, 2, 3\}$.

In order to investigate the performance of our algorithm in identifying the true repre-

Sim	(p_1,p_2,p_3)	(q_1,q_2,q_3)	$(w_1^{(1)},w_2^{(1)})$	$(w_1^{(2)},w_2^{(2)})$	$(w_1^{(3)},w_2^{(3)})$
1	(0.1,0.1,0.1)	(0.2,0.2,0.2)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)
2	(0.1,0.1,0.1)	(0.2,0.2,0.2)	(0.7,0.3)	(0.5,0.5)	(0.3,0.7)
3	(0.1,0.1,0.1)	(0.3,0.3,0.3)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)
4	(0.1,0.1,0.1)	(0.3,0.3,0.3)	(0.7,0.3)	(0.5,0.5)	(0.3,0.7)
5	(0.2,0.2,0.2)	(0.1,0.1,0.1)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)
6	(0.2,0.2,0.2)	(0.1,0.1,0.1)	(0.7,0.3)	(0.5,0.5)	(0.3,0.7)
7	(0.2,0.2,0.2)	(0.3,0.3,0.3)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)
8	(0.2,0.2,0.2)	(0.3,0.3,0.3)	(0.7,0.3)	(0.5,0.5)	(0.3,0.7)
9	(0.3,0.3,0.3)	(0.1,0.1,0.1)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)
10	(0.3,0.3,0.3)	(0.1,0.1,0.1)	(0.7,0.3)	(0.5,0.5)	(0.3,0.7)
11	(0.3,0.3,0.3)	(0.2,0.2,0.2)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)
12	(0.3,0.3,0.3)	(0.2,0.2,0.2)	(0.7,0.3)	(0.5,0.5)	(0.3,0.7)

Sim	$(\theta_{11}^{(1)},\theta_{12}^{(1)},\theta_{22}^{(1)})$	$(\theta_{11}^{(2)},\theta_{12}^{(2)},\theta_{22}^{(2)})$	$(\theta_{11}^{(3)},\theta_{12}^{(3)},\theta_{22}^{(3)})$
1	(0.8,0.2,0.8)	(0.8,0.2,0.8)	(0.8,0.2,0.8)
2	(0.7,0.05,0.8)	(0.7,0.05,0.8)	(0.7,0.05,0.8)
3	(0.8,0.2,0.8)	(0.8,0.2,0.8)	(0.8,0.2,0.8)
4	(0.7,0.05,0.8)	(0.7,0.05,0.8)	(0.7,0.05,0.8)
5	(0.8,0.2,0.8)	(0.8,0.2,0.8)	(0.8,0.2,0.8)
6	(0.7,0.05,0.8)	(0.7,0.05,0.8)	(0.7,0.05,0.8)
7	(0.8,0.2,0.8)	(0.8,0.2,0.8)	(0.8,0.2,0.8)
8	(0.7,0.05,0.8)	(0.7,0.05,0.8)	(0.7,0.05,0.8)
9	(0.8,0.2,0.8)	(0.8,0.2,0.8)	(0.8,0.2,0.8)
10	(0.7,0.05,0.8)	(0.7,0.05,0.8)	(0.7,0.05,0.8)
11	(0.8,0.2,0.8)	(0.8,0.2,0.8)	(0.8,0.2,0.8)
12	(0.7,0.05,0.8)	(0.7,0.05,0.8)	(0.7,0.05,0.8)

Table 3.1: Simulation regimes for 21-node networks and $C = 3$ clusters.

Sim	(p_1,p_2,p_3)
1	(0.09,0.10,0.11)
2	(0.10,0.10,0.10)
3	(0.10,0.10,0.10)
4	(0.11,0.10,0.10)
5	(0.19,0.19,0.20)
6	(0.20,0.20,0.20)
7	(0.20,0.20,0.19)
8	(0.20,0.20,0.19)
9	(0.31,0.30,0.31)
10	(0.31,0.30,0.31)
11	(0.30,0.29,0.30)
12	(0.30,0.30,0.28)

Table 3.2: Posterior means for false positive probabilities p_c, for $c \in \{1,2,3\}$, for simulation regimes 1-12.

Sim	(q_1,q_2,q_3)
1	(0.20,0.21,0.21)
2	(0.21,0.19,0.19)
3	(0.31,0.29,0.30)
4	(0.30,0.31,0.30)
5	(0.10,0.10,0.11)
6	(0.10,0.09,0.10)
7	(0.29,0.29,0.30)
8	(0.30,0.32,0.30)
9	(0.10,0.10,0.10)
10	(0.10,0.10,0.10)
11	(0.20,0.19,0.21)
12	(0.20,0.19,0.20)

Table 3.3: Posterior means for false negative probabilities q_c, for $c \in \{1,2,3\}$, for simulation regimes 1-12.

sentatives, we obtain the Hamming distance between the posterior representatives (after a burn-in of 150,000 and a lag of 50, leaving 7000 posterior draws) and the true representatives, and calculate the proportion of times that the distance is less or equal to 1,

Sim	$\boldsymbol{p_1}$	$\boldsymbol{p_2}$	$\boldsymbol{p_3}$	$\boldsymbol{q_1}$	$\boldsymbol{q_2}$	$\boldsymbol{q_3}$
1	(0.09,0.10)	(0.09,0.11)	(0.10,0.11)	(0.19,0.21)	(0.20,0.21)	(0.19,0.22)
2	(0.09,0.11)	(0.09,0.11)	(0.09,0.11)	(0.20,0.22)	(0.18,0.20)	(0.18,0.20)
3	(0.09,0.10)	(0.10,0.11)	(0.09,0.11)	(0.30,0.32)	(0.28,0.30)	(0.28,0.31)
4	(0.10,0.11)	(0.09,0.10)	(0.09,0.11)	(0.29,0.31)	(0.30,0.32)	(0.29,0.32)
5	(0.18,0.20)	(0.18,0.20)	(0.19,0.21)	(0.09,0.10)	(0.09,0.11)	(0.10,0.12)
6	(0.19,0.21)	(0.19,0.21)	(0.19,0.21)	(0.09,0.11)	(0.09,0.10)	(0.10,0.11)
7	(0.19,0.21)	(0.19,0.21)	(0.18,0.20)	(0.28,0.30)	(0.28,0.31)	(0.28,0.31)
8	(0.19,0.21)	(0.19,0.21)	(0.18,0.20)	(0.29,0.31)	(0.30,0.33)	(0.29,0.31)
9	(0.29,0.32)	(0.29,0.31)	(0.30,0.32)	(0.09,0.11)	(0.09,0.11)	(0.10,0.11)
10	(0.30,0.32)	(0.29,0.31)	(0.30,0.32)	(0.09,0.11)	(0.09,0.11)	(0.09,0.11)
11	(0.29,0.32)	(0.28,0.31)	(0.29,0.31)	(0.19,0.21)	(0.18,0.20)	(0.20,0.22)
12	(0.29,0.31)	(0.29,0.31)	(0.27,0.30)	(0.19,0.21)	(0.18,0.20)	(0.19,0.21)

Table 3.4: 95 % credible intervals for false positive probabilities p_c and false negative probabilities q_c, for $c \in \{1, 2, 3\}$, for simulation regimes 1-12.

Sim	$(\boldsymbol{\theta}^{(1)}_{11}, \boldsymbol{\theta}^{(1)}_{12}, \boldsymbol{\theta}^{(1)}_{22})$
1	(0.84,0.22,0.90)
2	(0.68,0.02,0.78)
3	(0.84,0.22,0.90)
4	(0.68,0.02,0.78)
5	(0.84,0.22,0.90)
6	(0.68,0.02,0.78)
7	(0.84,0.22,0.90)
8	(0.68,0.02,0.78)
9	(0.84,0.22,0.90)
10	(0.68,0.02,0.78)
11	(0.90,0.22,0.84)
12	(0.68,0.02,0.78)

Table 3.5: Posterior means for block specific edge probabilities $\boldsymbol{\theta_1}$ of cluster 1, for simulation regimes 1-12.

Sim	$(\boldsymbol{\theta}^{(2)}_{11}, \boldsymbol{\theta}^{(2)}_{12}, \boldsymbol{\theta}^{(2)}_{22})$
1	(0.83,0.21,0.82)
2	(0.60,0.05,0.71)
3	(0.83,0.21,0.82)
4	(0.60,0.05,0.71)
5	(0.83,0.21,0.82)
6	(0.60,0.05,0.71)
7	(0.83,0.21,0.82)
8	(0.71,0.05,0.60)
9	(0.82,0.21,0.83)
10	(0.60,0.05,0.71)
11	(0.82,0.21,0.83)
12	(0.71,0.05,0.60)

Table 3.6: Posterior means for block specific edge probabilities $\boldsymbol{\theta_2}$ of cluster 2, for simulation regimes 1-12.

Sim	$(\boldsymbol{\theta}^{(3)}_{11}, \boldsymbol{\theta}^{(3)}_{12}, \boldsymbol{\theta}^{(3)}_{22})$
1	(0.79,0.18,0.74)
2	(0.78,0.03,0.83)
3	(0.79,0.18,0.74)
4	(0.78,0.03,0.83)
5	(0.79,0.18,0.74)
6	(0.78,0.03,0.83)
7	(0.74,0.18,0.79)
8	(0.83,0.03,0.78)
9	(0.74,0.18,0.79)
10	(0.78,0.03,0.83)
11	(0.79,0.18,0.74)
12	(0.83,0.03,0.78)

Table 3.7: Posterior means for block specific edge probabilities $\boldsymbol{\theta_3}$ of cluster 3, for simulation regimes 1-12.

5 and 10 respectively, as seen in Table 3.13.

In addition, we assess the effectiveness of our algorithm in identifying the cluster membership of the networks, using the clustering entropy and clustering purity indices. Notably, we obtain the 7,000 posterior draws (burn-in of 150,000 and lag of 50) for the cluster membership z, calculate the clustering entropy and clustering purity with respect to the true membership of the networks and obtain the mean, for each simulation regime. We note here that a clustering entropy value of 0 and clustering purity value of 1, indicate a perfect cluster allocation of the networks. The results are illustrated in Table 3.14.

In Figures 3.2, 3.3, 3.4, 3.5, we display the traceplots and histograms for the false

Sim	$\boldsymbol{\theta}_{\mathbf{11}}^{(\mathbf{1})}$	$\boldsymbol{\theta}_{\mathbf{12}}^{(\mathbf{1})}$	$\boldsymbol{\theta}_{\mathbf{22}}^{(\mathbf{1})}$	$\boldsymbol{\theta}_{\mathbf{11}}^{(\mathbf{2})}$	$\boldsymbol{\theta}_{\mathbf{12}}^{(\mathbf{2})}$	$\boldsymbol{\theta}_{\mathbf{22}}^{(\mathbf{2})}$
1	(0.73,0.94)	(0.15,0.30)	(0.82,0.97)	(0.73,0.92)	(0.14,0.29)	(0.70,0.92)
2	(0.59,0.77)	(0.00,0.04)	(0.58,0.96)	(0.47,0.72)	(0.01,0.09)	(0.58,0.83)
3	(0.73,0.94)	(0.15,0.30)	(0.82,0.97)	(0.73,0.92)	(0.14,0.29)	(0.70,0.92)
4	(0.60,0.77)	(0.00,0.04)	(0.59,0.96)	(0.47,0.72)	(0.01,0.09)	(0.58,0.83)
5	(0.73,0.94)	(0.15,0.30)	(0.82,0.97)	(0.73,0.92)	(0.14,0.29)	(0.70,0.92)
6	(0.60,0.77)	(0.00,0.04)	(0.59,0.96)	(0.47,0.72)	(0.01,0.09)	(0.58,0.83)
7	(0.73,0.94)	(0.15,0.30)	(0.82,0.97)	(0.73,0.92)	(0.14,0.29)	(0.70,0.92)
8	(0.60,0.77)	(0.00,0.04)	(0.59,0.96)	(0.58,0.83)	(0.01,0.09)	(0.47,0.72)
9	(0.73,0.94)	(0.15,0.30)	(0.82,0.97)	(0.70,0.92)	(0.14,0.29)	(0.73,0.92)
10	(0.60,0.77)	(0.00,0.04)	(0.59,0.96)	(0.47,0.73)	(0.01,0.09)	(0.57,0.83)
11	(0.82,0.97)	(0.15,0.30)	(0.73,0.94)	(0.70,0.92)	(0.14,0.29)	(0.73,0.92)
12	(0.59,0.77)	(0.00,0.04)	(0.58,0.96)	(0.58,0.83)	(0.01,0.09)	(0.47,0.72)

Table 3.8: 95 % credible intervals for block specific edge probabilities $\boldsymbol{\theta_1}$ of cluster 1 and $\boldsymbol{\theta_2}$ of cluster 2, for simulation regimes 1-12.

Sim	$\boldsymbol{\theta}_{\mathbf{11}}^{(\mathbf{3})}$	$\boldsymbol{\theta}_{\mathbf{12}}^{(\mathbf{3})}$	$\boldsymbol{\theta}_{\mathbf{22}}^{(\mathbf{3})}$
1	(0.68,0.90)	(0.11,0.26)	(0.63,0.85)
2	(0.59,0.96)	(0.00,0.06)	(0.75,0.89)
3	(0.68,0.90)	(0.11,0.26)	(0.63,0.85)
4	(0.58,0.96)	(0.00,0.06)	(0.75,0.89)
5	(0.68,0.90)	(0.12,0.26)	(0.63,0.85)
6	(0.59,0.96)	(0.00,0.06)	(0.75,0.89)
7	(0.63,0.85)	(0.12,0.26)	(0.68,0.90)
8	(0.75,0.89)	(0.00,0.06)	(0.59,0.96)
9	(0.63,0.85)	(0.12,0.26)	(0.68,0.90)
10	(0.59,0.96)	(0.00,0.06)	(0.75,0.90)
11	(0.68,0.90)	(0.11,0.26)	(0.63,0.85)
12	(0.75,0.89)	(0.00,0.06)	(0.59,0.96)

Table 3.9: 95 % credible intervals for block specific edge probabilities $\boldsymbol{\theta_3}$ of cluster 3, for simulation regimes 1-12.

positive p and false negative q probabilities, for the third simulation regime of Table 3.1 indicatively, after a burn-in of 150,000 iterations. In Appendix A.2, we also present the traceplots and histograms for θ probabilities of an edge occurrence.

We further investigate the performance of the model for various noise levels in the network population. Specifically, we generate network populations by perturbing the edges of the representatives of SBM structure 1, illustrated in Figure 3.1, with varying sizes of the false positive and negative probabilities. Specifically we consider the simulation regimes presented in Tables 3.15 and 3.16.

For each regime presented in Table 3.15 and 3.16, we generate a network population, and run our MCMC algorithm for 500,000 iterations with a burn-in of 150,000. We obtain the posterior means for the varying false positive probability (Table 3.15) and

Sim	$(\boldsymbol{w_1^{(1)}}, \boldsymbol{w_2^{(1)}})$
1	(0.48,0.52)
2	(0.70,0.30)
3	(0.48,0.52)
4	(0.70,0.30)
5	(0.48,0.52)
6	(0.70,0.30)
7	(0.48,0.52)
8	(0.70,0.30)
9	(0.48,0.52)
10	(0.70,0.30)
11	(0.52,0.48)
12	(0.70,0.30)

Table 3.10: Posterior means for the probability of a node to belong to a block $\boldsymbol{w_1}$ for cluster 1 for simulation regimes 1-12.

Sim	$(\boldsymbol{w_1^{(2)}}, \boldsymbol{w_2^{(2)}})$
1	(0.52,0.48)
2	(0.52,0.48)
3	(0.52,0.48)
4	(0.52,0.48)
5	(0.52,0.48)
6	(0.52,0.48)
7	(0.52,0.48)
8	(0.48,0.52)
9	(0.48,0.52)
10	(0.52,0.48)
11	(0.48,0.52)
12	(0.48,0.52)

Table 3.11: Posterior means for the probability of a node to belong to a block $\boldsymbol{w_2}$ for cluster 2 for simulation regimes 1-12.

Sim	$(\boldsymbol{w_1^{(3)}}, \boldsymbol{w_2^{(3)}})$
1	(0.48,0.52)
2	(0.30,0.70)
3	(0.48,0.52)
4	(0.30,0.70)
5	(0.48,0.52)
6	(0.30,0.70)
7	(0.52,0.48)
8	(0.70,0.30)
9	(0.52,0.48)
10	(0.30,0.70)
11	(0.48,0.52)
12	(0.70,0.30)

Table 3.12: Posterior means for the probability of a node to belong to a block $\boldsymbol{w_3}$ for cluster 3 for simulation regimes 1-12.

Sim	$\boldsymbol{d_H <= 1}$	$\boldsymbol{d_H <= 5}$	$\boldsymbol{d_H <= 10}$
1	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
2	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
3	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
4	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
5	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
6	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
7	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
8	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
9	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
10	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
11	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)
12	(1.00,1.00,1.00)	(1.00,1.00,1.00)	(1.00,1.00,1.00)

Table 3.13: Proportion of times that the Hamming distance between the posterior representatives and the true representatives is less or equal than 1, 5 and 10 respectively, for simulation regimes 1-12.

the varying false negative probability (Table 3.16), for each cluster $c = \{1, 2, 3\}$, and plot them against their true values as seen in Figures 3.6 and 3.7 respectively.

We observe that the posterior means lie mostly close to the $y = x$ line, indicating that our model performs well in inferring the true false positive and false negative probabilities, even for high noise levels. However, in Figure 3.6, for the highest noise value of $p_c = 0.45$ for $c = \{1, 2, 3\}$, the posterior mean of the false negative probability of cluster 3, p_3, is equal to 0.2, which is away from its true value. These results suggest that our model performs well in most cases, even for high noise levels, however we must be cautious when drawing inference for network populations with great variability in their

Sim	Mean Entropy	Mean Purity
1	0	1
2	0	1
3	0	1
4	0	1
5	0	1
6	0	1
7	0	1
8	0	1
9	0	1
10	0	1
11	0	1
12	0	1

Table 3.14: Mean clustering entropy and clustering purity for simulation regimes 1-12.

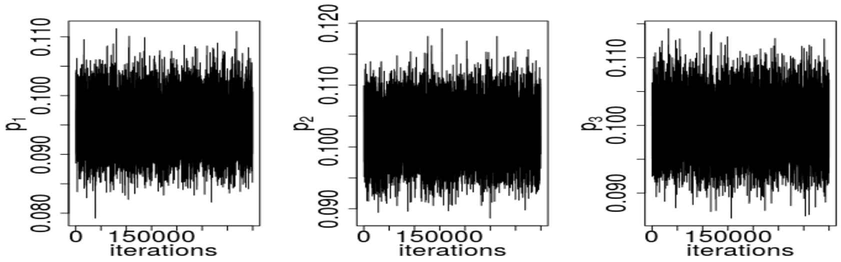

Figure 3.2: Traceplots of the posterior draws for false positive probability p_c, for $c \in \{1, 2, 3\}$, for 500,000 iterations with a burn-in of 150,000.

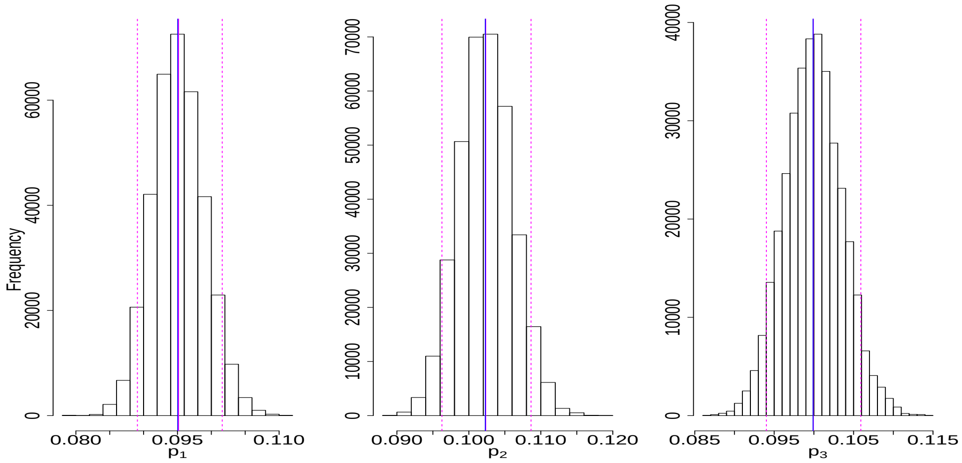

Figure 3.3: Histograms for false positive probability p_c, for cluster $c \in \{1, 2, 3\}$. The pink solid line indicates the posterior mean, the blue solid line the posterior median and the pink dashed lines indicate the 95% credible interval.

structure. Notably, different edge connectivity patterns among networks in a population indicate variability in the networks' structure, and one way to quantify such variability in a population is to perform EDA by calculating pairwise distances between networks, for different distance metrics.

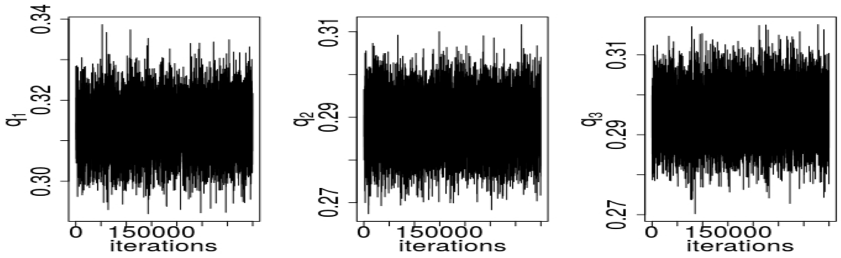

Figure 3.4: Traceplots of the posterior draws for false negative probability q_c, for $c \in \{1, 2, 3\}$, for 500,000 iterations with a burn-in of 150,000.

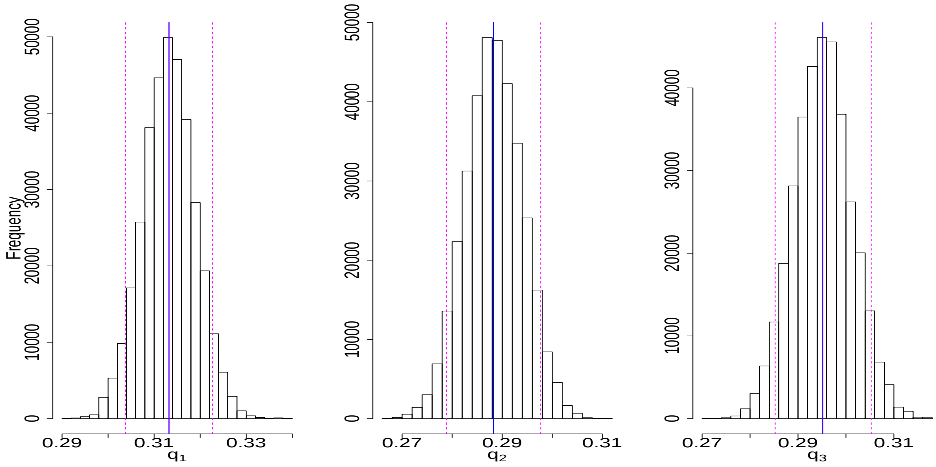

Figure 3.5: Histograms for false negative probability q_c, for cluster $c \in \{1, 2, 3\}$. The pink solid line indicates the posterior mean, the blue solid line the posterior median and the pink dashed lines indicate the 95% credible interval.

Sim	$(\boldsymbol{p_1, p_2, p_3})$	$(\boldsymbol{q_1, q_2, q_3})$
1	(0.01,0.01,0.01)	(0.1,0.1,0.1)
2	(0.05,0.05,0.05)	(0.1,0.1,0.1)
3	(0.1,0.1,0.1)	(0.1,0.1,0.1)
4	(0.15,0.15,0.15)	(0.1,0.1,0.1)
5	(0.2,0.2,0.2)	(0.1,0.1,0.1)
6	(0.25,0.25,0.25)	(0.1,0.1,0.1)
7	(0.3,0.3,0.3)	(0.1,0.1,0.1)
8	(0.35,0.35,0.35)	(0.1,0.1,0.1)
9	(0.4,0.4,0.4)	(0.1,0.1,0.1)
10	(0.45,0.45,0.45)	(0.1,0.1,0.1)

Table 3.15: Simulation regimes for varying sizes of false positive probabilities p_c and fixed false negative probabilities q_c, for $c \in \{1, 2, 3\}$ and 21-node networks.

Sim	$(\boldsymbol{p_1, p_2, p_3})$	$(\boldsymbol{q_1, q_2, q_3})$
1	(0.1,0.1,0.1)	(0.01,0.01,0.01)
2	(0.1,0.1,0.1)	(0.05,0.05,0.05)
3	(0.1,0.1,0.1)	(0.1,0.1,0.1)
4	(0.1,0.1,0.1)	(0.15,0.15,0.15)
5	(0.1,0.1,0.1)	(0.2,0.2,0.2)
6	(0.1,0.1,0.1)	(0.25,0.25,0.25)
7	(0.1,0.1,0.1)	(0.3,0.3,0.3)
8	(0.1,0.1,0.1)	(0.35,0.35,0.35)
9	(0.1,0.1,0.1)	(0.4,0.4,0.4)
10	(0.1,0.1,0.1)	(0.45,0.45,0.45)

Table 3.16: Simulation regimes for varying sizes of false negative probabilities q_c and fixed false positive probabilities p_c, for $c \in \{1, 2, 3\}$ and 21-node networks.

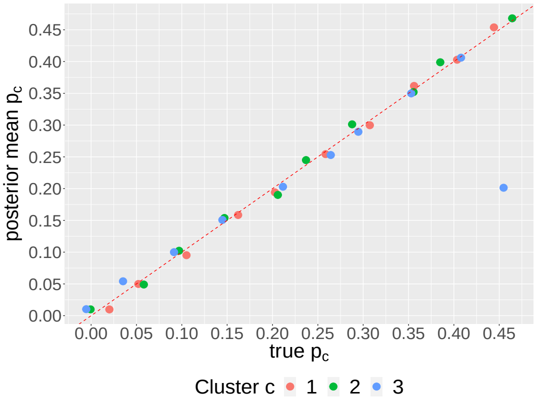

Figure 3.6: Posterior means for false positive probabilities p_c for $c \in \{1,2,3\}$ (y axis), plotted against the true values of p_c (x axis). The red dotted line corresponds to the $y = x$.

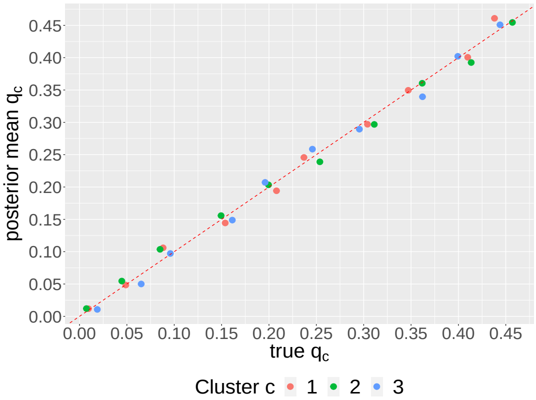

Figure 3.7: Posterior means for false negative probabilities q_c for $c \in \{1,2,3\}$ (y axis), plotted against the true values of q_c (x axis). The red dotted line corresponds to the $y = x$.

3.4.2 Varying sizes of networks and network populations

In this Section, we explore how well our model infers the parameters with respect to various network sizes and sample sizes. We keep $C = 3$ clusters and $B = 2$ blocks. We consider four different network sizes of 25, 50, 75 and 100 nodes, and simulate populations of 45, 90, 135, 180, 225, 270 and 315 networks, for each network size respectively.

To simulate the network populations, we first generate the network representatives of each cluster under the SBM network model. We specify the parameters of the SBM model so that the expected degree of the resulting network representatives is preserved, for all network sizes considered. The network representatives obtained for each cluster and network size considered, are visualised in Figures 3.8, 3.9, 3.10 and 3.11. The underlying community structures formed by the nodes of the representatives vary, as can be noticed from the Figures. The resulting network populations are obtained by perturbing the edges of each network representative, for each network size considered, with a false positive p_c and false negative q_c probability fixed at 0.08, for $c \in \{1, 2, 3\}$.

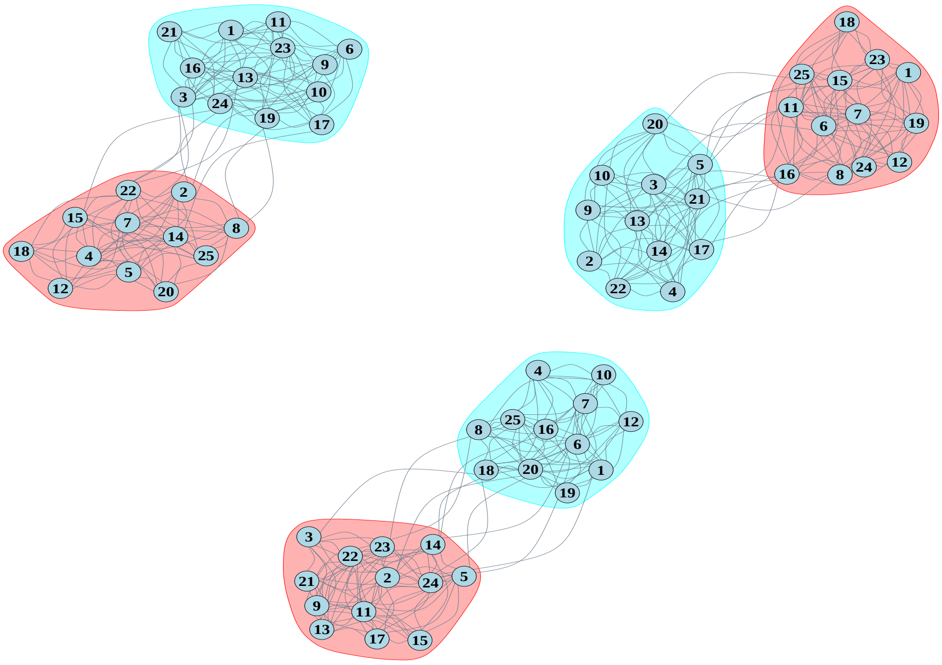

Figure 3.8: 25-node representatives of cluster 1 (top left), cluster 2 (top right) and cluster 3 (bottom).

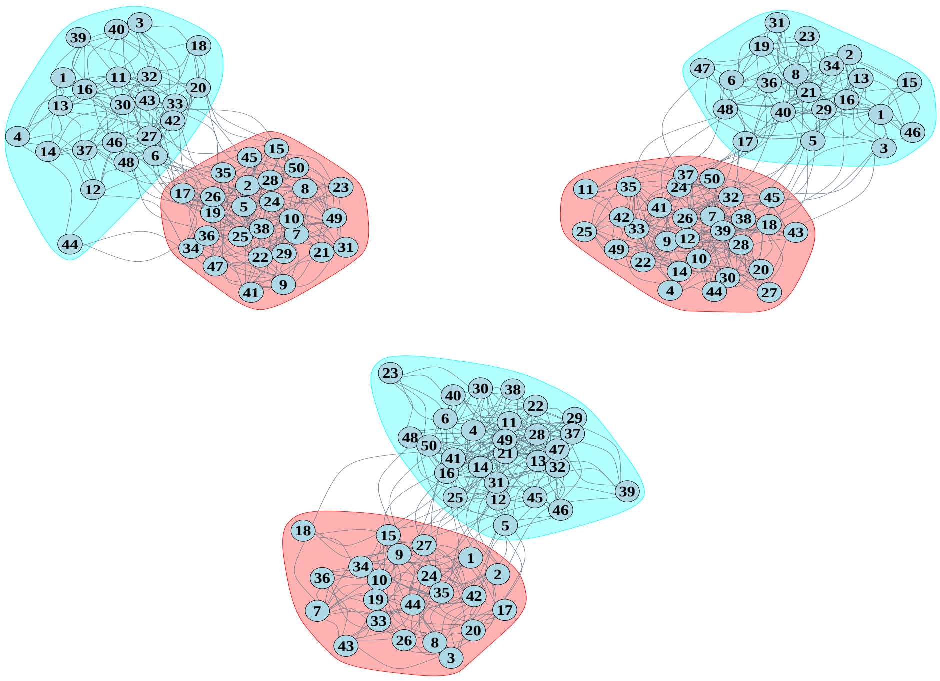

Figure 3.9: 50-node representatives of cluster 1 (top left), cluster 2 (top right) and cluster 3 (bottom).

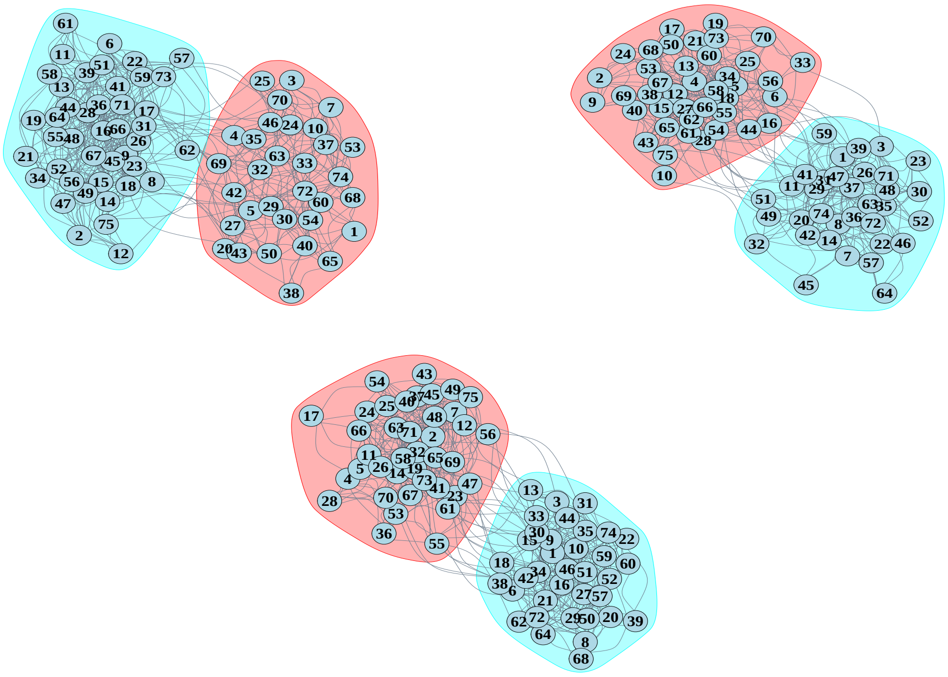

Figure 3.10: 75-node representatives of cluster 1 (top left), cluster 2 (top right) and cluster 3 (bottom).

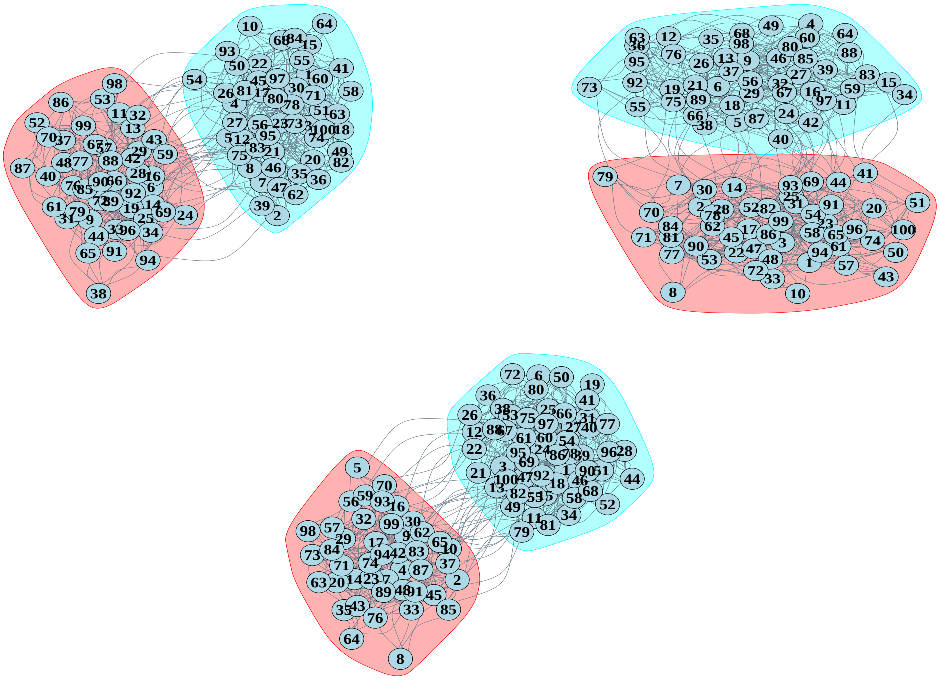

Figure 3.11: 100-node representatives of cluster 1 (top left), cluster 2 (top right) and cluster 3 (bottom).

For each simulated data set we run the MCMC for 500,000 iterations with a burn-in of 150,000, using Jeffrey's non-informative priors. We demonstrate the performance of our model by obtaining the absolute error of the model parameters for each simulation regime, as seen in Figures 3.12 and 3.13. Specifically, the plots demonstrate the variability of the absolute error (y axis) for various sample sizes (x axis). The absolute error is the absolute value of the difference between the posterior means obtained after burn-in, and the true value of the parameter. The different coloured lines and dots correspond to the different clusters considered. The multiple dots of the same colour correspond to 5 replications of the MCMC, each on a different randomly generated data set. The lines connect the absolute errors averaged across the replications.

The plots indicate that the sample size affects the performance of our model for the network sizes considered. We observe as the number of nodes increase, there is an increase in the required number of networks to preserve the same level of accuracy in estimation. Nevertheless, even for large networks and small sample sizes, it is encouraging to see the posterior means obtained are not far away from their true values.

We further visualise the performance of our model in identifying the true representatives of each cluster, for the various sample sizes and network sizes. Indicatively, we

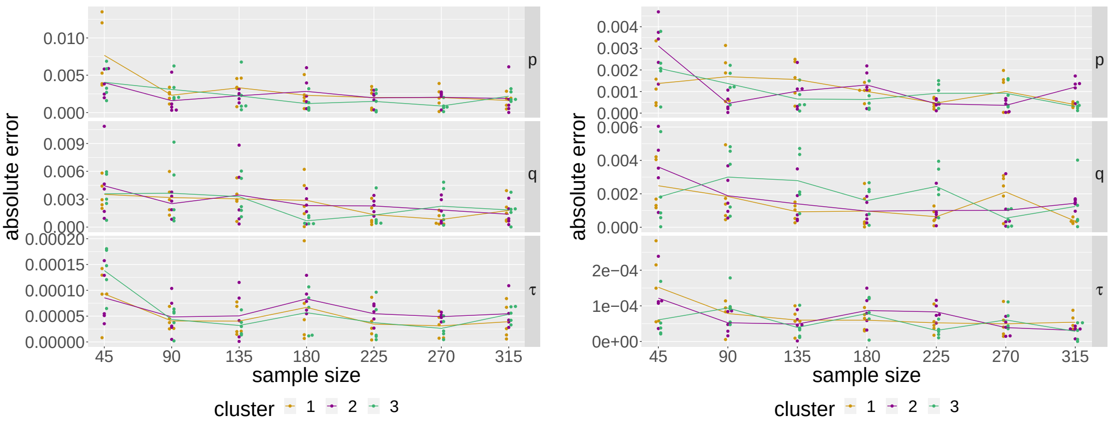

Figure 3.12: Left: Absolute error (y axis) for model parameters $\boldsymbol{p}$, $\boldsymbol{q}$ and $\boldsymbol{\tau}$, for 25-node networks and varying population sizes (x axis). Right: Absolute error (y axis) for model parameters $\boldsymbol{p}$, $\boldsymbol{q}$ and $\boldsymbol{\tau}$, for 50-node networks and varying population sizes (x axis).

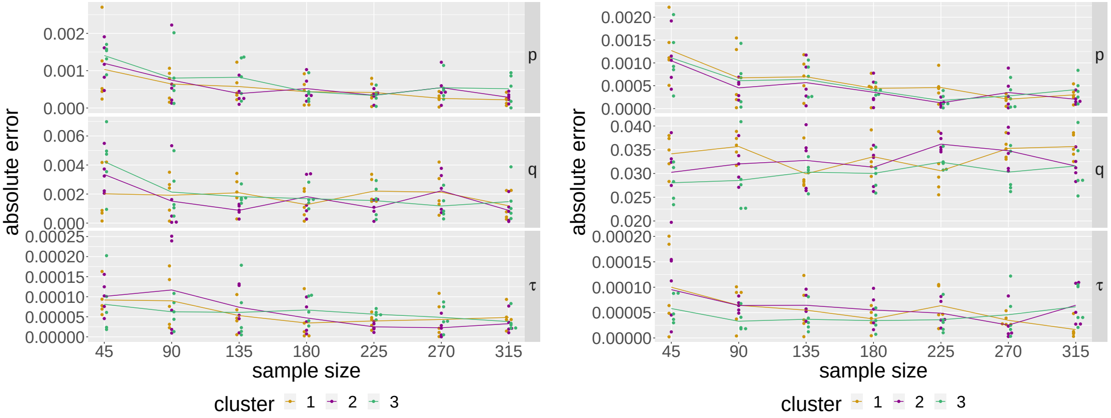

Figure 3.13: Left: Absolute error (y axis) for model parameters $\boldsymbol{p}$, $\boldsymbol{q}$ and $\boldsymbol{\tau}$, for 75-node networks and varying population sizes (x axis). Right: Absolute error (y axis) for model parameters $\boldsymbol{p}$, $\boldsymbol{q}$ and $\boldsymbol{\tau}$, for 100-node networks and varying population sizes (x axis).

obtain the proportion of times that the Hamming distance between the representative posterior draws and the true representative, for the corresponding cluster, is less or equal than 1, 5 and 10. We consider the 350,000 posterior draws for the representatives, after a burn-in of 150,000 iterations. The results are presented in Figure 3.14. The multiple subfigures correspond to the 25, 50, 75 and 100 node representatives, and y axis indicates the proportion of times the Hamming distance is less or equal to 1, 5 and 10 respectively.

We observe that for the 25-node and 50-node cases, the true representatives are closely identified 100% of the times, considering the Hamming distance between the posterior draws and the true representatives. For the 75-node case the posterior representative of each cluster is also closely revealed with only 10 edges different from the true representative, 100% of the times. Lastly, for the 100-node case, the identification of the true representative of each cluster is challenging due to the huge space of graphs

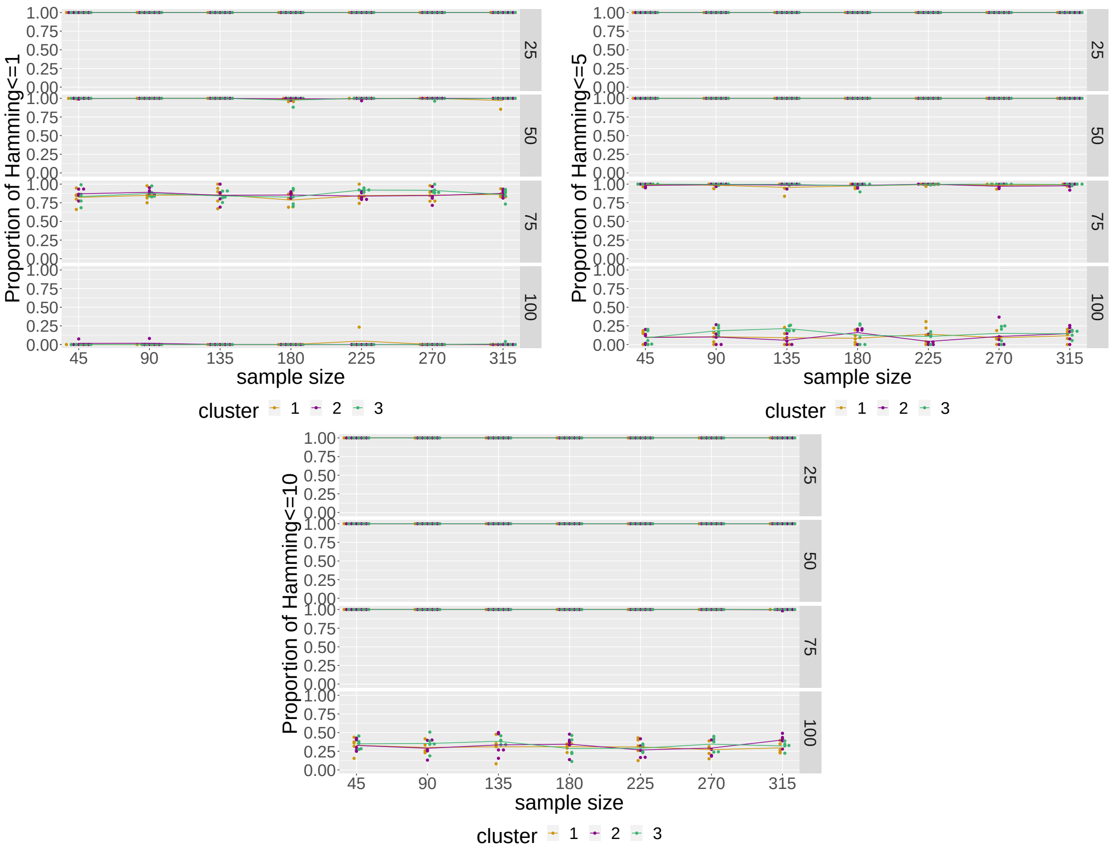

Figure 3.14: Top Left: Proportion of times the Hamming distance is less or equal to 1 (y axis) for 25-node, 50-node, 75-node and 100-node network representatives and varying population sizes (x axis). Top Right: Proportion of times the Hamming distance is less or equal to 5 (y axis) for 25-node, 50-node, 75-node and 100-node network representatives and varying population sizes (x axis). Bottom: Proportion of times the Hamming distance is less or equal to 10 (y axis) for 25-node, 50-node, 75-node and 100-node network representatives and varying population sizes (x axis).

that is being explored.

3.5 Real data examples

In this Section we apply our model on two real-world data examples. First, in Section 3.5.1 we apply our mixture model on the Tacita data introduced in Section 3.3.1, which motivated our study. Second, in Section 3.5.2 we apply our model on a collection of networks representing connections among regions of the brain for a group of individuals.

3.5.1 Tacita application

Data preprocessing and EDA

A motivating example for our study is the population of networks resulting from the Tacita mobile application. As introduced in Section 3.3.1, Tacita was created to enable users to interact with displays located on Lancaster University campus. Displays are large screens located at various public places, both indoors and outdoors, showing diverse content such as advertisements, news and bus timetables. Tacita application allows personalisation of the content shown on a display, according to the interests of the user detected in close proximity of the display. In Figure 3.15 (left) we show the display located at Alexandra Square on Lancaster University campus, while Figure 3.15 (right) shows the locations of the displays installed on Lancaster University campus.

Tacita records the consecutive displays visited by users, along with the time visited and the type of content shown on the display. One way to represent the data collected from the Tacita application is through a network, where nodes correspond to displays and edges correspond to movements of users among the displays. Consequently, we obtain a multiple network data set, where each network observation therein corresponds to a user's movements across displays. Under this representation of the Tacita data, we are

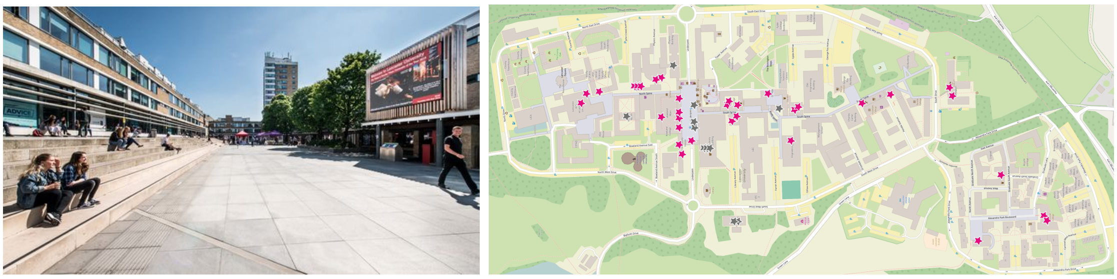

Figure 3.15: Left: Display located at Alexandra Square on Lancaster University campus. Right: Lancaster University campus map. Pink marks indicate the locations of the displays on campus.

interested in identifying different movement patterns of the individuals, and clustering them according to their similarities.

The Tacita data were originally in raw format, thus preprocessing was required to obtain the final network representations. As typically occurs with raw data, there were faulty entries that we needed to eliminate. The faulty entries involved NAN or 0 value ids of users, requests and displays, as well as other type of false entries such as the detection of a user at two distant displays with very short time difference. To eliminate the latter, we obtained a minimum time threshold for which a user can move from one display to another, considering the coordinates of the displays and a maximum speed that a user can move.

To derive a network representation for each user, we obtain the consecutive displays that the user visited in chronological order for all the days and all hours of the day that the user has been detected, and create the edges of the networks from the pairs of consecutive displays visited. Thus, each network represents the aggregated movements of each user across campus displays. The final data sample consists of 120 undirected and unweighted network observations, that share the same set of 37 nodes corresponding to displays presented in Table 3.17.

Display name	**Node label**
SCC (C-floor)	1
Infolab Foyer	2
Faraday Left	3
Engineering Foyer (far)	4
LZ1	5
LZ3	6
Furness 1	7
Furness 2	8
Furness College	9
Engineering Foyer (near)	10
LEC 1	11
LEC 2	12
County College	13
Lonsdale College	14
Grizedale College	15
The Base	16
Faraday B	17
Faraday C	18
Bowland JCR	19

Display name	**Node label**
ISS	20
Pendle College	21
New Engineering	22
Fylde College	23
Graduate College	24
Library A	25
Library B	26
Library C	27
Bowland Main B	28
Bowland North B	29
Hotel Conference	30
Chemistry A	31
Psychology	32
Physics	33
Law 2	34
Law 1	35
Welcome Screen 1	36
Welcome Screen 2	37

Table 3.17: Node label assigned to each display at Lancaster University.

As our mixture model requires the pre-specification of the number of clusters C, we conduct exploratory data analysis (EDA) to explore the potential number of clusters formed within the Tacita network population. One way to explore that, is through the use of network distance metrics. Specifically, for each distance metric we can derive a distance matrix that encloses the pairwise distances of the networks in the population. In that sense, the (i, j) element of a distance matrix encloses the value of the distance between graphs $\mathcal{G}_i$ and $\mathcal{G}_j$, for a specified distance metric. We note here that in our EDA we consider various distance metrics, as our model is not a distance-based model, thus different metrics can give us different information with respect to the presence of clusters in the network population.

One graphical representation of the pairwise distances between networks using distance matrices is the Multi Dimensional Scaling (MDS) plot. The MDS algorithm maps objects in a 2-d space respecting their pairwise distances. In Figures 3.16 and 3.17, we demonstrate the MDS representation of the distance matrices calculated under the Hamming distance, the Jaccard distance, the l_2 distance and the distance based on wavelets. For a description of the distance metrics considered, see Appendix A.3 and Section 1.1 of Chapter 1.

The EDA representations derived can be interpreted in more than one way. For the Hamming and the Jaccard distance metrics, a natural assumption is the presence of two clusters of networks in the population. However, one might argue that these plots suggest the formation of a single cluster of networks with several networks being away from that cluster, playing the role of outlier network observations. On the other hand, the visualisation of the distance matrices for the l_2 distance and the wavelets, suggest more clearly the presence of two or three clusters of networks.

Due to the multiple interpretability of the EDA results, we explore the following cases:

1. We assume $C = 2$ clusters of networks, with component specific representatives.

2. We assume the presence of a majority cluster of networks and a cluster of outlier networks.

3. We assume $C = 3$ clusters of networks, with component specific representatives.

The initialisation of our algorithm and the results are presented in the following

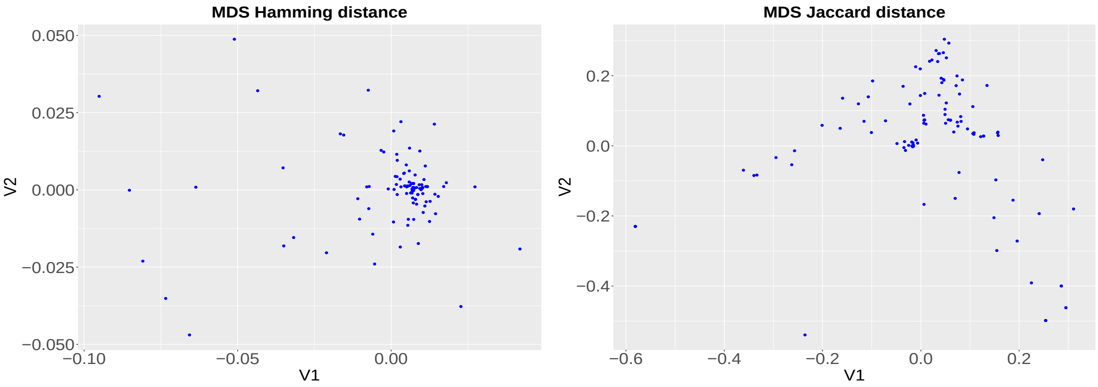

Figure 3.16: Left: MDS for Hamming distance matrix for the Tacita data. Right: MDS for Jaccard distance matrix for the Tacita data.

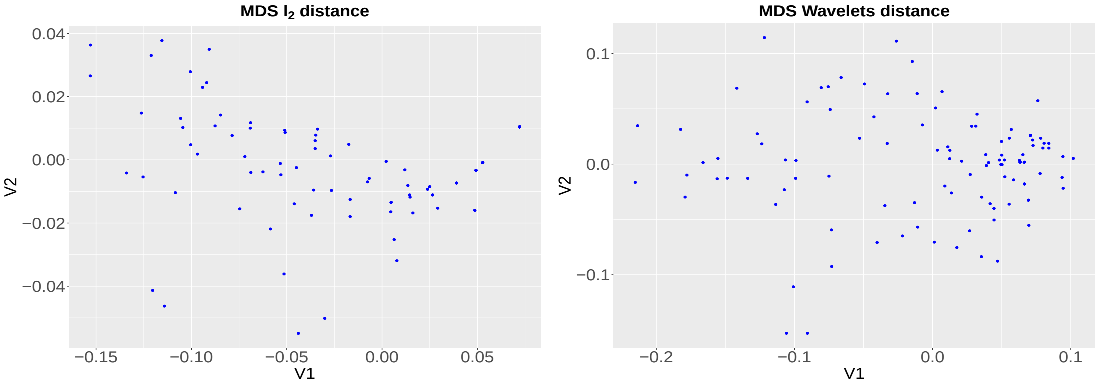

Figure 3.17: Left: MDS for wavelets distance matrix for the Tacita data. Right: MDS for l_2 distance matrix for the Tacita data.

section.

Results

We first start with the case where we have pre-specified a number of clusters $C = 2$. To initialise the networks' cluster membership, we combine the results from three different distance metrics, namely the Jaccard, the l_2 and the wavelets distance. Specifically, we use a k-means algorithm using the R package `kmed` (Budiaji [2019]) to determine three different cluster memberships, corresponding to the three different metrics considered, and determine the final cluster membership initialisation using majority vote, i.e. by determining which networks are consistently allocated to one of the two clusters among the three memberships obtained. We initialise the representative of each cluster by generating its edges using independent Bernoulli draws. The probability with which we draw an edge between two specific nodes of the representative, corresponds to the proportion of times that we see that edge in the network data of the corresponding cluster.

To initialise the block membership of the nodes of each initialised cluster representative, we use the R package `blockmodels` (INRA and Leger [2015]) to obtain SBM estimates for each representative. The results suggest the presence of two underlying blocks of nodes in each initialised representative, however, the estimated block membership of the nodes is different between the two representatives. As previously stated, our algorithm allows inferences of different block structures for the representatives of different clusters, thus we are able to initialise the block structure of each representative using the results from the R package `blockmodels`. This is a particularly useful attribute of our model, as it is anticipated that network data allocated in different clusters can have different structures, and the structure of the networks in each cluster is depicted by the network representative of the cluster.

We run our MCMC for 500,000 iterations with a burn-in of 100,000. In Figures 3.18 and 3.19, we present the traceplots of the false negative and false positive probabilities for each of the two clusters. In Figure 3.20, we obtain the proportion of times that an individual is allocated to one of the two clusters, while in Figures 3.21 we present the proportion of times that the nodes of each representative belong in each of the two blocks specified. To visualise a network representative for each cluster, we obtain the posterior mode from the posterior network draws of each cluster. In Figure 3.22, we visualise the posterior mode of each cluster network representative, with posterior masses 100% and 17% for cluster labelled 1 and 2 respectively. The posterior mass of each posterior mode corresponds to the proportion of times that the corresponding network representative was drawn among the last 100,000 posterior draws. The node labels of the posterior mode network representatives correspond to the numbering of the displays presented in Table 3.17, and the nodes positions (layout) correspond to the positions of the displays in the physical space, thus distances correspond to real physical distances between the displays. The two different colours of the nodes denote the block structure inferred by our algorithm for each network representative.

For cluster 1, we observe that the representative concentrating the whole posterior mass, is sparse having only few edges. Specifically, we note that the edges of the representative correspond to movements of individuals among displays that are very close to each other (e.g. edge between nodes 4 and 10 corresponding to displays in the Engineering building). In addition, most of the networks in the population are allocated to

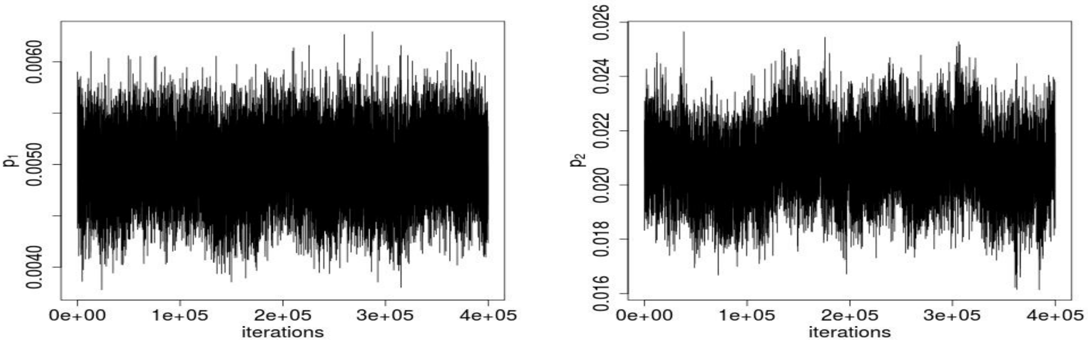

Figure 3.18: Traceplots for false positive probabilities p_c for $c = 1$ (left) and $c = 2$ (right), for 500,000 iterations and a burn-in of 100,000.

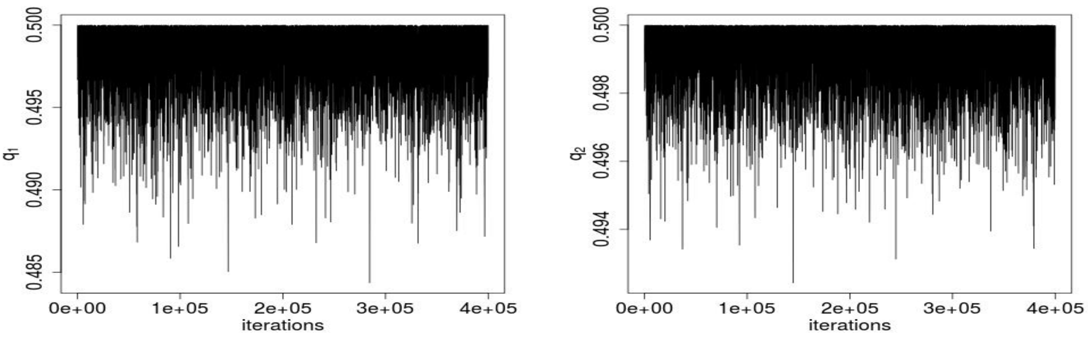

Figure 3.19: Traceplots for false negative probabilities q_c for $c = 1$ (left) and $c = 2$ (right), for 500,000 iterations and a burn-in of 100,000.

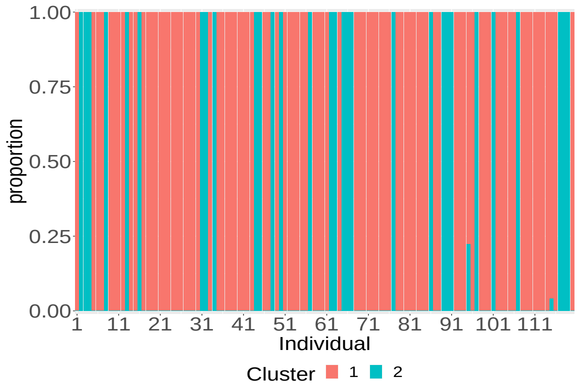

Figure 3.20: Proportion of times (y axis) an individual (x axis) is allocated to each of the 2 clusters, after a burn-in of 100,000 iterations.

this cluster, with a very small false positive probability with posterior mean 0.0049 and very large false negative probability with posterior mean 0.49. The small false positive probability indicates that the edges observed in the network data are correctly recorded, while the high false negative probability indicates that there are edges in the network data that we do not observe. One way to explain the high false negative probabilities

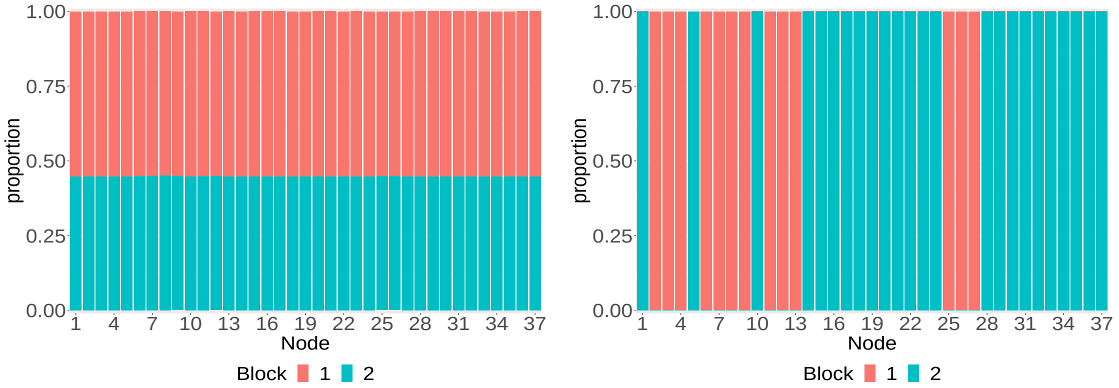

Figure 3.21: Left: Proportion of times (y axis) that each node (x axis) of the representative network of cluster labelled 1 is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations. Right: Proportion of times (y axis) that each node (x axis) of the representative network of cluster labelled 2 is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations.

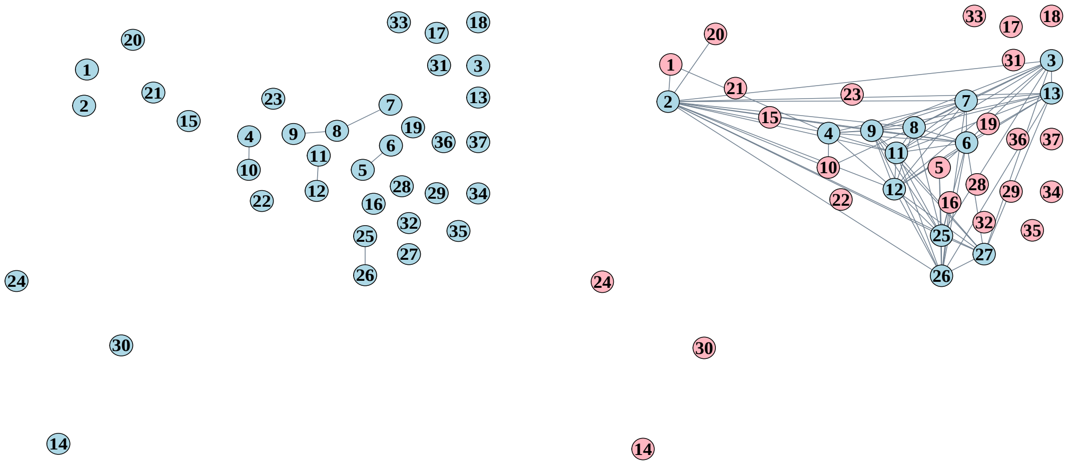

Figure 3.22: Left: Posterior mode of representative of cluster $c = 1$ with posterior mass 100%, for last 100,000 iterations. Right: Posterior mode of representative of cluster $c = 2$ with posterior mass 17%, for last 100,000 iterations.

inferred by our model is the presence of very sparse networks in the population, suggesting that there might be movements of users that we do not observe. Unrecorded movements of users can be attributed to lost or weak Wi-Fi connection which affects the functionality of Tacita application. This remark can also be justified by the movements observed in the representative network that correspond to movements among displays very close to each other, suggesting that when a user is in a building the WiFi connection is preserved, thus the application can record the movements of the user. Another possible triggering factor for the high false negative probabilities inferred for both clusters, can be the misspecification of the number of clusters in our model. The assumption of two underlying clusters might be restrictive for the Tacita data, causing the formation of clusters of users with very dissimilar movements, which subsequently leads to high noise levels. Specifically, the high false negative probability inferred for cluster $c = 1$ with the sparser networks allocated to it, could be attributed to our algorithm clustering users with short, different movements into a single cluster. Later in this Section, we further investigate how the assumption of $C = 3$ clusters affects inferences for the Tacita data.

For Cluster 2, we observe that the posterior mode of the representative network concentrates a relatively smaller posterior mass, while being denser compared to representative of cluster 1. Moreover, an SBM structure is revealed, with the mostly connected nodes belonging in the same block. We observe that the block with the denser connections mostly represents the central part of campus close to the central square, namely Alexandra Square, which is a highly visited area of Lancaster University campus. However, we also notice that displays located at the eastern and western part of campus are also allocated to the block with the centrally located displays. Notably, the Library (nodes 25, 26, 27) and the Furness College (nodes 7, 8, 9) are centrally located buildings on campus, while County College (node 13) and Infolab (node 2) are away from central campus. The block structure assumed for the representative allows us to understand the movement behaviour of the users within the cluster, by indicating the groups of displays among which is more probable to see a connection. The goal of our analysis is to understand the structural properties of the users' movements without imposing any restrictions with respect to spatial information, however it would be interesting to explore how inferences change using the information about the displays' locations in the physical space. Lastly, we observe that a smaller proportion of networks is allocated to Cluster 2, indicating that the denser networks in the population are fewer. Notably, the posterior mean of the probability of a network to belong to cluster c, τ_c, is equal to 0.26 for cluster $c = 2$ while for $c = 1$ it is 0.74 as can also be deduced by Figure 3.20. Nonetheless, the small false positive with posterior mean 0.02 and high false negative probability with posterior mean 0.49 observed again for Cluster 2, indicate that there might be movements of users not recorded by the application.

The ability of our model to interpret the clusters by the representatives inferred, allows us to reveal interesting information about the data. The clusters formed are characterised by the density of the graphs therein, indicating the sparsity of the network population analysed. In order to explore how the networks' sparsity affects inference, we apply our model on a subsample of the Tacita users whose movements between displays are denser. One way to determine the denser networks in the sample is by utilising the inferential results from the application of our model for $C = 2$ clusters. We note here that the subsample of denser networks considered in this analysis is indicative, and serves as an example to demonstrate how inference changes when implementing our model on a sample of denser networks. In this regard, an alternative way to determine a subsample of denser networks could have been considered, for example by considering only the network observations having density above a specified threshold. The results presented in Figures 3.23, 3.24, 3.25 and 3.26, correspond to the application of our model on a subsample of 27 network observations of the original sample, determined by the inferential results for $C = 2$. Notably, we run our MCMC for 500,000 iterations with 100,000 burn-in.

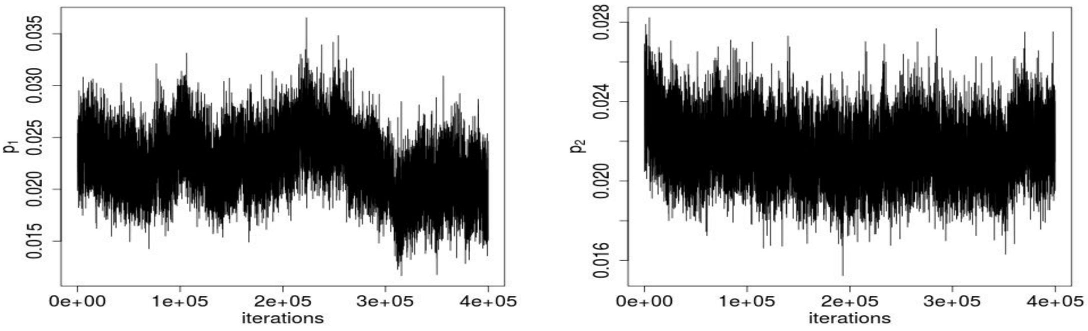

Figure 3.23: Trace plots for false positive probabilities p_c for $c = 1$ (left) and $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000, for denser network data.

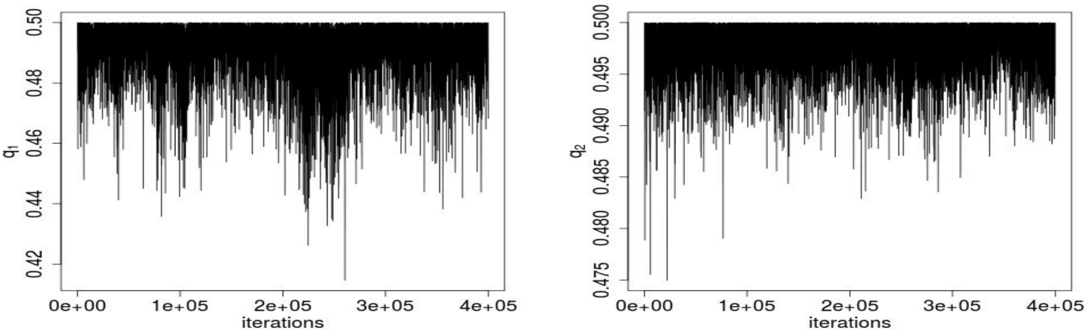

Figure 3.24: Trace plots for false negative probabilities q_c for $c = 1$ (left) and $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000, for denser network data.

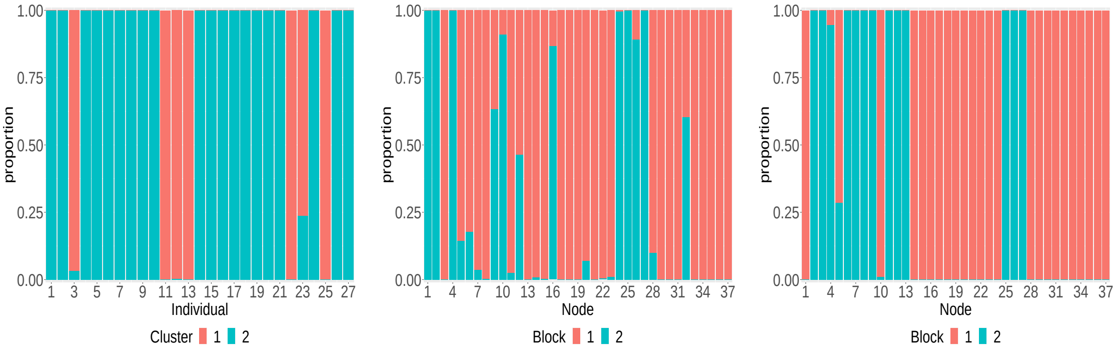

Figure 3.25: Left:Proportion of times (y axis) an individual (x axis) is allocated to each of the 2 clusters, after a burn-in of 100,000 iterations. Middle: Proportion of times (y axis) that each node (x axis) of the representative network of cluster labelled 1 is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations. Right: Proportion of times (y axis) that each node (x axis) of the representative network of cluster labelled 2 is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations.

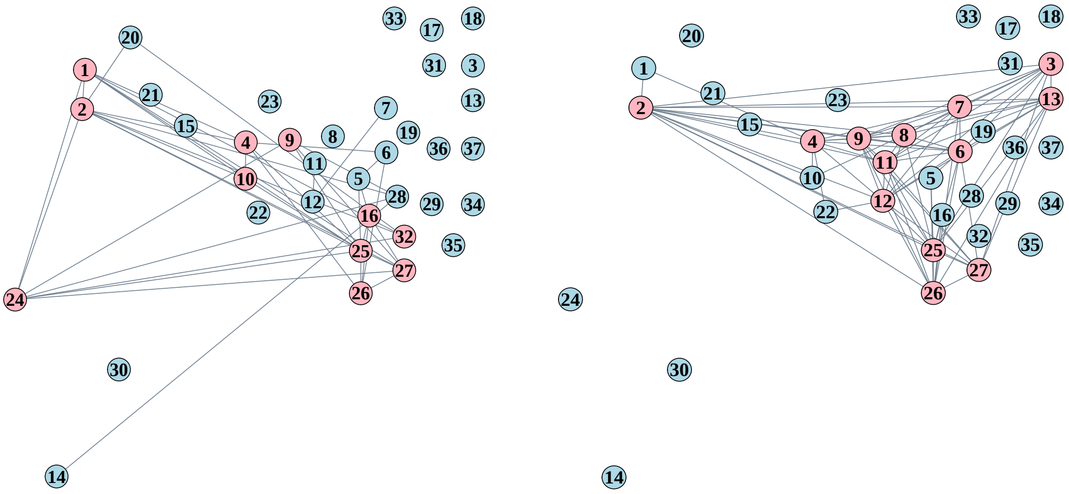

Figure 3.26: Left: Posterior mode of representative of cluster $c = 1$ with posterior mass 6%, for last 100,000 iterations, for the denser networks in Tacita population. Right: Posterior mode of representative of cluster $c = 2$ with posterior mass 8%, for last 100,000 iterations, for the denser networks in Tacita population.

We observe a stationarity issue, mainly for the chains of the false positive probability of cluster 1, p_1, and the vector of probabilities $\boldsymbol{\theta}$ of the SBM structure (Appendix A.3.1). With respect to p_1, the non-stationarity of the chain can be attributed to the small size of the sample of networks. Specifically, we observe that only 7 networks are allocated to Cluster 1, imposing challenges with respect to the inference of the component specific parameters.

Irrespectively of the denser network data considered in this example, the inference for the false positive and false negative rates of both clusters does not change much compared to the whole data set considered previously. However, we observe a meaningful change in the representatives inferred for both clusters. Notably, the representative of cluster 1 reveals movements of users mainly among the Info lab (node 2), the Furness College (nodes 7, 8, 9) and the Library (nodes 25, 26, 27), while it introduces a new finding with the Graduate College (node 24) becoming a commonly visited location as well. On the other hand, the representative of cluster 2 is very similar to the representative of cluster 2 inferred in the previous application of our algorithm on the whole data set, suggesting dense connections among 4 different locations on campus, indicatively the Infolab (2), the Furness College (nodes 7, 8, 9), the Library (nodes 25, 26, 27) and County College (13). Finally, we observe that only 26% of the network observations are allocated to Cluster 1, while the rest 74% of the networks are allocated in Cluster 2.

We further apply the outlier cluster model, to explore the inferential results under this set up. In the case of the outlier cluster model, we have a single network representing the network population, a group of networks forming the majority cluster and a group of networks forming the outlier cluster. We first apply this model on the whole data population observed for the Tacita application. We initialise the algorithm similarly to the case were we had two clusters of networks with component specific representatives and run our MCMC for 500,000 iterations with 100,000 burn-in. The results are presented in Figures 3.27, 3.28, 3.29 and 3.30.

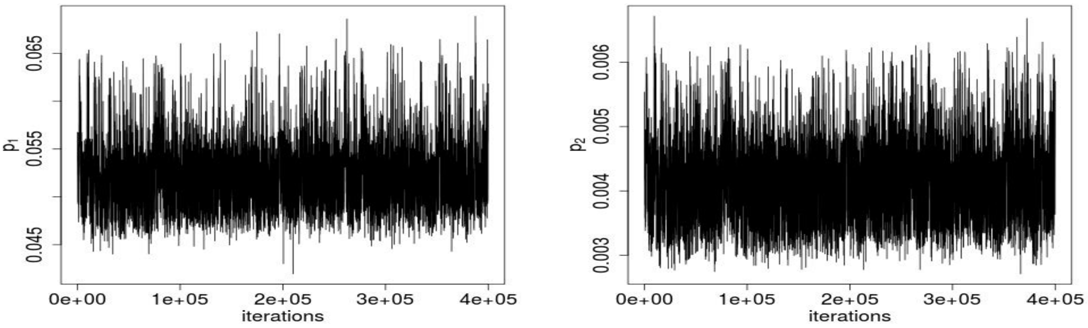

Figure 3.27: Trace plots for false positive probabilities p_c for outlier cluster labelled by $c = 1$ (left) and majority cluster labelled by $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000.

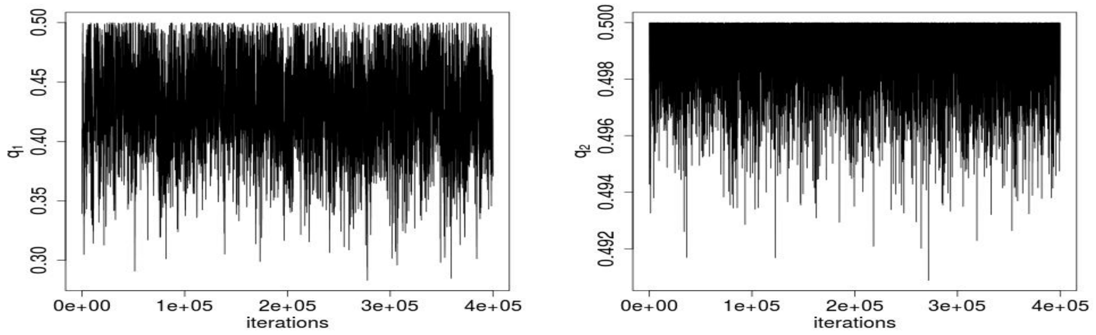

Figure 3.28: Trace plots for false negative probabilities q_c for outlier cluster labelled by $c = 1$ (left) and majority cluster labelled by $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000.

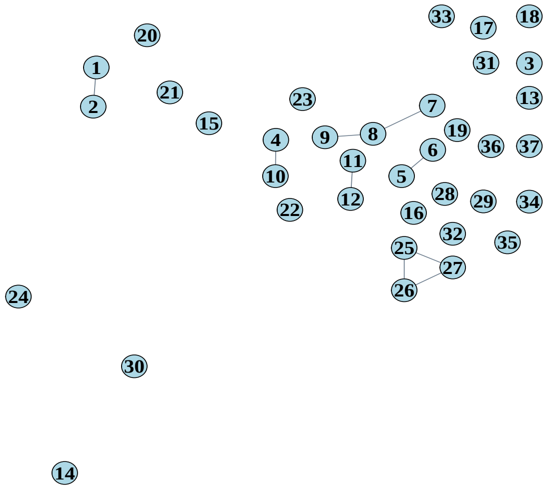

Figure 3.29: Posterior mode of representative network with posterior mass 74%, for last 100,000 iterations.

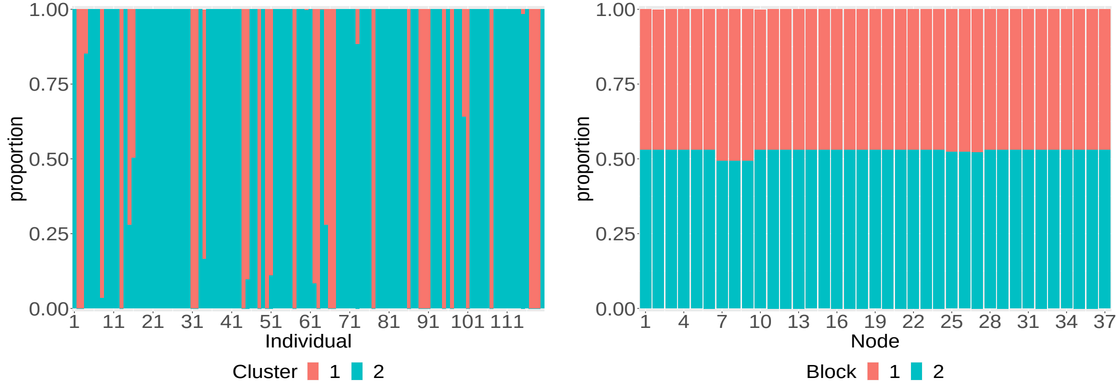

Figure 3.30: Left: Proportion of times (y axis) an individual (x axis) is allocated to each of the 2 clusters, after a burn-in of 100,000 iterations. Right: Proportion of times (y axis) that each node (x axis) of the representative network is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations.

We observe that the outlier cluster is the cluster with the denser networks, as a result the representative is highly determined by the majority cluster which encloses the sparser networks. The posterior mode of the representative concentrates a high posterior mass of 75%, however and SBM structure for the representative is not identified. We further observe good chain mixing for the false positive and false negative probabilities, and a slightly lower posterior mean for the false negative of the outlier cluster labelled by 1, which can be justified by the higher density of the networks in that cluster. We further investigate the impact of the sparsity of the network data on the results of the outlier detection, and specifically on the false negative rates, by fitting the outlier cluster model for the denser data. In Figures 3.31, 3.32, 3.33 and 3.34, we present the results for after fitting our outlier cluster algorithm to the denser network observations. We run our MCMC for 500,000 iterations with 100,000 burn-in.

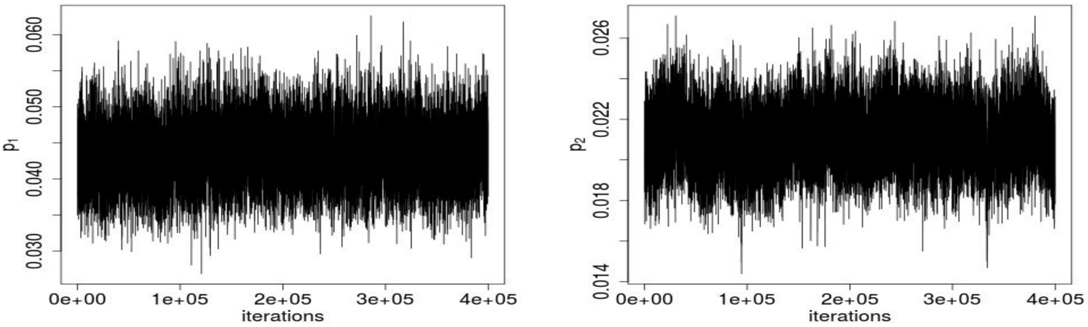

Figure 3.31: Traceplots for false positive probabilities p_c for outlier cluster labelled by $c = 1$ (left) and majority cluster labelled by $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000, for the denser networks in Tacita population.

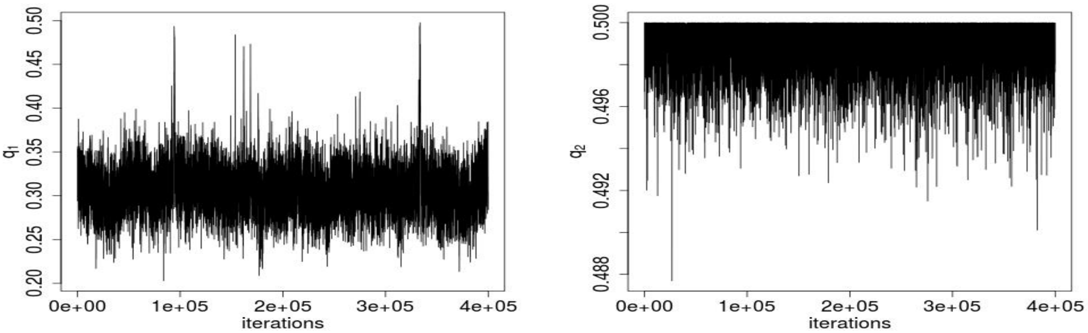

Figure 3.32: Traceplots for false negative probabilities q_c for outlier cluster labelled by $c = 1$ (left) and majority cluster labelled by $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000, for the denser networks in Tacita population.

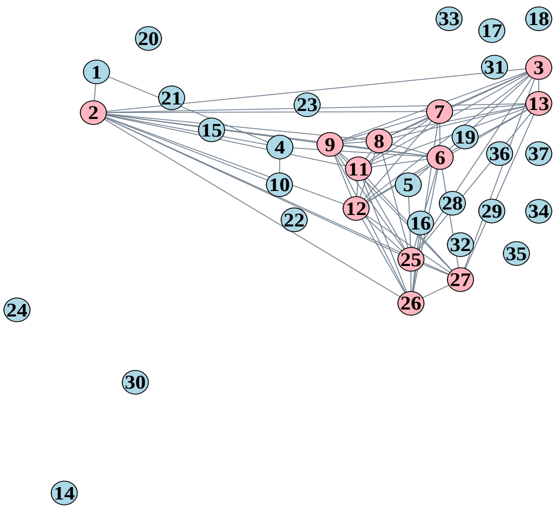

Figure 3.33: Posterior mode of representative network with posterior mass 10%, for last 100,000 iterations, for the denser networks in Tacita population.

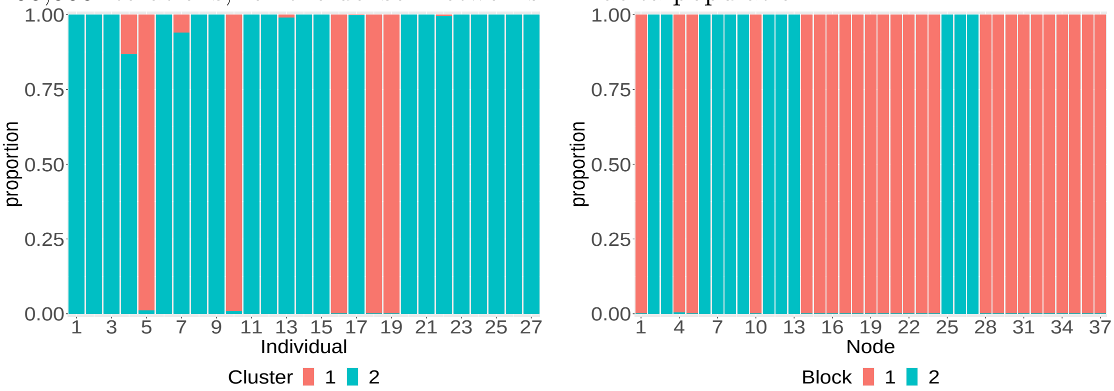

Figure 3.34: Left: Proportion of times (y axis) an individual (x axis) is allocated to each of the 2 clusters, after a burn-in of 100,000 iterations, for the denser networks in Tacita population. Right: Proportion of times (y axis) that each node (x axis) of the representative network is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations, for the denser networks in Tacita population.

The false negative probability inferred for the outlier cluster labelled by 1 (Figure 3.32) drops significantly when we consider only the denser network data for the outlier cluster model. In addition, there is a discrepancy between the region of values that are being explored for the false negative probabilities of each cluster. Specifically, for the outlier cluster labelled by 1, the algorithm explores a region of smaller values for the false negative probability, as opposed to the majority cluster labelled by 2 for which the region of values being explored for the false negative probability is larger, which can be explained by the sparser networks in that cluster. Finally, the representative has a similar block structure with those seen in the representative of the previous applications of our algorithm, with mostly connected nodes being the nodes representing the displays at the Info lab (node 2), the Furness College (nodes 7, 8, 9), the Library (nodes 25, 26, 27) and the County College (node 13).

We further investigate how the inferential results change when specifying a simpler network model for the representatives of the clusters, namely the Erdös-Rényi model. As discussed in Chapter 1, Section 1.2, the Erdös-Rényi model assumes that all edges in a network are equally probable to occur with a common probability θ. Our motivation for the Erdös-Rényi model results from the non-identified block structure for the representatives in the previous implementations of our algorithms, as seen in Figures 3.29 and 3.22. In appendix A.3.1, we present the results after specifying the Erdös-Rényi model, for both the case of $C = 2$ and the outlier cluster model, applied on the whole population of users. The inferential results for the parameters and latent variables of our model under the Erdös-Rényi model specification are similar to the results obtained with the SBM presented in this Section, suggesting that our method is consistent under any model specification for the representatives.

Lastly, we implement our clustering algorithm for $C = 3$ on the whole multiple network data set, assuming a SBM structure for the component specific network representatives. We initialise the algorithm similarly to the initialisation performed for the $C = 2$ cluster case, with sole difference that we now consider four distance metrics, namely the Hamming, the Jaccard, the l_2 and the wavelets distances, to initialise the cluster membership of the networks using majority vote. We run our MCMC for 500,000 iterations with 100,000 burn-in, and illustrate the results in Figures 3.35, 3.36, 3.37, 3.38 and 3.39.

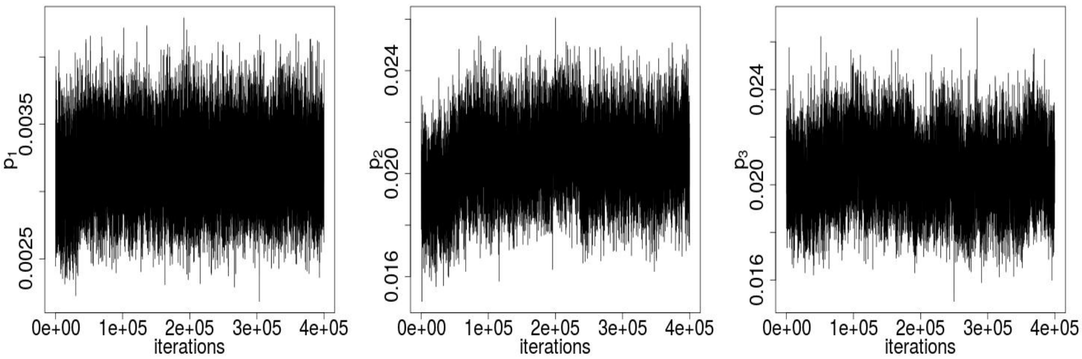

Figure 3.35: Trace plots of false positive probabilities p_c for $c = \{1, 2, 3\}$, for 400,000 iterations of the MCMC after a burn-in of 100,000 iterations.

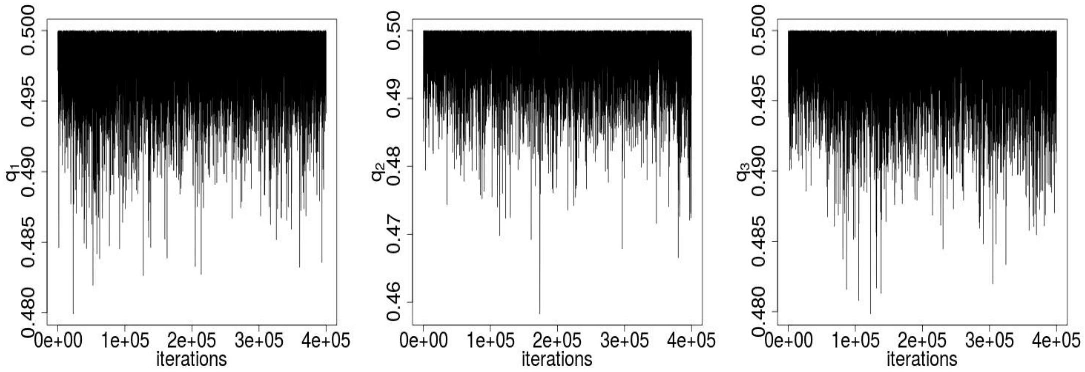

Figure 3.36: Trace plots of false negative probabilities q_c for $c = \{1, 2, 3\}$, for 400,000 iterations of the MCMC after a burn-in of 100,000 iterations.

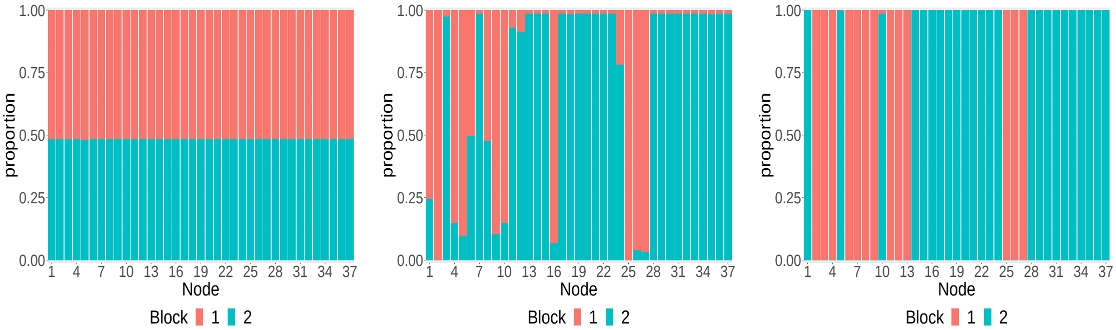

Figure 3.37: Left: Proportion of times (y axis) that each node (x axis) of the representative network of cluster labelled 1 is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations. Middle: Proportion of times (y axis) that each node (x axis) of the representative network of cluster labelled 2 is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations. Right: Proportion of times (y axis) that each node (x axis) of the representative network of cluster labelled 3 is allocated in Blocks 1 or 2, after a burn-in of 100,000 iterations.

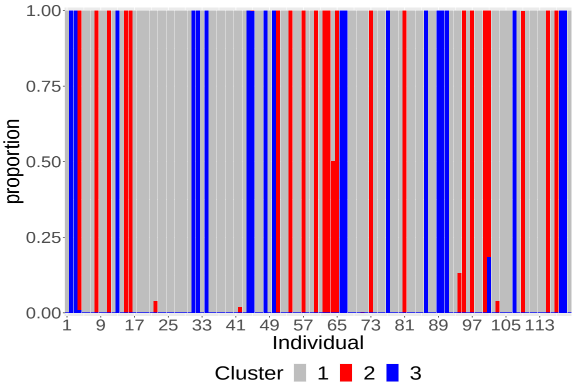

Figure 3.38: Proportion of times (y axis) an individual (x axis) is allocated to each of the 3 clusters, after a burn-in of 100,000 iterations.

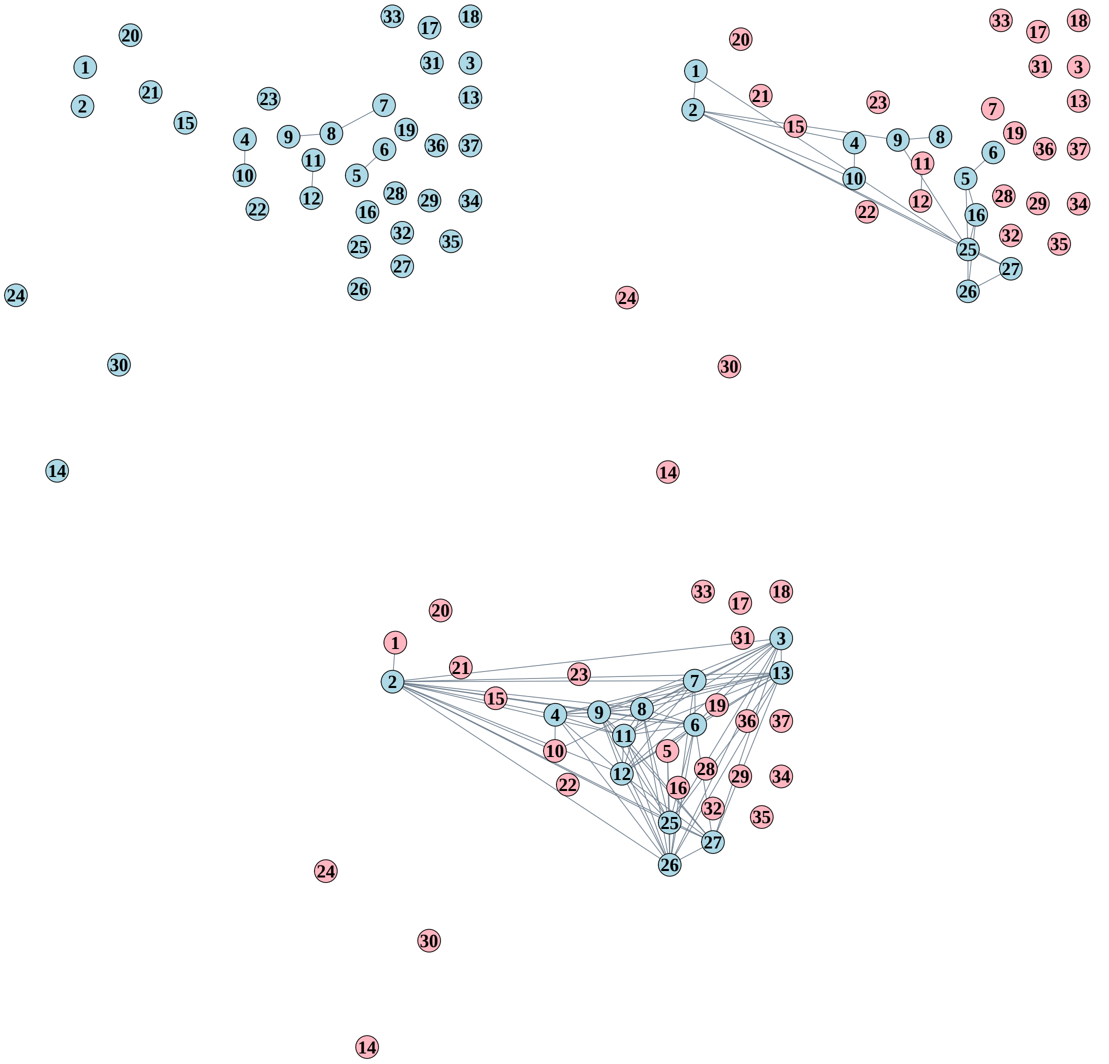

Figure 3.39: Top Left: Posterior mode for the representative of cluster labelled by 1, with posterior mass 100%, for the last 100,000 iterations. Top Right: Posterior mode for the representative of cluster labelled by 2, with posterior mass 37%, for the last 100,000 iterations. Bottom: Posterior mode for the representative of cluster labelled by 3, with posterior mass 15%, for the last 100,000 iterations.

From Figure 3.37, we observe that a block structure is not identified for the representative of cluster $c = 1$, while for the representatives of clusters 2 and 3, a similar block membership is revealed. However, we notice some differences between the block memberships of the representatives' nodes for clusters 2 and 3, namely for nodes labelled 1, 3, 5, 7, 10, 11, 12 and 13. In addition, for the representative of cluster 3, the nodes are more clearly allocated to each of the two blocks, as seen from the proportions in Figure 3.37.

As commented earlier, the high false negative probabilities inferred by our algorithm can be attributed to a misspecification of the number of clusters C. Specifically, the assumption of $C = 3$ clusters may be restrictive, thus forcing dissimilar network observations to group together, which subsequently increases the noise in each cluster. In this regard, the specification of more than 3 clusters could potentially improve the noise levels in each cluster, however, it could also lead to the formation of smaller clusters with only few observations, making inferences of cluster-specific representatives impractical. This can also be noticed from Figure 3.38 where clusters 2 and 3 contain only 17% and 18% of the network observations respectively, which observations previously formed a single cluster under case $C = 2$. Thus, the cluster with the less observations under case $C = 2$, split into two smaller clusters under case $C = 3$. Nonetheless, from Figure 3.38 we observe that most individuals are clearly allocated to one of the three clusters.

For cluster $c = 1$, we observe that the representative concentrating the whole posterior mass is sparse having only few edges and no SBM structure. We notice that the connectivity patterns of the network representative of cluster 1, are similar to the connectivity patterns of the sparser network representative obtained under the $C = 2$, in Figure 3.22, with edges connecting displays closely located to each other (e.g. edge between nodes 4 and 10 corresponding to displays both located in the same building). In addition, most of the networks in the population are allocated to this cluster with a very small false positive probability with posterior mean 0.003 and very large false negative probability with posterior mean 0.49, similarly to previous results discussed in this Section.

For cluster $c = 2$, we observe that the posterior mode of the representative network concentrates a posterior mass of 37%, while being slightly denser compared to the representative of cluster 1. Moreover, an SBM structure is revealed, with the mostly connected nodes belonging in the same block. This representative reveals a specific movement pattern of the users allocated in cluster 2, corresponding to the displays located at the central part of the campus (nodes 5, 6, 16, 25, 26, 27), as well as displays located at Infolab (nodes 1 and 2), and Furness College (4 and 10). However, there is a smaller proportion of networks allocated to cluster 2 compared to cluster 1. Furthermore, the posterior mean of the false positive probability is equal to 0.02 while the posterior mean of the false negative probability is equal to 0.49.

Lastly, for cluster $c = 3$ the posterior mode of the network representative concentrates a posterior mass equal to 15%, which is smaller compared to the posterior masses of the posterior modes obtained for the other two representatives. We notice similarities both in the block structure and the connectivity patterns of the representatives of clusters 2 and 3. However, the representative of cluster 3 is notably denser, and some new movements of individuals at displays labelled 3 and 13 (Faraday College and County College) are discovered. We also note that for cluster 3, the posterior means obtained for the false positive and false negative probabilities are similar to those of cluster 2. In addition, clusters 2 and 3 have a similar proportion of individuals. Overall, a common movement pattern is discovered for individuals in clusters 2 and 3, indicating movements among displays located in the central part of the campus.

In general, we see that under the specification of $C = 2$, the individuals previously allocated to cluster 2 with the denser network representative (Figure 3.22) are now divided into clusters 2 and 3, under the case of $C = 3$ specified. Thus, pre-defining the number of clusters to $C = 3$ leads to the discovery of a new, slightly different movement pattern of the individuals.

In conclusion, we observe that our model identifies clusters of individuals with different movement patterns, which was the main research question raised for the Tacita data. Our modelling framework allows inferences of cluster-specific representative networks, that usefully summarise information about the network observations in the corresponding cluster. As we would anticipate, most network representatives indicate movements of users between displays located at the most central part of campus, close to Alexandra Square, which is a popular area on Lancaster University campus. Nonetheless, we further observe high false negative probabilities inferred for each cluster. There are multiple ways to explain the high false negative probabilities inferred. One triggering factor is the networks' sparsity, as indicated by the lower noise levels inferred for the denser network observations under the outlier cluster model. The sparsity of the networks can be attributed to the Tacita application not recording movements of users due to weak signal or users' disengagement from the application. Another triggering factor for the high false negative noise can be the misspecification of the number of clusters C. In our analysis, we considered both the case of $C = 2$ and $C = 3$ clusters, however, the specification of more than three clusters could potentially assist in obtaining clusters with lower noise levels.

3.5.2 Brain data application

In this section we analyse a real multiple network data example, emerging from the field of Neuroscience. In this data example, the brain connectivity patterns across different regions of the brain were measured for 30 healthy individuals at resting state. For each individual, a series of 10 measurements were taken over a one-month period, using diffusion magnetic resonance imaging (dMRI). The measurements are represented as networks, with nodes corresponding to fixed regions of the brain and edges denoting the connections recorded among those regions. Specifically, the network data consist of 200 nodes (regions of the brain) according to the CC200 atlas (Craddock et al. [2012]). However, we have no information about the correspondence between the nodes' labels and the physical structure of the brain at the time of our analysis. The resulting network population consists of 300 undirected networks, corresponding to the 10 brain-scans taken for the 30 individuals.

This data have been discussed in the study of Zuo et al. [2014], Arroyo et al. [2019] and Lunagómez et al. [2021], with the latter two studies analysing the data from a modelling perspective. Specifically, Arroyo et al. [2019] investigate the ability of their method to identify differences among individuals with respect to communities formed from the networks' nodes, while Lunagómez et al. [2021], assume unimodality of the probabilistic mechanism that generates the network population, and infer a representative network for the population of individuals, according to a pre-specified distance metric. None of these approaches seek to determine and interpret clusters of networks. In this study we aim to answer the following research questions about the brain network data: First, can we detect clusters of individuals with similar brain connectivity patterns? Specifically, can we detect similarities between the brain scan measurements taken for the same individual, and cluster those accordingly? Can we obtain meaningful summaries for the clusters utilising the parameterisation of our model? Notably, can we infer network representatives for the clusters as well as make inferences for their structure assuming a Stochastic Block Model?

Before implementing our mixture model on the brain network population, we first conduct EDA to investigate information contained in the data with respect to the formation of underlying clusters. Similarly to the EDA performed for the Tacita application in Section 3.5.1, we obtain the pairwise distances among the network data, with respect to the Jaccard distance, the wavelets and the l_2 distance. Thence, we obtain a 2-d representation of the distance matrices using MDS. In Figures 3.40 and 3.41, we illustrate the MDS representations of the three distance metrics considered.

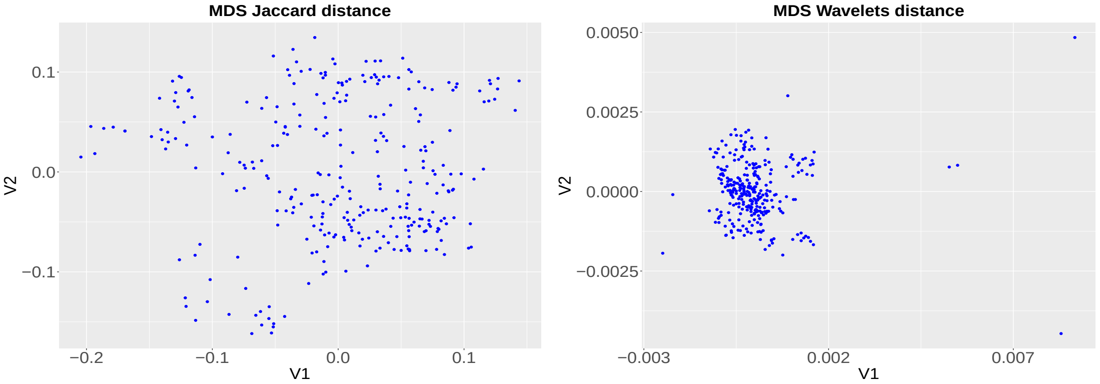

Figure 3.40: Left: MDS for Jaccard distance matrix for brain network data. Right: MDS for Wavelets distance matrix for brain network data.

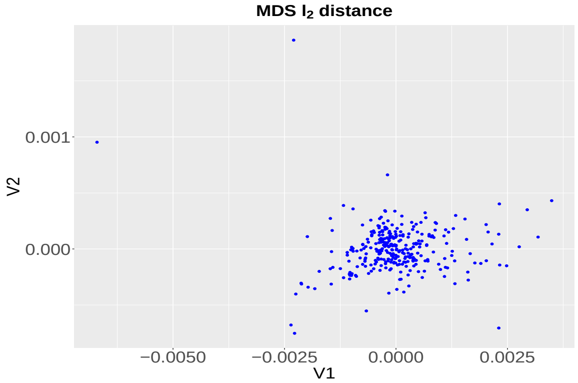

Figure 3.41: MDS for l_2 distance matrix for brain network data.

From the MDS plots, we observe that the three different distance metrics reveal different information about similarities between the brain networks, similarly to the Tacita application (Section 3.5.1). The MDS plot for the spectral distances being the wavelets and the l_2 distance, suggest that the network population forms a single cluster, with some networks being away from that cluster. In contrast, considering the MDS plots of the Jaccard distance, we observe that the network data are more scattered on the 2-d surface, potentially suggesting the formation of multiple clusters of networks.

To begin, we implement the outlier cluster detection algorithm on the brain networks, assuming a single representative that has SBM structure with $B = 2$ blocks. We initialise the algorithm in the same way we initialised the algorithm for the Tacita data (Section 3.5.1). Hence, we determine an initial membership of networks in two different clusters by implementing a k-means algorithm using the R package `kmed` (Budiaji [2019]). This is done for three distance matrices that correspond to the Jaccard, the wavelets and the l_2 distance metrics. We combine the results using majority vote, to obtain the initial cluster membership of each network. We initialise the network representative by generating its edges through independent Bernoulli draws, with probabilities equal to the proportion of times the corresponding edge is observed in the network data. We initialise the block membership of the representative's nodes by SBM estimation on the initial representative using the R package `blockmodels` (INRA and Leger [2015]). For the rest of the parameters of the model we consider three different random initialisations, and run the MCMC for each initialisation for 1,000,000 iterations.

In Figures 3.42 and 3.43, we present the traceplots of the false positive and false negative probabilities, for the majority cluster and the outlier cluster, after running our algorithm for 1,000,000 iterations under three different initialisations. We observe that under the three different initialisations the algorithm converges very quickly to the same region. This is encouraging and suggests that a high posterior region has been identified.

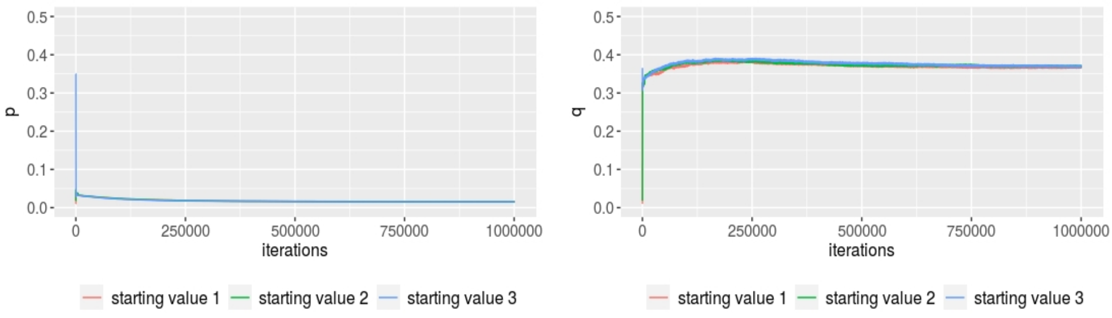

Figure 3.42: Left: Trace plot for false positive probability p for majority cluster for 1,000,000 iterations and three different initialisations. Right: Trace plot for false negative probability q for majority cluster for 1,000,000 iterations and three different initialisations.

We also compare the results of our algorithm under the three different initialisations, by obtaining the posterior mode of the representative network for the last 50,000 iterations. In Figure 3.44 (left), we obtain the posterior mode for the network representative under the first initialisation of our algorithm, with posterior mass 0.08. The colours of the nodes correspond to the block membership of the representative's nodes. In Figure 3.44 (middle), we visualise only the not in common edges between the posterior modes of the first and second initialisation to facilitate comparisons. The black edges correspond to the edges present in posterior mode of the first initialisation and not present in the posterior mode of the second initialisation, and the pink edges correspond to the

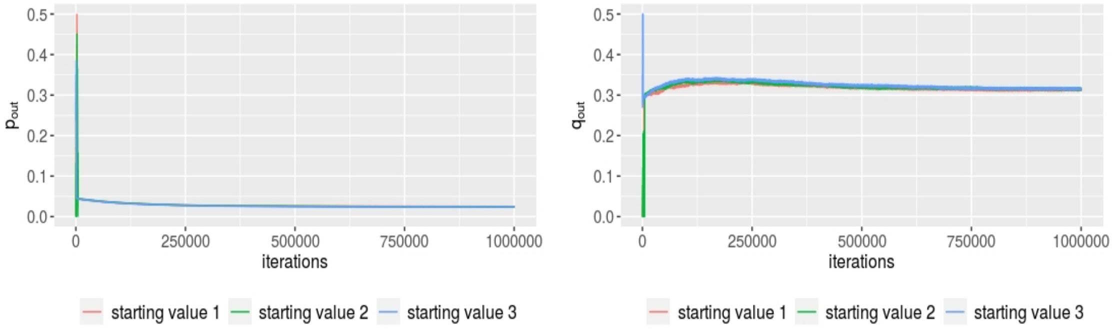

Figure 3.43: Left: Trace plot of false positive probability for outlier cluster p_{out} for 1,000,000 iterations and three different initialisations. Right: Trace plot of false negative probability for outlier cluster q_{out} for 1,000,000 iterations and three different initialisations.

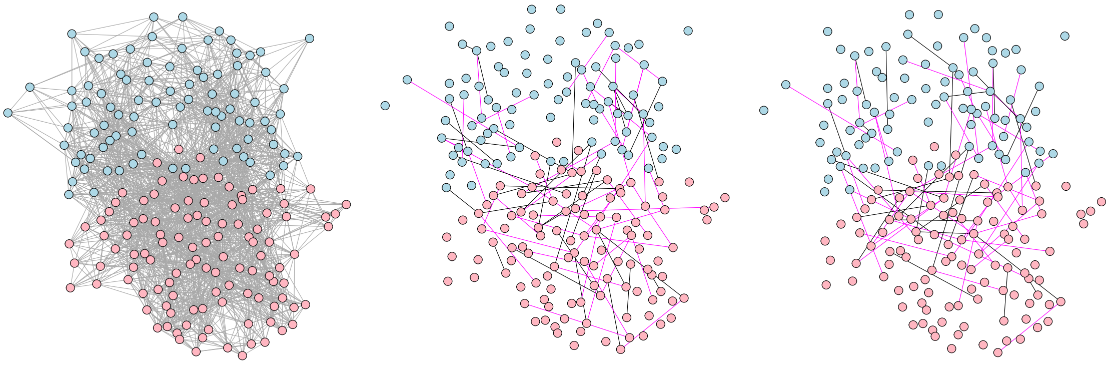

Figure 3.44: Left: Posterior mode of representative network from 1st initialisation. Middle: Network with not in common edges between posterior modes of representatives from 1st and 2nd initialisation. Right: Network with not in common edges between posterior modes of representatives from 1st and 3rd initialisation. The nodes' colours correspond to the block structure identified under each initialisation.

edges present in the posterior mode under the second initialisation and not present in the posterior mode of the first initialisation. In Figure 3.44 (right), we visualise the not in common edges between the posterior modes of the first and third initialisation. In addition, the posterior mode of the second initialisation has posterior mass of 0.08, while the posterior mode of the third initialisation has posterior mass of 0.16.

In Figure 3.45, we further illustrate the empirical marginal probability of observing an edge in the representative network inferred under each initialisation, considering the last 50,000 posterior draws. The darker colours in the matrices correspond to higher marginal probabilities, while the brighter colours correspond to smaller marginal probabilities. To illustrate the block membership structure in the matrices, we reorder the rows and columns according to the block structure inferred for the representative from the first initialisation, and keep the same order for the matrices of the representatives from the

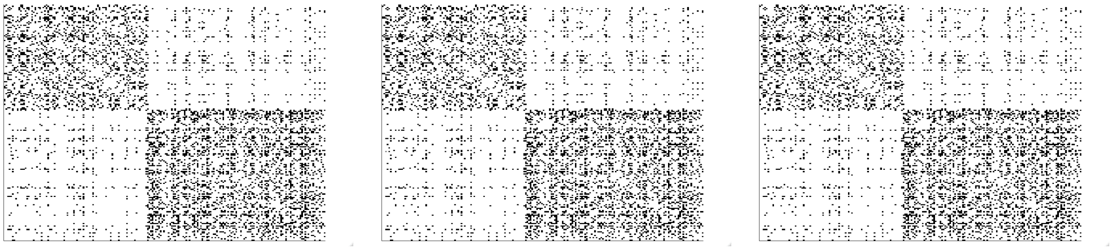

Figure 3.45: Left: Matrix of empirical marginal probability of each respective edge for representative under 1st initialisation. Middle: Matrix of empirical marginal probability of each respective edge for representative under 2nd initialisation. Right: Matrix of empirical marginal probability of each respective edge for representative under 3rd initialisation. The darker colours correspond to higher marginal probabilities.

second and third initialisation, so that the results are comparable. The matrices show the two underlying blocks in the representatives under each initialisation, with the higher marginal probabilities of an edge occurring for nodes belonging in the same block. We also notice that the matrices obtained under all three different initialisation are very similar.

There are three interesting findings with respect to the representative inferred under the three different initialisation. First, there is only a small proportion of edges not in common among the posterior modes under the three different initialisations, considering the density of the graphs. Second, our algorithm infers the same block structure for the three posterior modes of the representative networks. Third, the posterior masses concentrated by the posterior mode representatives are small, as anticipated due to the high dimensional space of the networks.

We further observe a smaller false negative rate inferred for the outlier cluster compared to the majority cluster, suggesting that the edges not observed in the network data for the outlier cluster are more likely to be correct compared to the majority cluster. This finding also suggests that the network data in the outlier cluster are sparser compared to the majority cluster. Also, the small false positive probabilities for both clusters indicate that the edges observed in the network data are likely correct.

In Figure 3.46, we present the results for the cluster membership z of the network observations for the first initialisation. Specifically, we calculate the proportion of times that a network observation is allocated to the majority or the outlier cluster, labelled by 1 and 2 respectively, for the last 100,000 iterations. Each subfigure in Figure 3.46, shows the cluster allocation of the multiple brain scans obtained for the same individual. We see that our model mostly allocates scans of the same individual to the same cluster. Thus, our model detects similarities among the brain scans of the same individual, giving credence to our model clustering the networks sensibly. From Figures 3.47 and 3.48, we further notice that the same cluster membership is inferred for the networks under the second and third initialisation of the algorithm. In addition, the posterior means for the cluster weights τ_1 and τ_2 after a burn-in of 200,000 iterations of our MCMC are 0.62 and 0.38 respectively, under all three initialisations.

This is a common finding with Arroyo et al. [2019] who performed semi-supervised classification on the brain network population. However, our approach is different to Arroyo et al. [2019] in two ways. First, we implement an unsupervised method to infer underlying clusters of networks. We only pre-define the number of cluster in the population. Second, the interpretation of the results of our model-based clustering method compared to Arroyo et al. [2019] differs significantly, as our model reveals a cluster of individuals whose brain connectivity patterns are different to a majority group, and are interpreted through a parametric model.

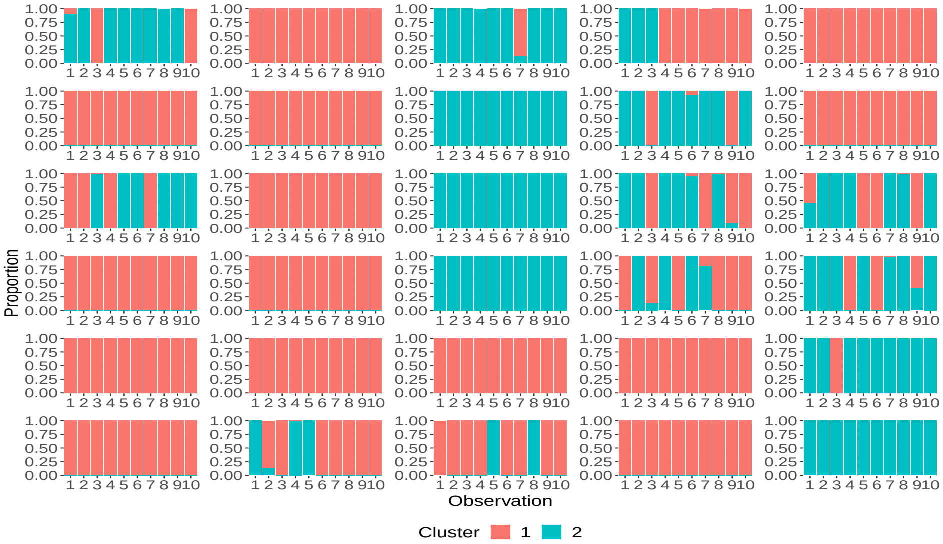

Figure 3.46: Proportion of times that each of the 10 brain scans taken for the 30 individuals is allocated in cluster labelled 1 and/or 2, under the first initialisation.

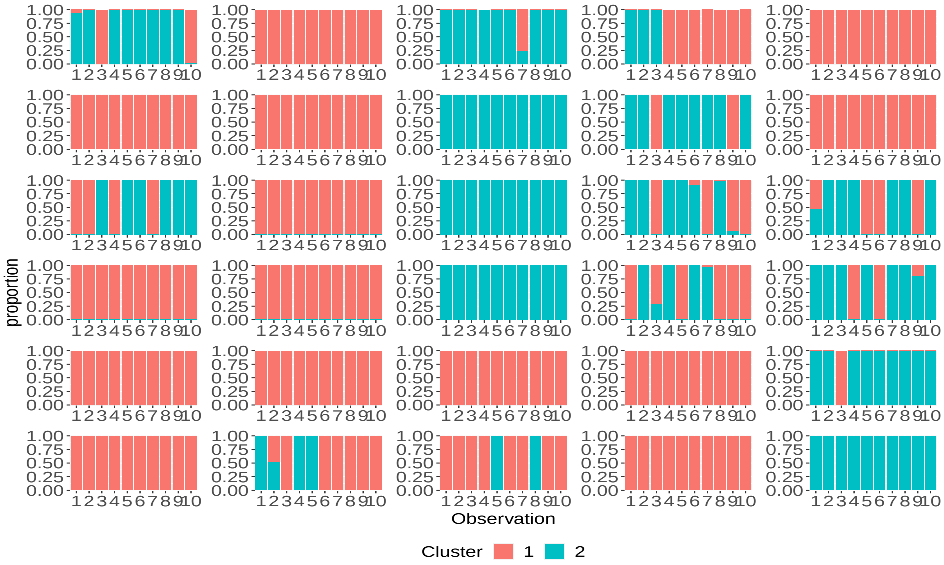

Figure 3.47: Proportion of times that each of the 10 brain scans taken for the 30 individuals is allocated in cluster labelled 1 and/or 2, under the second initialisation.

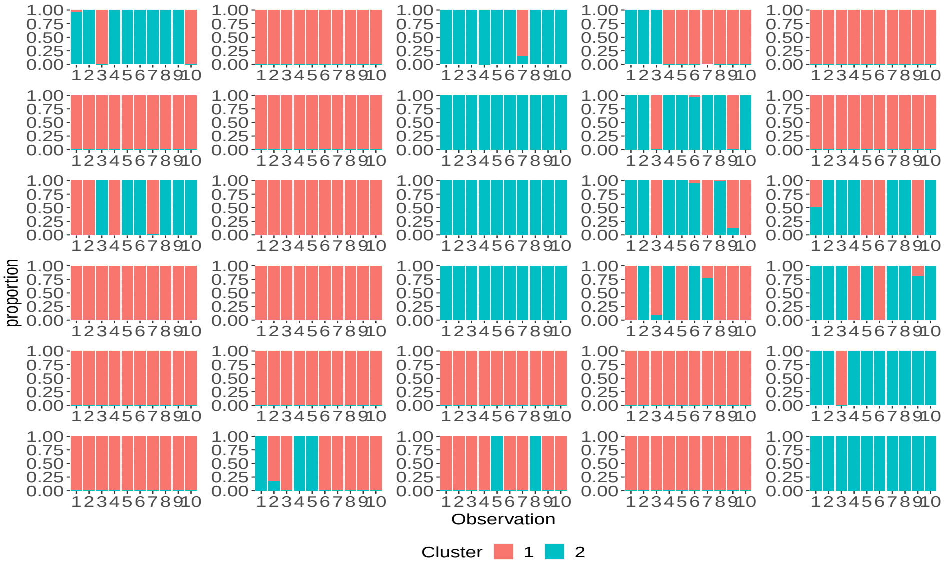

Figure 3.48: Proportion of times that each of the 10 brain scans taken for the 30 individuals is allocated in cluster labelled 1 and/or 2, under the third initialisation.

We now implement our cluster detection algorithm, assuming $C = 2$ clusters of networks, with component specific network representatives having SBM structure with $B = 2$ blocks. Considering the dimensionality of the problem, inferring two 200-node network representatives is a very challenging task. Challenges arise not only from the dimensionality of the space of graphs with 200 nodes, but also from the association between the sample size N and the networks' size n.

We initialise our algorithm similarly to the outlier cluster detection algorithm presented in this Section, and run the MCMC for 1,000,000, under the three different initialisations. The trace plots for the false positive and false negative probabilities, for clusters labelled 1 and 2, are presented in Figures 3.49 and 3.50, respectively.

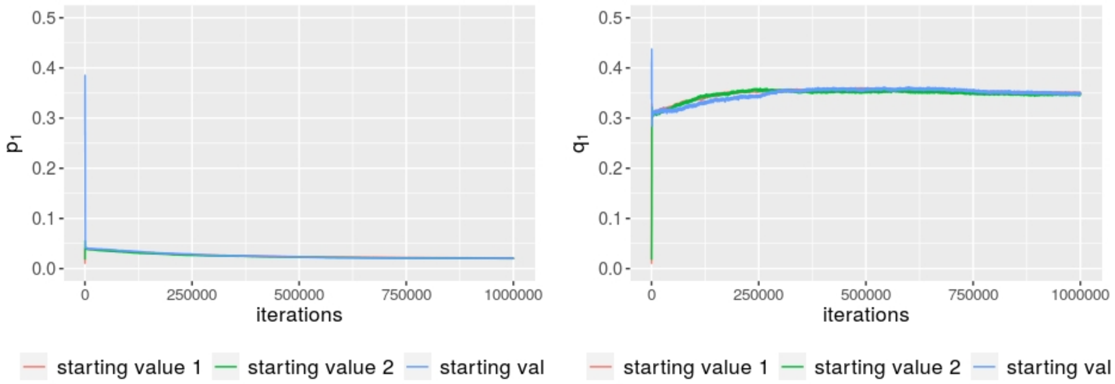

Figure 3.49: Left: Trace plot for false positive probability p_1 for cluster labelled by 1, for 1,000,000 iterations and three different initialisations. Right: Trace plot for false negative probability q_1 for cluster labelled by 1, for 1,000,000 iterations and three different initialisations.

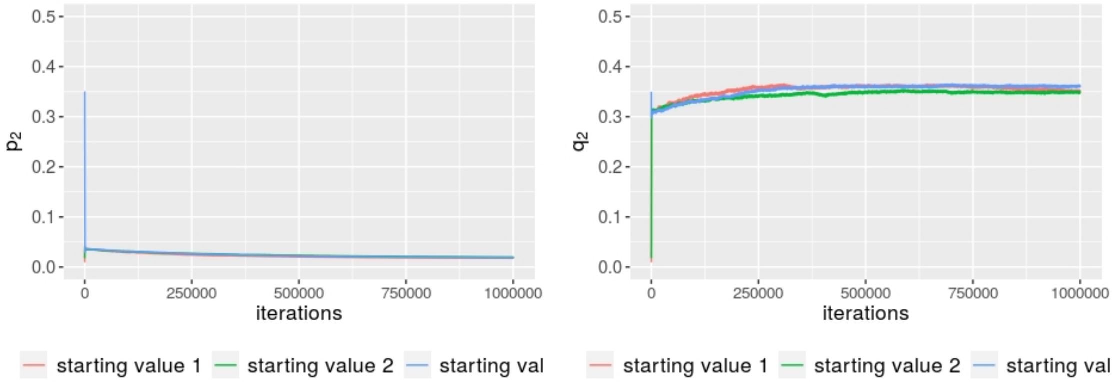

Figure 3.50: Left: Trace plot for false positive probability p_2 for cluster labelled by 2, for 1,000,000 iterations and three different initialisations. Right: Trace plot for false negative probability q_2 for cluster labelled by 2, for 1,000,000 iterations and three different initialisations.

Similarly to the results obtained from the outlier cluster detection, we observe that the algorithm converges to the same region of values under the three initialisations, for both the false positive and false negative probabilities. However, we note a stationarity issue of the chains, suggesting that potentially more iterations of our MCMC are required. Again, we notice that the false positive probabilities inferred are significantly lower than the false negative probabilities, for both clusters. As discussed in the previous Section 3.5.1 for the Tacita data, this finding may suggest that the assumption of $C = 2$ clusters may be restrictive for the data. This result could also potentially suggest that there are edges not recorded in the brain networks leading to an increase of the false negative rates. Constructing networks from brain images may result in false recording of edges and non-edges, due to the use of thresholds for determining the presence or absence of edges between brain regions (Prajapati and Emerson [2020]).

In Figures 3.51 (left) and 3.53 (left), we visualise the posterior modes for the network representatives of clusters 1 and 2 respectively, while in Figures 3.51 (middle and right) and 3.53 (middle and right), we visualise networks with the not in common edges between the posterior modes from each initialisation, similarly to the visualisation obtained for the outlier detection for the brain data presented earlier. The posterior modes for the network representatives of each cluster under the three different initialisations vary significantly, as can be seen from the not in common edges visualised in the middle and right networks of Figures 3.51 and 3.53. The results indicate the challenging nature of the problem, as anticipated. However, the block structure inferred for the network representative of each cluster remains largely stable under all three initialisations, as can be noticed from the coloured nodes of the posterior modes in Figures 3.51 and 3.53. This can also be noticed from Figures 3.52 and 3.54 that illustrate the empirical marginal probabilities of edges in representatives of cluster 1 and 2 respectively, similarly to the visualisation obtained for the representative of the outlier cluster model on the brain data, which illustrate that the probability of observing an edge is higher for nodes within the same block.

In Figures 3.55, 3.56 and 3.57, we present the cluster membership of the networks inferred from each initialisation of the algorithm. The figures show that our model tends to allocate brain-scans of the same individual, to the same cluster, however, the results are not very consistent under the three different initialisations. Specifically, the posterior means for the cluster weights τ_1 and τ_2 after a burn-in of 200,000 iterations of our MCMC, are 0.61 and 0.39 respectively under the first initialisation, 0.54 and 0.46 respectively under the second initialisation and 0.59 and 0.41 respectively under the third initialisation.

In summary, we observe that both the implementation of our model for $C = 2$ clusters as well as the implementation of our outlier cluster model on the brain networks, reveal a sensible clustering of the network observations as can be deduced from the results showing that brain scans of the same individual are mostly allocated in the same cluster. In addition, for the outlier cluster model, we observe that the clustering of the

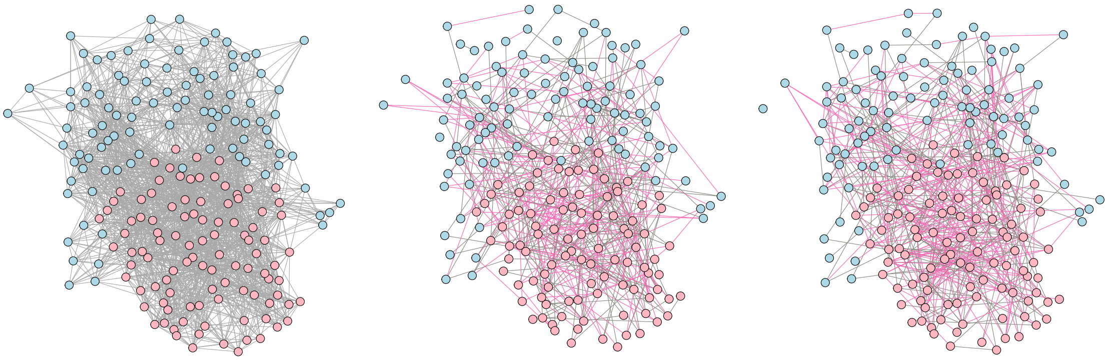

Figure 3.51: Left: Posterior mode of representative network of cluster 1 from 1$^{\text{st}}$ initialisation. Middle: Network with not in common edges between posterior modes of representatives of cluster 1 from 1$^{\text{st}}$ and 2$^{\text{nd}}$ initialisation. Right: Network with not in common edges between posterior modes of representatives of cluster 1 from 1$^{\text{st}}$ and 3$^{\text{rd}}$ initialisation. The nodes' colours correspond to the block structure identified under each initialisation.

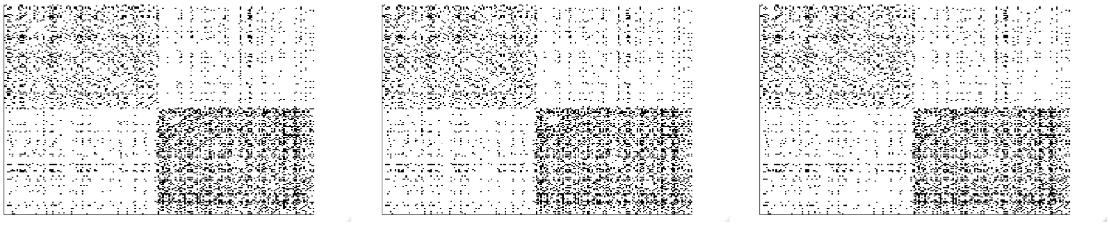

Figure 3.52: Left: Matrix of empirical marginal probability of each respective edge for representative of cluster 1 under 1$^{\text{st}}$ initialisation. Middle: Matrix of empirical marginal probability of each respective edge for representative of cluster 1 under 2$^{\text{nd}}$ initialisation. Right: Matrix of empirical marginal probability of each respective edge for representative of cluster 1 under 3$^{\text{rd}}$ initialisation. The darker colours correspond to higher marginal probabilities.

networks is consistent under all three initialisations, however, this is not the case with the two-cluster model. The unstable clustering results of the $C = 2$ cluster model can be explained by the challenging nature of inferring two 200-node network representatives (one for each cluster) considering the size of the sample $N = 300$ networks. This can also be noticed from the posterior modes obtained for the network representatives of the clusters under the $C = 2$ cluster model, which are also unstable under the three different initialisations. Nonetheless, inferring a single network representative under the outlier cluster model is a feasible task as can be noticed from the stability of the results under the three initialisations, indicating that we can obtain a meaningful summary for the data. Lastly, it is encouraging that the traceplots of the false positive and false negative probabilities explore the same region of values under all three initialisations, however, high false negative rates may suggest the presence of more than two clusters in the data.

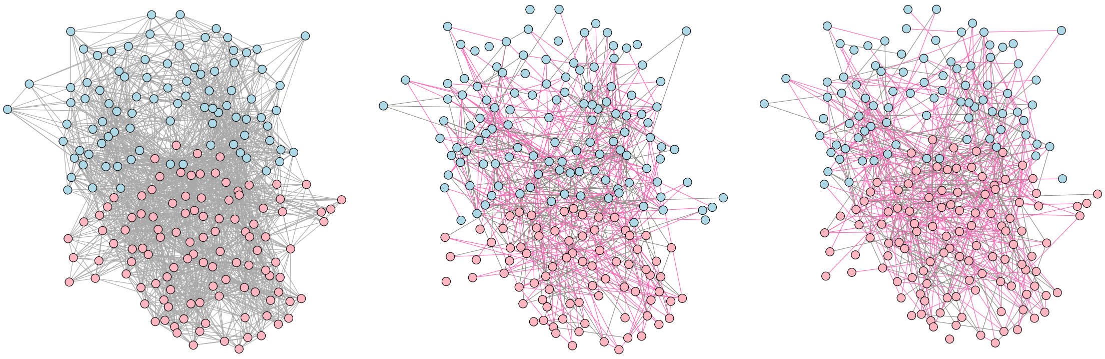

Figure 3.53: Left: Posterior mode of representative network of cluster 2 from 1st initialisation. Middle: Network with not in common edges between posterior modes of representatives of cluster 2 from 1st and 2nd initialisation. Right: Network with not in common edges between posterior modes of representatives of cluster 2 from 1st and 3rd initialisation. The nodes' colours correspond to the block structure identified under each initialisation.

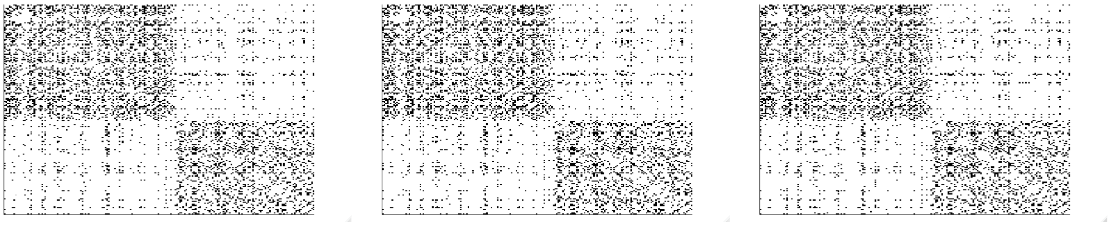

Figure 3.54: Left: Matrix of empirical marginal probability of each respective edge for representative of cluster 2 under 1st initialisation. Middle: Matrix of empirical marginal probability of each respective edge for representative of cluster 2 under 2nd initialisation. Right: Matrix of empirical marginal probability of each respective edge for representative of cluster 2 under 3rd initialisation. The darker colours correspond to higher marginal probabilities.

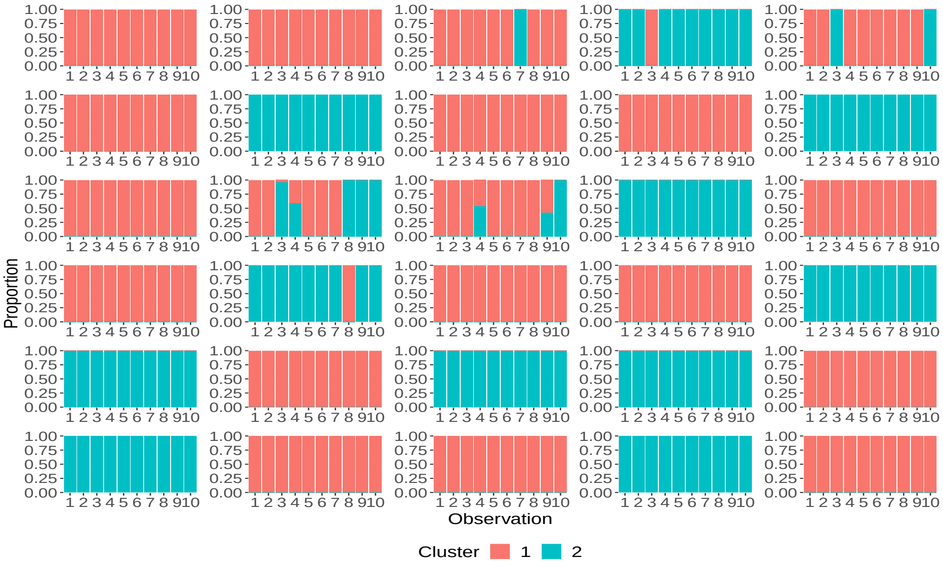

Figure 3.55: Proportion of times that each of the 10 brain scans taken for the 30 individuals is allocated in cluster labelled 1 and/or 2, under the first initialisation.

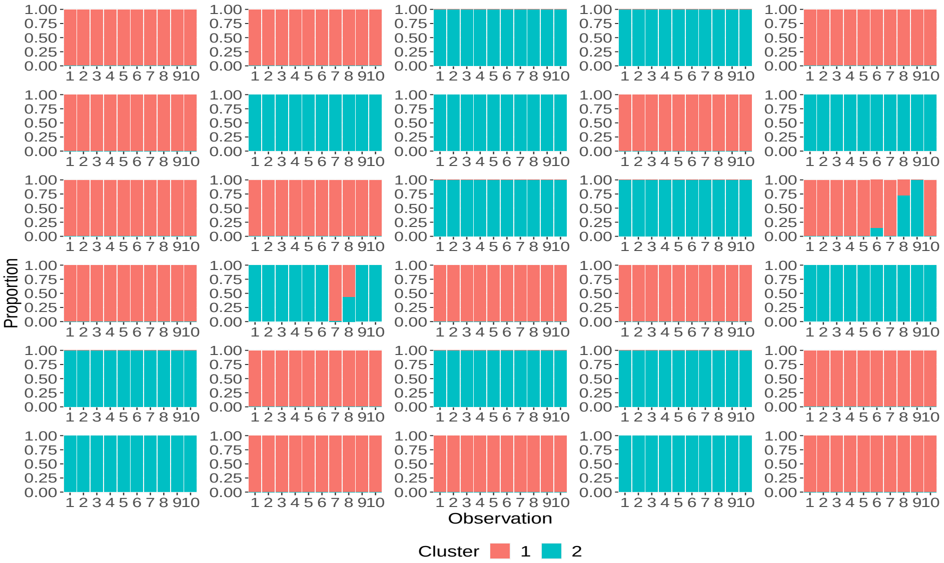

Figure 3.56: Proportion of times that each of the 10 brain scans taken for the 30 individuals is allocated in cluster labelled 1 and/or 2, under the second initialisation.

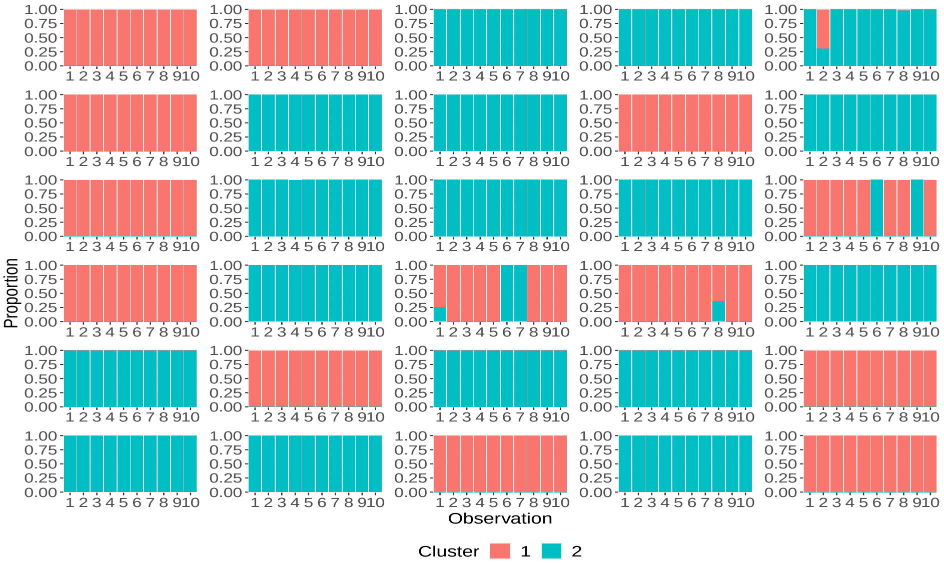

Figure 3.57: Proportion of times that each of the 10 brain scans taken for the 30 individuals is allocated in cluster labelled 1 and/or 2, under the third initialisation.

3.6 Discussion

In this Chapter we introduced a mixture model for multiple network data that allows us to identify clusters of networks in a population. To achieve this, we formulated a mixture of measurement error models and developed a Bayesian framework that allows us to make inferences for all model parameters jointly. This framework permits a diverse specification of the network model for the representative networks in the population, determined according to the type of information we want to exploit based on the data.

We examined the performance of our model in both the task of clustering network data and inferring the model parameters, through extensive simulation studies. Notably, for moderate-sized networks our algorithm successfully inferred the model parameters and accurately clustered the network data, even for regimes with high noise levels. This is an interesting result, as for high noise levels there is great variability in the structure of the simulated network population, making inference a challenging task. The results suggested that our model can perform well for a range of real data applications.

In Section 3.4.2, we further examined the model performance for large network and population sizes. This had not been explored by Signorelli and Wit [2020] who also develop a model-based approach for clustering multiple network data. The simulations performed under this setting have led to two interesting findings. First, we observed that the absolute errors of the posterior means for the parameters were small, even for large network sizes and a relatively small sample size. This suggests that our algorithm does not require a large number of network observations to accurately infer the model parameters. Second, the representative networks for each cluster were inferred with good accuracy, for network sizes of up to 75 nodes. Thence, in contrast to Signorelli and Wit [2020] who infer only scalar parameters for interpreting the clusters detected, we are able to infer network representatives characterising the network data of the cluster with good accuracy. However, we note that for the 100-node case the network representatives are not closely identified. This is not surprising considering the size of the space of 100-node graphs.

The flexibility of our modelling framework in answering diverse research questions was demonstrated through two real-world applications. The clustering performed on the Tacita application revealed different movement patters of the users. This can be deduced by the network representatives inferred for the clusters, which are primarily characterised by their density. The clusters characterised by the sparser representative, reveal movements among closely located displays on campus. This can be interpreted in two ways: (i) the Tacita application fails to record movements of the individuals due to weak WiFi connection, as a result only movements between displays at close proximity are recorded (ii) the users are not engaged to the Tacita application. The clusters represented by the denser networks, reveal a specific movement patterns of the individuals. As the majority of the networks are very sparse it was encouraging to see that our model was able to separate out the denser networks and further distinguish different clusters among this subset of individuals.

The analysis of the brain network data led to some interesting findings. First, the results from both the cluster detection and the outlier cluster detection, suggest that we identified a high posterior region for the false positive and false negative probabilities of each cluster. Second, for the outlier cluster detection performed, a similar posterior mode for the network representative was inferred under the three different initialisations. These are especially encouraging given the high dimensional space spanned by 200-node network representatives. Lunagómez et al. [2021] who also obtain a network representative in terms of a Fréchet mean for the same brain network population, resorted to divide-and-conquer methods to be able to make inferences, while Arroyo et al. [2019] do not characterise the population via a parameterisation, and specifically via a network representation.

Our model could potentially incorporate covariates at the node or edge level to inform the inference. In some network applications, it is common to have additional information about the nodes or the edges of the network. For example, the Tacita mobile application also records the type of content shown by the display at the time visited by the user. The incorporation of this additional information could potentially lead to interesting additional findings.

For outlier cluster detection, although we assume one majority and one outlier cluster, our framework can be more flexible. We could assume the presence of two or more clusters of outlier networks that could describe different levels of variability in the data. However, the data may not always support such inferences.

An interesting result from both the Tacita and the brain networks application, is the high false negative probabilities inferred for the clusters. Notably, for the Tacita application this finding suggests that there are edges in the network data not recorded by the application, which is a meaningful result considering that the application might fail to record a user due to weak signal, or due to inactive users. We further investigated whether the sparsity of the network data results in high false negative probabilities by applying our algorithm on the denser network data of the Tacita population. From this application we noticed that lower false negative probabilities were inferred, indicating that network data sparsity results in higher false negative rates.

Network sparsity is a common issue in many real world network applications. One way to deal with the sparsity would be to consider shrinkage priors e.g. formulating the Horseshoe priors (Carvalho et al. [2009]). Another way to account for the networks' sparsity, is to assume that the networks are partially observed. In the literature, partially observed networks have been considered under two different perspectives. The coarsening approach focuses on incorporating the coarsening mechanism, that allows us to only partially observe the networks in the model, and efficiently impute the partially observed data thereafter (Heitjan and Rubin [1991], Handcock and Gile [2010], Heitjan and Rubin [1990], Kim and Hong [2012]). Another approach focuses on the missingness of certain edges and performs edge prediction (Koskinen et al. [2013], Marchette and Hohman [2015], Zhao et al. [2017], Airoldi and Blocker [2013]). Under the first approach, one way we could incorporate the coarsening mechanism is through the assumption of a sampling design, while under the second approach we could have a two-stage method which would first involve performing link prediction, and second performing inference. All the aforementioned approaches require significant modifications of our model and present interesting avenues for future research.

Finally, a natural extension of our model, is to allow simultaneous inference for the number of clusters as well. To achieve this, methods in the literature such as the reversible jump MCMC (Richardson and Green [1997]) and Dirichlet Process (DP) mixture models (Neal [2000]) could be adopted. Reversible jump MCMC would be particularly challenging due to the MCMC moving between very different dimensions to make inferences. DP mixture models would require a different MCMC scheme but present an interesting possibility to consider.

Chapter 4

Bayesian Inference for Models in the Spherical Network Family (SNF) using Importance Sampling (IS)

4.1 Introduction

In this Chapter we return to the Spherical Network Family (SNF) of models presented earlier in Chapter 2, Section 2.2.1 of this thesis. The SNF model was introduced by Lunagómez et al. [2021] as a model for capturing the probabilistic mechanism that generates a network population. The key idea behind the modelling approach proposed by Lunagómez et al. [2021], is the presence of a network representative playing the role of the mean in a network population, and a dispersion parameter describing the concentration of the observed network data around the network representative, with respect to a distance metric specified by the statistician. Under this assumption, the likelihood of observing a population of graphs $\mathcal{G}_1, \cdots, \mathcal{G}_N$ is

$$P(A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N} \mid A_{\mathcal{G}^m}, \gamma) \propto \exp\left\{-\gamma \cdot \sum_{i=1}^{N} \phi(d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}))\right\},$$

where $\mathcal{G}^m$ is the network representative, $d_G(\cdot, \cdot)$ is a distance metric, γ is the dispersion with $\gamma > 0$ and $\phi(\cdot) > 0$ is a monotone increasing function. The normalising constant of this model is the reciprocal of

$$Z(A_{\mathcal{G}^m}, \gamma) = \sum_{A_{\mathcal{G}} \in \{\mathcal{G}_{|n|}\}} \exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}}, A_{\mathcal{G}^m}))\},$$

which involves the sum over the space of n-node graphs, denoted by $\{\mathcal{G}_{|n|}\}$. The number of graphs that a space of n-node undirected graphs encloses is equal to $2^{\frac{n(n-1)}{2}}$, thus, as the network size n increases the corresponding space of graphs increases drastically, making the calculation of the normalising constant computationally infeasible even for small network sizes. Several approximation techniques have been proposed for tackling the problem of an intractable normalising constant (Chen and Shao [1997], Meng and Wong [1996], Gelman and Meng [1994], Murray et al. [2012], Alquier et al. [2016]). Lunagómez et al. [2021] circumvent the problem of the double-intractable normalising constant for the SNF model using the Auxiliary Variable method presented by Møller et al. [2006]. More details about the implementation of the Auxiliary Variable technique for the SNF model are discussed in the following Section 4.2.1. However, for some network distance metrics, the Auxiliary Variable method provides poor mixing results. In our work, we propose the use of an alternative method to the Auxiliary Variable method, namely an Importance Sampler (Chen and Shao [1997]), as it allows better mixing of the MCMC chains under different distance metric specifications in the SNF model. In addition, we introduce a new network distance metric that measures dissimilarities between networks with respect to their cycles, as cycles reveal information about network topology (Maugis et al. [2017], Fan et al. [2019]), and is a motif of interest in many applications (Sokhn et al. [2012], Koutrouli et al. [2020], Sizemore et al. [2018], Han et al. [2017]) .

A multiple network data set that motivated our work is a collection of ecological networks representing aggressive interactions between species of fish, at multiple reefs in the Indo-Pacific ocean. From an ecological perspective, it is of interest to identify competitive behaviours among species of fish in the observed network data (Keith et al. [2018]). One way to make inferences about the population of fish networks is to formulate the SNF model that allows us to determine a distance metric, according to the type of information that we want to capture. In light of this, we elicited a network distance metric that captures information about the competitive behaviours between species of fish, namely the Hamming-Symmetric difference (HS) distance metric, which involves the symmetric difference between the networks' cycles. However, making inferences using the SNF model with our proposed HS distance metric is challenging due to the presence of the intractable normalising constant.

Among the most widely used approaches for addressing the problem of the intractable normalising constant, is the Importance Sampling (IS) approximation technique. In broad terms, IS is a method that uses random samples from a proposal density, instead of sampling directly from the distribution of interest. In the context of intractable distributions, IS and generalisations of it, namely the Bridge Sampling, the Path Sampling and the Sequential Monte Carlo Samplers (Chen and Shao [1997], Gelman and Meng [1994], Everitt et al. [2017a], Everitt et al. [2017b]), are used to estimate either the normalising constant of the distribution of interest directly, or the ratio of two normalising constants, or the evidence and the Bayes Factor (BF) of two models under comparison. On the other hand, exact sampling methods involve the use of auxiliary variables in a Metropolis-Hastings algorithm that allow the cancellation of the normalising constants from the MH ratio, as seen in Møller et al. [2006] and Murray et al. [2012]. For a more descriptive review on the approaches developed for approximating normalising constants, refer to Section 4.2.2.

Lunagómez et al. [2021] formulate a Metropolis-Hastings algorithm to infer the parameters of the SNF model, and apply the Auxiliary Variable technique presented in Møller et al. [2006], to circumvent the problem of the double-intractable normalising constant. However, the implementation of Moller's technique in the case of the SNF model encompasses the following limitations:

L1 The mixing of the MCMC chains is poor, thus an extended number of iterations is required for the chains to converge.

L2 For some distance metrics, including the HS distance metric elicited for the fish networks, the implementation of the Auxiliary Variable technique for the SNF model leads to inferences that do not encode any uncertainty for the parameters under estimation.

L3 The implementation of Moller's technique as seen in Lunagómez et al. [2021] requires sampling from the SNF model, which can be computationally challenging for some distance metrics.

These limitations led us to consider an alternative method to overcome the problem of the intractable normalising constant, inspired by Vitelli et al. [2017] who develop a Bayesian inference scheme for the Mallow's model (Mallows [1957]). The Mallow's model is a widely used model for analysing rank data, that has the same functional form with the SNF model. However, a key difference between the Mallow's model and the SNF model is that the normalising constant in the latter involves both the representative network and the dispersion parameter, while for the Mallow's model the normalising constant depends only on the dispersion parameter, for right-invariant distance metrics considered in Vitelli et al. [2017].

This attribute of the Mallow's model allows an off-line approximation of the normalising constant by implementing an IS scheme, using a pseudo-likelihood approximation of the target distribution. On the other hand, in our work we formulate an Importance Sampler within our MCMC algorithm, under a different specification of the IS density for sampling network data, that allows a good approximation of the normalising constant for the SNF model.

The contributions of this Chapter can be summarised as follows. First, our MCMC scheme with IS step allows better mixing of the MCMC chains in less iterations of the algorithm, compared to the Auxiliary Variable technique (Møller et al. [2006]). Second, we develop a new network distance metric, namely the HS distance, that has not been considered in the network literature, motivated by the ecological application. Third, our MCMC scheme works under distance metric specifications, such as the HS distance metric, for which the Auxiliary Variable technique showed poor performance in identifying a posterior distribution for the model parameters.

The remainder of this Chapter is organised as follows. In Section 4.2, we provide the background to our proposed MCMC with IS step. In Section 4.3, we introduce the HS distance metric, along with the motivation for formulating it, and we present our proposed MCMC with IS step algorithm for inferring the parameters of the SNF model. In addition in Section 4.3, we discuss some challenges arising with the formulation of the HS distance and with the implementation of an IS step within the MCMC. In Section 4.4, we present the simulation studies conducted to explore the behaviour of the HS distance and assess the performance of the MCMC with IS step. In Section 4.5, we apply our MCMC with IS step for the SNF model with the HS distance metric on the ecological application that motivated our study. Lastly, in Section 4.6, we conclude with a discussion on the methods presented in this Chapter and potential future work directions.

4.2 Background

In this Section we discuss the relevant literature to the MCMC scheme with IS step that we propose in Section 4.3. First, we detail the MCMC scheme developed in Lunagómez et al. [2021] for the SNF model in Section 4.2.1. Second, in Section 4.2.2, we provide a brief review of the alternative approximation methods for distributions involving intractable normalising constants. Lastly, in Section 4.2.3, we review the study by Vitelli et al. [2017] on the Mallow's model for rank data, that inspired our proposed approach presented in Section 4.3.

4.2.1 Bayesian inference for the SNF model

In this Section, we provide details about the inferential scheme proposed by Lunagómez et al. [2021] for the intractable SNF model. Notably, Lunagómez et al. [2021] formulate a Bayesian inference scheme to estimate the network representative $\mathcal{G}^m$ and the dispersion parameter γ of the SNF model. In this regard, they specify a prior distribution $P(\gamma \mid \alpha_0)$ for the dispersion parameter which has support on $\mathbb{R}^+$. The choice of the prior distribution and its associated support is strongly related to the distance metric specified in the SNF model. In addition, a prior distribution for the network representative $P(A_{\mathcal{G}^m}|A_{\mathcal{G}_0}, \gamma_0)$ is specified, that has the same functional form as that of the SNF model. Thence, the posterior distribution of the model parameters given the network data, takes the form

$$P(A_{\mathcal{G}^m}, \gamma \mid A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}) \propto \frac{1}{Z(A_{\mathcal{G}_0}, \gamma_0)} \exp\left\{-\gamma_0 \phi(d_{\mathcal{G}}(A_{\mathcal{G}^m}, A_{\mathcal{G}_0}))\right\} P(\gamma \mid \alpha_0) \cdot \\ \frac{1}{Z(A_{\mathcal{G}^m}, \gamma)^N} \exp\{-\gamma \sum_{i=1}^{N} \phi(d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}))\}.$$

Under this hierarchical model, Lunagómez et al. [2021] formulate a Metropolis-Hastings algorithm to sample from the posterior of the network representative and the dispersion parameter. The normalising constant of the prior for the centroid cancels in the MH ratio, as it does not depend on the model parameters. However, the normalising constant of the likelihood depends on both the centroid and the dispersion parameter updated in the MCMC algorithm, leading to the problem of double-intractability. The authors overcome that issue using the Auxiliary Variable Method presented by Møller et al. [2006], that introduces an auxiliary variable into the computations for making posterior inferences that allow the normalising constants to cancel in the MH ratio.

Notably, the formulation of the Auxiliary Variable Method in the case of the SNF model involves the simulation of a set of auxiliary variables $\mathcal{G}^*$, defined in the same state space as the network data $\{\mathcal{G}_i\}_{i=1}^N$. Lunagómez et al. [2021] exploit the probabilistic mechanism of the CER model to specify the conditional density of the auxiliary network variables $\mathcal{G}_1^*, \ldots, \mathcal{G}_N^*$, which is a special case of the SNF model under the specification of the Hamming distance metric. Thus, the functional form of the conditional density of the auxiliary network variables is

$$f(A_{\mathcal{G}_1^*}, \ldots, A_{\mathcal{G}_N^*} \mid A_{\mathcal{G}^m}, \tilde{\alpha}) = \tilde{\alpha}^{\sum_{i=1}^N d_H(A_{\mathcal{G}_i^*}, A_{\mathcal{G}^m})} (1-\tilde{\alpha})^{N \cdot \frac{n(n-1)}{2} - \sum_{i=1}^N d_H(A_{\mathcal{G}_i^*}, A_{\mathcal{G}^m})}, \quad (4.1)$$

where $\tilde{\alpha}$ denotes the posterior mean of α, obtained after fitting the data $\{\mathcal{G}_i\}_{i=1}^N$ to the CER model.

Following the technique proposed by Møller et al. [2006], in each iteration of the MH algorithm a new state of both the model parameters and the auxiliary network variables will be proposed, with the latter sampled from a proposal distribution that has the same functional form as the likelihood. Under this formulation, the normalising constants of the likelihood and the proposal distribution of the auxiliary variables cancel out in the MH ratio, which takes the following form,

$$MH_{(A_{\mathcal{G}^m}^{(prop)}, \gamma^{(prop)} | A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)})} = \frac{f(A_{\vec{\mathcal{G}}^*}^{(prop)} | A_{\mathcal{G}^m}^{(prop)}, \tilde{\alpha})}{f(A_{\vec{\mathcal{G}}^*}^{(curr)} | A_{\mathcal{G}^m}^{(curr)}, \tilde{\alpha})} \times \frac{P(A_{\mathcal{G}^m}^{(prop)}, \gamma^{(prop)} | A_{\mathcal{G}_0}, \gamma_0, \alpha_0)}{P(A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)} | A_{\mathcal{G}_0}, \gamma_0, \alpha_0)} \times$$

$$\frac{P(A_{\vec{\mathcal{G}}} | A_{\mathcal{G}^m}^{(prop)}, \gamma^{(prop)})}{P(A_{\vec{\mathcal{G}}} | A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)})} \times \frac{P(A_{\vec{\mathcal{G}}^*}^{(curr)} | A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)})}{P(A_{\vec{\mathcal{G}}^*}^{(prop)} | A_{\mathcal{G}^m}^{(prop)}, \gamma^{(prop)})} \times \frac{Q(A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)} | A_{\mathcal{G}^m}^{(prop)}, \gamma^{(prop)})}{Q(A_{\mathcal{G}^m}^{(prop)}, \gamma^{(prop)} | A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)})}$$

where $\vec{\mathcal{G}} = \{\mathcal{G}_1, \ldots, \mathcal{G}_N\}$ are the data, $\vec{\mathcal{G}}^* = \{\mathcal{G}_1^*, \ldots, \mathcal{G}_N^*\}$ are the auxiliary variables and:

- $f(A^{(\cdot)}_{\vec{\mathcal{G}}^*}|A^{(\cdot)}_{\mathcal{G}^m}, \tilde{\alpha})$ is the conditional density for the auxiliary variables, as per equation (4.1).

- $P(A^{(\cdot)}_{\mathcal{G}^m}, \gamma^{(\cdot)}|A_{\mathcal{G}_0}, \gamma_0, \alpha_0)$ is the prior for the parameters $(A_{\mathcal{G}^m}, \gamma)$, with $A_{\mathcal{G}_0}, \gamma_0, \alpha_0$ denoting the hyperparameters of the prior.

- $P(A_{\vec{\mathcal{G}}}|A^{(\cdot)}_{\mathcal{G}^m}, \gamma^{(\cdot)})$ is the probability mass function of the SNF model, evaluated at the data $\vec{\mathcal{G}}$.

- $P(A^{(\cdot)}_{\vec{\mathcal{G}}^*}|A^{(\cdot)}_{\mathcal{G}^m}, \gamma^{(\cdot)})$ is the proposal distribution of the auxiliary variables $\vec{\mathcal{G}}^*$, that has the functional form of the SNF model.

- $Q(A^{(\cdot)}_{\mathcal{G}^m}, \gamma^{(\cdot)}|A^{(\cdot)}_{\mathcal{G}^m}, \gamma^{(\cdot)})$ is the proposal distribution for the parameters $(\mathcal{G}^m, \gamma)$.

A main challenge arising with the implementation of the Auxiliary Variable Method for the SNF model is the slow mixing of the chain for the γ parameter. Notably, there are batches of iterations with no proposals being accepted, resulting in the chain sticking on a single state for a series of iterations. Depending on the distance metric choice, this issue can become more apparent even for small network sizes. The occurrence of this phenomenon can be attributed to the discrepancy between the likelihood of the data, being the SNF model, and the choice of the auxiliary density, being the CER model. Depending on the choice of the distance metric for the SNF model, this discrepancy may become more drastic leading to a bad mixing of the chain or, in some cases, to the chain not exploring the state space at all. In Appendix B.1, we illustrate the mixing issue arising with the implementation of the Auxiliary Variable method on the SNF model, and compare it to the mixing of the MCMC chains under our proposed MCMC scheme with IS step.

4.2.2 Approximation Methods for Distributions with Intractable Normalising Constants

Statistical models of the form

$$P(\omega) = q(\omega)/\boldsymbol{Z}, \tag{4.2}$$

with $\boldsymbol{Z} = \int q(\omega)d\omega$ denoting the normalising constant and ω denoting a high-dimensional random variable, arise in a range of statistical problems (Ising [1925], Goodreau et al.

[2009], Rue and Held [2005]). Statistical inference on such models can become a challenging task, as typically the calculation of the normalising constant becomes computationally infeasible. The problem of approximating intractable normalising constants has been widely studied by researchers. Notably, there are two classes of approximation techniques developed in the literature, (i) the inexact-approximate techniques and (ii) the exact-approximate techniques, also called pseudo-marginal approaches. In this section we discuss the alternative methods developed under each class of approaches (i) and (ii).

Under the inexact-approximation techniques, the methods developed focus on obtaining a consistent estimator of the ratio of normalising constants,

$$r = \frac{\boldsymbol{Z_1}}{\boldsymbol{Z_2}}, \tag{4.3}$$

resulting from two intractable distributions of the form $P_i(\omega) = q_i(\omega)/\boldsymbol{Z_i}$ for $i = 1, 2$. A widely used approach for achieving this goal is through Importance Sampling.

As seen in the review presented by Chen and Shao [1997], there are two versions for approximating ratios of the form (4.3) using Importance Sampling. For the first version, a consistent estimator of (4.3) is obtained after specifying two IS densities $P_i^{IS}(\omega)$ for $i = 1, 2$, and drawing two independent samples $\omega_{i1}, \ldots, \omega_{in_i}$ from $P_i^{IS}(\omega)$ to yield the following estimator of r,

$$\hat{r} = \frac{(1/n_1)\sum_{j=1}^{n_1} q_1(\omega_{1j})/P_1^{IS}(\omega_{1j})}{(1/n_2)\sum_{j=1}^{n_2} q_2(\omega_{2j})/P_2^{IS}(\omega_{2j})}. \tag{4.4}$$

A second method for approximating r using Importance Sampling can be implemented when $\mathit{\Omega_1} \subset \mathit{\Omega_2}$ where $\mathit{\Omega_i}$ is the support of P_i. Under this condition, the ratio in (4.3) can be re-written as,

$$r = \frac{\boldsymbol{Z_1}}{\boldsymbol{Z_2}} = \mathbb{E}_2\left[\frac{q_1(\omega)}{q_2(\omega)}\right], \tag{4.5}$$

and can be approximated by drawing a sample $\omega_{21}, \ldots, \omega_{2n}$ from P_2 and obtaining,

$$\hat{r} = \frac{1}{n}\sum_{i=1}^{n} \frac{q_1(\omega_{2i})}{q_2(\omega_{2i})}. \tag{4.6}$$

A similar approach to the standard IS methods discussed above is the Bridge Sampling technique presented by Meng and Wong [1996]. Under the Bridge Sampling technique, the ratio r seen in (4.3) can be written as,

$$r = \frac{\boldsymbol{Z_1}}{\boldsymbol{Z_2}} = \frac{\mathbb{E}_2[q_1(\omega)\alpha(\omega)]}{\mathbb{E}_1[q_2(\omega)\alpha(\omega)]}, \tag{4.7}$$

where $\alpha(\omega)$ is an arbitrary function defined on $\mathit{\Omega_1} \cap \mathit{\Omega_2}$. Thence, r can be approximated by,

$$\hat{r} = \frac{(1/n_2)\sum_{j=1}^{n_2} q_1(\omega_{2j})\alpha(\omega_{2j})}{(1/n_1)\sum_{j=1}^{n_1} q_2(\omega_{1j})\alpha(\omega_{1j})}, \tag{4.8}$$

where $\omega_{i1}, \ldots, \omega_{in_i} \sim P_i$. A limitation of the latter two approximations is their poor performance when there is little overlap between P_i. Gelman and Meng [1994] address this restriction by implementing the Path Sampling technique. Under this framework, a scalar quantity $\lambda \in [0, 1]$ is introduced and the logarithm $\xi = -\log(r)$ is approximated instead of r, where ξ can be written as,

$$\xi = -\log(r) = -\log\left(\frac{\boldsymbol{Z}(\lambda_1)}{\boldsymbol{Z}(\lambda_2)}\right) = \mathbb{E}\left[\frac{U(\Omega, \Lambda)}{\pi_\lambda(\Lambda)}\right], \tag{4.9}$$

with $\lambda_1 = 0$ and $\lambda_2 = 1$, $U(\omega, \lambda) = (d/d\lambda)\log(q(\omega|\lambda))$, $\pi_\lambda(\lambda)$ is a prior density for λ and the expectation is taken with respect to the joint density $q(\omega, \lambda) = q(\omega|\lambda)\pi_\lambda(\lambda)$. For (ω_i, Λ_i), $i = 1, \ldots, n$ drawn from $q(\omega, \lambda)$, a consistent estimator of ξ is,

$$\hat{\xi} = \frac{1}{n}\sum_{i=1}^{n} \frac{U(\omega_i, \Lambda_i)}{\pi_\lambda(\Lambda_i)}. \tag{4.10}$$

The last inexact-approximation method that we review in this section is Ratio Importance Sampling introduced by Chen and Shao [1997]. Under this method, Chen and Shao [1997] introduce an arbitrary density $g(\omega)$, namely the Ratio IS density and re-write (4.3) as,

$$r = \frac{\boldsymbol{Z_1}}{\boldsymbol{Z_2}} = \frac{\mathbb{E}_g[q_1(\omega)/g(\omega)]}{\mathbb{E}_g[q_2(\omega)/g(\omega)]}, \tag{4.11}$$

which is a generalisation of the ratio obtained in (4.4). Hence, by drawing a sample $\omega_1, \ldots, \omega_n \sim g$, they obtain an estimate of r,

$$\hat{r} = \frac{\sum_{i=1}^{n} q_1(\omega_i)/g(\omega_i)}{\sum_{i=1}^{n} q_2(\omega_i)/g(\omega_i)}, \tag{4.12}$$

which is consistent even for a dependent sample $\omega_1, \ldots, \omega_n$ drawn.

We now introduce the exact-approximate or pseudo-marginal approaches, that focus on obtaining an unbiased estimator of the normalising constant or the ratio of normalising constants. A key feature of these methods is the utilisation of an auxiliary variable for the approximation. We re-write (4.2) by introducing θ to denote the model parameters as follows,

$$P(\omega|\theta) = q(\omega|\theta)/\boldsymbol{Z}(\theta). \tag{4.13}$$

Møller et al. [2006] propose the use of an auxiliary variable x that has the same support as that of ω, with density $f(x|\theta, \omega)$ to obtain an unbiased estimator of $\boldsymbol{Z}(\theta)$. In light of Importance Sampling, under Møller et al. [2006] $\boldsymbol{Z}(\theta)$ can be written as,

$$\boldsymbol{Z}(\theta) = \mathbb{E}\left[\frac{q(x|\theta)}{f(x|\theta, \omega)}\right], \tag{4.14}$$

where the expectation is taken with respect to the density of the auxiliary variable x, $f(x|\theta, \omega)$. In this regard, they propose sampling x from the distribution $P(\cdot|\theta)$ and use the approximation,

$$\boldsymbol{Z}(\theta) \approx \frac{q(x|\theta)}{f(x|\theta, \omega)}. \tag{4.15}$$

Thus, they can substitute the normalising constant $\boldsymbol{Z}(\cdot)$ by its unbiased estimator $q(x|\cdot)/f(x|\cdot, \omega)$ in the MH acceptance ratio.

Analogously to Møller et al. [2006], Murray et al. [2012] also use an exact-approximate method to obtain an unbiased estimator for the ratio of normalising constants found in the acceptance ratio of a Metropolis-Hastings algorithm. Specifically, they also utilise an auxiliary variable x drawn from distribution $P(\cdot|\theta^{(prop)})$, and obtain the unbiased estimator,

$$\frac{\boldsymbol{Z}(\theta)}{\boldsymbol{Z}(\theta^{(prop)})} \approx \frac{q(x|\theta)}{q(x|\theta^{(prop)})}. \tag{4.16}$$

A key weakness of the exact-approximate methods is the incidence of non-acceptance of newly proposed states for (θ, x), for a large number of iterations. Notably, Møller et al. [2006] explain that this behaviour of the algorithm can be attributed to the choice of the auxiliary density $f(x|\theta, \omega)$, with regard to how good it approximates the likelihood $q(\cdot)/\boldsymbol{Z}(\theta)$. The problem of non-acceptance for long runs, also increases the autocorrelation between posterior draws, implying that more iterations are required for the algorithm to converge and explore the posterior distribution.

Finally, a more recent class of algorithms related to the inexact-approximate methods discussed, is the Noisy MCMC algorithms. Notably, the Noisy Metropolis-Hastings algorithm as introduced in Alquier et al. [2016] focuses on replacing the intractable acceptance ratio in the Metropolis-Hastings algorithm with an approximation, by drawing values from a suitable probability distribution. They further consider an extension of the Exchange algorithm (Murray et al. [2012]), namely the Noisy Exchange algorithm, that replaces the acceptance ratio with its estimator, by substituting the ratio of normalising constants with

$$\frac{\boldsymbol{Z}(\theta)}{\boldsymbol{Z}(\theta^{(prop)})} \approx \frac{1}{n}\sum_{i=1}^{n}\frac{q(x_i|\theta)}{q(x_i|\theta^{(prop)})}, \tag{4.17}$$

where $x_1, \ldots, x_n$ sampled from the distribution $P(\cdot|\theta)$. For $n = 1$ the Noisy Exchange algorithm is the Exchange algorithm as presented in Murray et al. [2012]. An investigation on the stability of Noisy Metropolis-Hastings algorithms is provided in Medina-Aguayo et al. [2016].

This review should not be considered as an extensive review on approximating intractable distributions. We note that there is a vast literature on this topic, while in this section we outlined the methods that are most relevant to our work. In view of this, there is more than one way to deal with the intractability of the SNF model. As discussed in Section 4.2.1, the Exact-Approximate methods and specifically the Auxiliary Variable technique introduced by Møller et al. [2006] provide poor results with respect to the MCMC convergence for some distance metric specifications for the SNF model. In this respect, we formulated an Importance Sampling step within our MCMC as that proposed by Chen and Shao [1997], namely the Ratio Importance Sampling presented in this section. Under this approach, we are able to confront the mixing issues arising under the Auxiliary Variable technique.

4.2.3 Importance Sampling for the Mallows Model

We now review the Importance Sampling technique used in Mallow's model for multiple rank data, which inspired the development of our MCMC scheme described in Section 4.3. Rank data is a data type used to represent individual preferences over a set of n items, labelled $\mathcal{A} = \{A_1, \ldots, A_n\}$. There are various real-life instances where rank data emerge, from sports, where teams are ranked according to their performance, to marketing, where different products or services are ranked according to consumer preferences.

Let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_N$ denote N permutations of the set of labels $\mathcal{A}$, for N assessors. Let also $\mathcal{P}_n$ denote the $n!$-sized space of permutations of n items. Thence, the ranking of the j^{th} assessor can be represented by $\boldsymbol{R}_j = (R_{1j}, \dots, R_{nj})$, with $R_{ij} = \boldsymbol{X}_j^{-1}(A_i)$ denoting the rank given to the i^{th} item, from the j^{th} assessor.

Mallows model (Mallows [1957]) has been broadly used for statistical inference on multiple rank data, as it allows to model a set of N rankings $\boldsymbol{R}_1, \dots, \boldsymbol{R}_N$ with respect to a consensus ranking $\boldsymbol{\rho}$ and a scale parameter a, under a specified distance function $d(\cdot, \cdot) : \mathcal{P}_n \times \mathcal{P}_n \to [0, \infty)$. Notably, Mallows model has the same functional form as the SNF model, thus, the likelihood of observing a set of N rankings $\boldsymbol{R}_1, \dots, \boldsymbol{R}_N$ has the form,

$$P(\boldsymbol{R}_1, \dots, \boldsymbol{R}_N \mid \alpha, \boldsymbol{\rho}) = \frac{1}{Z(\alpha, \boldsymbol{\rho})^N} \exp\left\{ -\frac{\alpha}{n} \sum_{j=1}^{N} d(\boldsymbol{R}_j, \boldsymbol{\rho}) \right\} \prod_{j=1}^{N} \{1_{\mathcal{P}_n}(\boldsymbol{R}_j)\},$$

with $Z(\alpha, \boldsymbol{\rho}) = \sum_{\boldsymbol{r} \in \mathcal{P}_n} \exp\{-\frac{\alpha}{n} d(\boldsymbol{r}, \boldsymbol{\rho})\}$ representing the normalising constant depending on both the latent consensus ranking $\boldsymbol{\rho}$ and scale parameter α. Drawing inference for this model can be a challenging task for specific distance functions and large number of items n, due to the presence of the normalising constant.

In the case of rank data, there are distance functions which remain unchanged after a relabelling of the items, called right-invariant distance functions. For a right-invariant distance function, we have that $d(\boldsymbol{r}_1, \boldsymbol{r}_2) = d(\boldsymbol{r}_1 \boldsymbol{r}_2^{-1}, \boldsymbol{1}_n)$, with $\boldsymbol{1}_n = \{1, 2, \dots, n\}$, which leads to the following simplified form of the normalising constant

$$Z(\alpha) = \sum_{\boldsymbol{r} \in \mathcal{P}_n} \exp\{-\frac{\alpha}{n} d(\boldsymbol{r}, \boldsymbol{1}_n)\},$$

that depends only on the scale parameter α. The exclusion of the consensus ranking $\boldsymbol{\rho}$ from the normalising constant is advantageous for the calculation of the latter in the Mallows model.

Despite this simplification of the form of the normalising constant, there are distances in the class of right-invariant distances, for which $Z(\alpha)$ still can not be calculated analytically. In the work by Vitelli et al. [2017] only right-invariant distances are considered, and the authors propose an Off-line Importance Sampling scheme to approximate the normalising constant $Z(\alpha)$ for the right-invariant distances, that render the exact calculation of the normalising constant infeasible.

Notably they consider drawing a sample of K rankings, $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_K$, from an IS density $q(\boldsymbol{R})$, to obtain an unbiased estimate of the normalising constant as follows,

$$\hat{Z}(\alpha) = \frac{1}{K} \sum_{k=1}^{K} \frac{\exp\{-\frac{\alpha}{n} d(\boldsymbol{R}_k, \mathbf{1}_n)\}}{q(\boldsymbol{R}_k)}.$$

The choice of the IS density $q(\boldsymbol{R})$ plays a key role to obtain an unbiased estimate of the normalising constant. Ideally, one would need to simulate an Importance sample directly from the likelihood, however, this is infeasible in the case of the Mallows model. To obtain an IS density that closely approximates the likelihood, Vitelli et al. [2017] use a pseudo-likelihood approximation of the Mallows model in the following way:

- Draw a uniform sample $\{i_1, \ldots, i_n\}$ from the set of permutations of n items, $\mathcal{P}_n$.
- Obtain the pseudo-likelihood factorization:

$$\begin{aligned} P(\boldsymbol{R} \mid \mathbf{1}_n) = P(R_{i_1} \mid R_{i_2}, \ldots, R_{i_n}, \mathbf{1}_n) P(R_{i_2} \mid R_{i_3}, \ldots, R_{i_n}, \mathbf{1}_n) \cdots \\ P(R_{i_{n-1}} \mid R_{i_n}, \mathbf{1}_n) P(R_{i_n} \mid \mathbf{1}_n), \end{aligned}$$

  with $P(\cdot \mid \cdot, \mathbf{1}_n)$ having the form of the Mallows model.

- Sample first R_{i_n} from its conditional distribution $P(R_{i_n} \mid \mathbf{1}_n)$, and then conditionally on R_{i_n} sample $R_{i_{n-1}}$ from its conditional distribution $P(R_{i_{n-1}} \mid R_{i_n}, \mathbf{1}_n)$, and so forth.
- Thence, the k^{th} sample $\boldsymbol{R}^k$ has density $q(\boldsymbol{R}^k) = P(\boldsymbol{R}^k \mid \mathbf{1}_n)$.

In our work, we formulate an Importance Sampler using a different set up for the IS density to that presented in Vitelli et al. [2017]. The right-invariance property does not hold for the majority of distance functions measuring dissimilarities among networks, as there are different properties governing rank data compared to network data. Thus, it is not straightforward to obtain a simplification of the normalising constant for the SNF model. Alternatively, we formulate an approximation of the normalising constant which depends both on the representative network and the dispersion, using the CER model presented in Lunagómez et al. [2021] as the IS density. Our approach allows us to infer the parameters of the SNF model for various distance metric specifications.

4.3 MCMC scheme with IS step for the SNF model

In this section we introduce a new network distance metric, namely the HS distance, and describe our proposed MCMC scheme with IS step for the SNF model. Notably, in Section 4.3.1, we introduce the HS distance metric and the motivation for formulating it. In Section 4.3.2, we present the IS step for approximating the normalising constant in the SNF model. In Section 4.3.3, we discuss alternative specifications of the IS density. In Section 4.3.4, our proposed MCMC scheme with IS step is introduced, while the tuning of the IS density is discussed in Section 4.3.5. In Section 4.3.6, we describe our investigation on the bias of the estimate of the normalising constant with respect to increasing sample size N, and lastly in Section 4.3.7, we discuss our approach on dealing with the computationally intensive task of detecting cycles in a network, associated with the HS distance metric.

4.3.1 A Motivating Example

In this section we present a multiple network data example that motivated the work presented in this Chapter. Data have been collected on the aggressive interactions among species of fish in different coral reefs, at multiple regions in the Indo-Pacific ocean (Keith et al. [2018]). We use a network representation for the data collected, where nodes represent fish species and edges represent aggressive interactions between them.

From an ecological perspective, directed cycles indicate a form of competition among fish, called intransitive competition. Intransitive competition is a type of competition in which no species become dominant, and it is important because it stabilises species coexistence in the system (Muyinda et al. [2020a], Muyinda et al. [2020b], Mohd [2019]). However, this type of competition can be liable to disruption due to environmental disturbance. Hence, a research question arising for the network data collected is the following: To what extent are cycles formed from the aggressive interactions between fish in the network data?

One way to answer this research question is to consider the SNF model that allows inferring a representative network for a network population, determined through a user-specified measure of dissimilarity. In addition, the SNF model involves a dispersion parameter that quantifies the level of dissimilarity between the network data and the network representative, with respect to the specified distance metric. The flexibility in the choice of the distance metric was a key factor for implementing the SNF model for the multiple network data example presented in this section. As our interest lies in the cycles formed in the network data, we construct a dissimilarity measure that captures information about dissimilarities between the cycles of two networks. Specifically, we obtain the Hamming-Symmetric difference (HS) distance metric consisting of two parts:

1. The Hamming distance, that counts the not in common edges and non-edges between two graphs.

2. The symmetric difference between the cycles formed in two graphs, that counts the number of not in common cycles in two graphs.

Hence, a mathematical representation of the constructed distance metric is,

$$d_{\mathrm{HS}}(\mathcal{G}_1, \mathcal{G}_2) = d_{\mathrm{H}}(\mathcal{G}_1, \mathcal{G}_2) + \lambda \cdot \mid C_{\mathcal{G}_1} \Delta C_{\mathcal{G}_2} \mid,$$

where $d_{\mathrm{H}}(\cdot, \cdot)$ denotes the Hamming distance, $C_{\mathcal{G}_i}$ denotes the cycles in graph i, Δ indicates the symmetric difference and $\lambda \in \mathbb{R}$ is a weighting factor. In Appendix B.2, we show that HS is a distance metric. Under this construction, we encode information about dissimilarities in the structure of the networks, with respect to both their edges and cycles. The tuning of the λ parameter corresponds to how much influence we allow the symmetric difference to have on the total distance. In Section 4.4.1, we explore the behaviour of the HS distance for various sizes of λ through synthetic data experiments. In the simulations studies and real data examples presented in this Chapter, we assume λ to be equal to 1 suggesting equal importance between the Hamming and the Symmetric difference distance.

The specification of the HS distance metric for the SNF model induces challenges with respect to the Bayesian inference scheme proposed by Lunagómez et al. [2021]. Notably, the problem of non-acceptance of new states for a great number of iterations becomes more evident with the use of the HS distance metric. As a result, the inference drawn for the dispersion parameter γ resembles to obtaining a single estimate for it, rather than draws from its posterior distribution, thus no uncertainty is encoded about γ. This finding led us to consider an alternative method to infer the SNF model parameters, and specifically to deal with the problem of the intractable normalising constant in the likelihood.

Inspired by Vitelli et al. [2017], we formulate an Importance Sampling scheme in our MCMC algorithm, with research objective the development of a Bayesian inference scheme for the SNF model that is applicable for a range of different distance metrics. This includes the newly proposed HS distance, as well as considering the Jaccard distance which is a well-known distance metric capturing information about the edge structure of the networks (Levandowsky and Winter [1971], Donnat and Holmes [2018]). We note here that for both the Jaccard and the HS distance, the Auxiliary Variable technique (Møller et al. [2006]) adapted in Lunagómez et al. [2021] provided poor chain mixing results.

4.3.2 Formulation of IS step for the SNF model

Importance Sampling is a method originating from the problem of evaluating the integral,

$$\mathbb{E}_f[h(X)] = \int_{S_X} h(x)f(x)dx, \tag{4.18}$$

with S_X denoting the support of the random variable X (Robert and Casella [2013]). Notably, (4.18) can be rewritten as,

$$\mathbb{E}_f[h(X)] = \int_{S_X} h(x)\frac{f(x)}{g(x)}g(x)dx, \tag{4.19}$$

and can be approximated by sampling $X_1, \ldots, X_K$ from distribution g and calculating the empirical mean

$$\mathbb{E}_f[h(X)] \approx \frac{1}{K}\sum_{j=1}^{K}\frac{f(X_j)}{g(X_j)}h(X_j), \tag{4.20}$$

when sampling directly from the target distribution f is not feasible.

The problem of approximating normalising constants is associated with this class of problems, as it involves the calculation of an intractable sum or integral as that of equation (4.18). Hence, Importance Sampling and variations of it, have been used extensively for statistical models that involve an intractable normalising constant. In our case, the normalising constant of the SNF model has the following form,

$$Z(A_{\mathcal{G}^m}, \gamma) = \sum_{A_{\mathcal{G}} \in \{\mathcal{G}_{|n|}\}} \exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}}, A_{\mathcal{G}^m}))\}, \tag{4.21}$$

involving a sum over the space of n-node graphs, $\{\mathcal{G}_{|n|}\}$. Considering the formulation seen in (4.19), we can rewrite the sum as

$$\sum_{A_{\mathcal{G}} \in \{\mathcal{G}_{|n|}\}} \exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}}, A_{\mathcal{G}^m}))\} = \sum_{A_{\mathcal{G}} \in \{\mathcal{G}_{|n|}\}} \frac{\exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}}, A_{\mathcal{G}^m}))\}}{g(A_{\mathcal{G}})} g(A_{\mathcal{G}}) = \mathbb{E}_g\left[\frac{\exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}}, A_{\mathcal{G}^m}))\}}{g(A_{\mathcal{G}})}\right], \tag{4.22}$$

which can then be approximated by drawing a sample of networks $\mathcal{G}_1, \dots, \mathcal{G}_K$ from the IS density g and calculating,

$$\hat{Z}(A_{\mathcal{G}^m}, \gamma) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{\exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}_k}, A_{\mathcal{G}^m}))\}}{g(A_{\mathcal{G}_k})}. \tag{4.23}$$

The choice of the IS density plays a key role to obtain a good approximation of the normalising constant. In the next section, we discuss the specification of the IS density for the SNF model and the possible sampling schemes.

4.3.3 IS density specification for multiple network data

One advantage of the IS method is the flexibility that allows with respect to the specification of the IS density. In this regard, choices of distributions that are easy to sample from are preferred (Robert and Casella [2013]). However, different IS densities can have different performances in estimating the normalising constant in equation (4.21).

In the network literature, there are several frameworks under which we can simulate a population of networks, as reviewed in Chapter 2. In our problem, a natural choice of the IS density is the distance-based CER model for two main reasons, (i) the CER model is a member of the Spherical Network Family (SNF) of models (Lunagómez et al. [2021]), and (ii) sampling network data from the CER model is quick, thus will result in a less computationally intensive MCMC algorithm. In this section we consider both a single CER model as well as a mixture of CER models as the IS density, with the latter allowing a heavy tailed proposal density to be implemented (Robert and Casella [2013]).

IS1 Single CER model

We now briefly review the CER model introduced by Lunagómez et al. [2021]. For a more detailed description of the model, refer to Section 2.2. Under the CER model, Lunagómez et al. [2021] assume that a population of networks is generated by perturbing the edges of a centroid network $A_{\mathcal{G}^m}$, using a sequence of Bernoulli random variables with probability α as follows:

$$A_{\mathcal{G}}(i,j) \mid (A_{\mathcal{G}^m}(i,j), \alpha) = |A_{\mathcal{G}^m}(i,j) - Z(i,j)|. \tag{4.24}$$

where $Z(i,j)$'s are *iid* Ber(α), with $0 < \alpha < 0.5$.

Thus the CER model is a network distribution which depends on the Hamming distance metric, and the probability of observing a graph $A_{\mathcal{G}}$ under this model has the form:

$$p(A_{\mathcal{G}}|A_{\mathcal{G}^m}, \alpha) = \alpha^{d_H(A_{\mathcal{G}}, A_{\mathcal{G}^m})} \cdot (1-\alpha)^{\frac{n(n-1)}{2} - d_H(A_{\mathcal{G}}, A_{\mathcal{G}^m})} \tag{4.25}$$

where $d_H(\cdot,\cdot)$ denotes the Hamming distance metric and n is the number of the networks' nodes.

Sampling from the CER model can be performed in two different ways:

1. Using an MCMC scheme with target distribution the density of the CER model (4.25) with fixed parameters $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$, as seen in Lunagómez et al. [2021].
2. Using a Monte Carlo scheme to sample independent network data from the CER model with parameters $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$.

Under sampling scheme (1), an MH algorithm is formulated to sample network data $A_{\mathcal{G}}$ from the CER model. This sampling scheme was proposed in Lunagómez et al. [2021] to sample network data from the SNF model. As CER is a member of the SNF, we adapt a similar sampling scheme for the case of the CER model. Notably, we propose a new network with adjacency $A_{\mathcal{G}}^{(prop)}$ by perturbing the edges of the current network with adjacency $A_{\mathcal{G}}^{(curr)}$ in the following way:

$$A_{\mathcal{G}}^{(prop)}(i,j) = \begin{cases} 1 - A_{\mathcal{G}}^{(curr)}(i,j), \text{ with probability } \omega \\ A_{\mathcal{G}}^{(curr)}(i,j), \text{ with probability } 1-\omega \end{cases}, \tag{4.26}$$

and accept it with probability,

$$\min\left\{1, \frac{\tilde{\alpha}^{d_{\mathrm{H}}(A_{\mathcal{G}}^{(prop)}, \tilde{A}_{\mathcal{G}^m})}(1-\tilde{\alpha})^{\frac{n(n-1)}{2} - d_{\mathrm{H}}(A_{\mathcal{G}}^{(prop)}, \tilde{A}_{\mathcal{G}^m})}}{\tilde{\alpha}^{d_{\mathrm{H}}(A_{\mathcal{G}}^{(curr)}, \tilde{A}_{\mathcal{G}^m})}(1-\tilde{\alpha})^{\frac{n(n-1)}{2} - d_{\mathrm{H}}(A_{\mathcal{G}}^{(curr)}, \tilde{A}_{\mathcal{G}^m})}}\right\}. \tag{4.27}$$

We note here that under this proposal scheme, the proposal distribution cancels from the MH ratio.

Under sampling scheme (2), we obtain a sample of independent network data from the CER model with parameters $\tilde{A}_{\mathcal{G}^m}$ and $\tilde{\alpha}$. Each network in the sample is drawn independently by perturbing the edges of the centroid $\tilde{A}_{\mathcal{G}^m}$ using a sequence of iid Bernoulli random variables with probability $\tilde{\alpha}$, as per equation (4.24).

In Figures 4.1 and 4.2, we illustrate the differences between the two sampling schemes for a specific example of 5-node networks. In this illustration, we consider 5-node networks as the space of 5-node graphs consists of 1024 networks, allowing us to identify the number of times that each network in the space of graphs is being sampled, under each of the two sampling schemes for the CER model with specified parameters $\tilde{A}_{\mathcal{G}^m}$ and $\tilde{\alpha}$.

A key difference between the two sampling procedures lies in the network samples obtained under each scheme. Notably, the MCMC sampling scheme that involves an accept/reject step is more rigid with the networks being sampled, so that they maintain a Hamming distance from the centroid according to the size of the dispersion $\tilde{\alpha}$. On the other hand, the Monte Carlo sampling scheme allows greater variability in the network data sampled that resembles sampling Uniformly from the space of graphs, even for small sizes of the dispersion parameter.

Even though the variability of the network data sampled under the MC scheme can be advantageous for the performance of the IS, a challenge arises with the HS distance metric specification. Notably, the HS distance involves the calculation of cycles formed within a network which is a computationally intensive task, even for moderate network sizes. The variability of the networks sampled under the MC

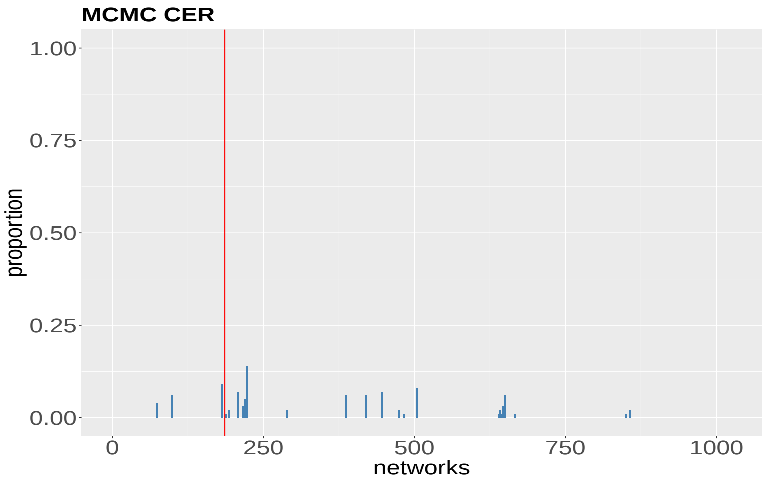

Figure 4.1: Proportion of times that each network in the space of 5-node graphs is being sampled under the MCMC sampling scheme. The red line indicates the centroid $\tilde{A}_{\mathcal{G}^m}$. The dispersion is $\tilde{\alpha} = 0.2$. Networks on the x axis are ordered according to their density, from sparser to denser.

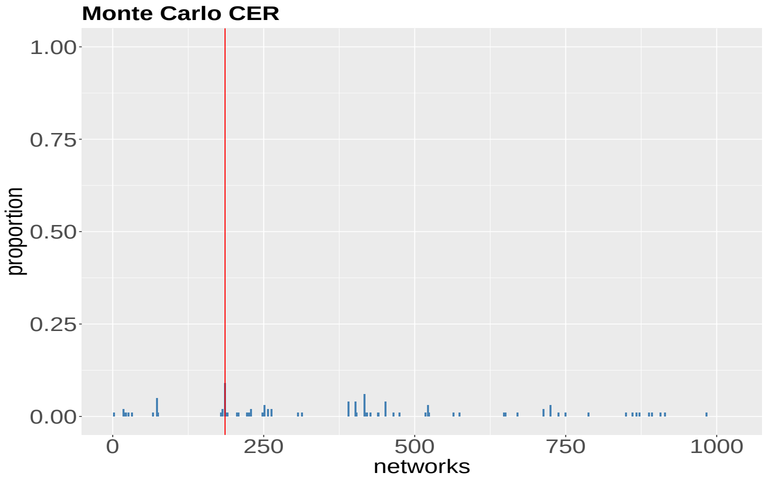

Figure 4.2: Proportion of times that each network in the space of 5-node graphs is being sampled under the MC sampling scheme. The red line indicates the centroid $\tilde{A}_{\mathcal{G}^m}$. The dispersion is $\tilde{\alpha} = 0.2$. Networks on the x axis are ordered according to their density, from sparser to denser.

scheme implies denser networks in the IS sample, rendering the cycle detection task computationally intensive, and in some cases infeasible. In view of that limitation, in the simulation studies performed in Section 4.4, we consider the MCMC sampling scheme to sample from the CER model as a more holistic approach. We further note here that dependent IS samples still allow for a consistent estimator of the ratio of normalising constants to be obtained, as presented in Chen and Shao [1997].

IS2 Mixture of CER models

To guarantee the derivation of a finite variance estimator for the normalising constant, a key attribute of the IS density is to be heavier tailed than the target distribution (Robert and Casella [2013]). In this regard, another choice of IS density considered herein is a mixture of CER models, that allows sampling of a wider range of network data, resulting in a heavier tailed IS distribution compared to considering a single CER model.

Notably, in our problem we consider a mixture of CER models with varying sizes of the dispersion parameter $\tilde{\alpha}$, while keeping fixed the centroid parameter $\tilde{A}_{\mathcal{G}^m}$. Under this set up, the sampling distribution of $A_{\mathcal{G}}$ has the form

$$p(A_{\mathcal{G}}|\tilde{A}_{\mathcal{G}^m}, \tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}) = \sum_{j=1}^{J} \beta_j \tilde{\alpha}_j^{d_H(A_{\mathcal{G}}, \tilde{A}_{\mathcal{G}^m})} \cdot (1 - \tilde{\alpha}_j)^{\frac{n(n-1)}{2} - d_H(A_{\mathcal{G}}, \tilde{A}_{\mathcal{G}^m})}, \tag{4.28}$$

where $j \in \{1, \ldots, J\}$ indicates the mixture component, and $\boldsymbol{\beta}$ is a vector of probabilities for the mixture components.

To sample from the mixture of CER models, we adapt the standard procedure for sampling from a mixture model, involving the following steps:

1. Specify the number of mixture components J and the vector of probabilities $\{\beta_j\}_{j=1}^J$ such that $\sum_{j=1}^J \beta_j = 1$.
2. Draw a categorical random variable with probability $\boldsymbol{\beta}$.
3. Depending on the category drawn, sample from the corresponding CER model with dispersion parameter $\tilde{\alpha}_j$ being component specific, and centroid $\tilde{A}_{\mathcal{G}^m}$ using the MC sampling scheme presented in this section.

We note here that similarly to the MC sampling scheme for the single CER model, sampling from the mixture of CER models results in greater variability of the networks drawn, imposing challenges when implemented for the HS distance specification. In Section 4.4.2, we explore the performance of the different specifications of the IS density and the different sampling schemes presented in this section.

In the next section, we introduce the MCMC scheme formulated to sample from the posterior distribution of the parameters of the SNF model, using an IS step for the approximation of the normalising constant.

4.3.4 MCMC scheme with IS step

In this section we present the algorithmic scheme formulated to obtain draws from the posterior distribution of the SNF model parameters. As seen in subsection 4.2.1, the joint posterior distribution of the centroid $A_{\mathcal{G}^m}$ and the dispersion parameter γ has the following hierarchical structure,

$$\begin{aligned} P(A_{\mathcal{G}^m}, \gamma \mid A_{\mathcal{G}_1}, \cdots, A_{\mathcal{G}_N}) \propto \frac{1}{Z(A_{\mathcal{G}_0}, \gamma_0)} \exp\left\{-\gamma_0 \phi(d_{\mathcal{G}}(A_{\mathcal{G}^m}, A_{\mathcal{G}_0}))\right\} P(\gamma \mid \alpha_0) \cdot \\ \frac{1}{Z(A_{\mathcal{G}^m}, \gamma)^N} \exp\{-\gamma \sum_{i=1}^{N} \phi(d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}))\}. \end{aligned} \tag{4.29}$$

Similarly to Lunagómez et al. [2021], we implement a Metropolis-Hastings algorithm to sample from the joint posterior of the parameters. However, to overcome the double-intractability problem, we implement an IS scheme as seen in section 4.3.2, conversely to the Auxiliary Variable Method adapted in Lunagómez et al. [2021]. Notably, we obtain posterior draws from the target distribution seen in equation (4.29), after substituting the normalising constant in the likelihood with its estimator, that takes the following form under the specification of a single CER model for the IS density:

$$\hat{Z}(A_{\mathcal{G}^m}, \gamma) \approx \frac{1}{K} \sum_{k=1}^{K} \frac{\exp\{-\gamma \cdot \phi(d_{\mathcal{G}}(A_{\mathcal{G}_k}, A_{\mathcal{G}^m}))\}}{\tilde{\alpha}^{d_{\mathrm{H}}(A_{\mathcal{G}_k}, \tilde{A}_{\mathcal{G}^m})} (1-\tilde{\alpha})^{\frac{n(n-1)}{2} - d_{\mathrm{H}}(A_{\mathcal{G}_k}, \tilde{A}_{\mathcal{G}^m})}}, \tag{4.30}$$

where $\{A_{\mathcal{G}_k}\}_{k=1}^{K}$ are network data sampled from the CER model with parameters $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$. To obtain posterior draws for the parameters $A_{\mathcal{G}^m}$ and γ, we use a mixture of kernels similarly to the scheme proposed in Lunagómez et al. [2021].

Notably, we update the adjacency matrix of the centroid $A_{\mathcal{G}^m}$ using either of the following two proposals,

(I) We perturb the edges of the current centroid $A_{\mathcal{G}^m}^{(curr)}$ as follows:

$$A_{\mathcal{G}^m}^{(prop)}(i,j) = \begin{cases} 1 - A_{\mathcal{G}^m}^{(curr)}(i,j), \text{ with probability } \omega \\ A_{\mathcal{G}^m}^{(curr)}(i,j), \text{ with probability } 1-\omega \end{cases}.$$

(II) We propose a new network representative $A_{\mathcal{G}^m}^{(prop)}$, with each edge of the proposed representative being drawn independently, from a Bernoulli distribution with parameter $\frac{1}{N}\sum_{l=1}^{N} A_{\mathcal{G}_l}(i,j)$.

Under case (I), we accept the proposed network representative $A_{\mathcal{G}^m}^{(prop)}$ with probability

$$\min\left\{1, \frac{\exp\left\{-\gamma_0 d_{\mathcal{G}}(A_{\mathcal{G}^m}^{(prop)}, A_{\mathcal{G}_0})\right\} \widehat{Z}(A_{\mathcal{G}^m}^{(prop)}, \gamma^{(curr)})^{-N} \exp\{-\gamma^{(curr)} \sum_{i=1}^{N} d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}^{(prop)})\}}{\exp\left\{-\gamma_0 d_{\mathcal{G}}(A_{\mathcal{G}^m}^{(curr)}, A_{\mathcal{G}_0})\right\} \widehat{Z}(A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)})^{-N} \exp\{-\gamma^{(curr)} \sum_{i=1}^{N} d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}^{(curr)})\}}\right\},$$

while under case (II), we accept the proposed network representative $A_{\mathcal{G}^m}^{(prop)}$ with probability

$$\min\left\{1, \frac{\exp\left\{-\gamma_0 d_{\mathcal{G}}(A_{\mathcal{G}^m}^{(prop)}, A_{\mathcal{G}_0})\right\} \frac{\exp\{-\gamma^{(curr)} \sum_{i=1}^{N} d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}^{(prop)})\}}{\widehat{Z}(A_{\mathcal{G}^m}^{(prop)}, \gamma^{(curr)})^{N}} Q(A_{\mathcal{G}^m}^{(curr)} | A_{\mathcal{G}^m}^{(prop)})}{\exp\left\{-\gamma_0 d_{\mathcal{G}}(A_{\mathcal{G}^m}^{(curr)}, A_{\mathcal{G}_0})\right\} \frac{\exp\{-\gamma^{(curr)} \sum_{i=1}^{N} d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}^{(curr)})\}}{\widehat{Z}(A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)})^{N}} Q(A_{\mathcal{G}^m}^{(prop)} | A_{\mathcal{G}^m}^{(curr)})}\right\},$$

We note here that the proposal distribution under case (I) is symmetric, thus it cancels out from the Metropolis ratio, while under case (II) the proposal distribution $Q(A_{\mathcal{G}^m}^{(\cdot)} | A_{\mathcal{G}^m}^{(\cdot)})$ does not cancel.

Accordingly, we use a mixture of K random walks to propose values for the dispersion parameter γ, as follows:

1. Draw a uniform random variable $u \sim$ Unif$(-v_k, v_k)$, with k indicating the k^{th} proposal.
2. Perturb the current state $\gamma^{(curr)}$ by the uniform random variable drawn, $y = \gamma^{(curr)} + u$.
3. The newly proposed value for γ is $\gamma^{(prop)} = \begin{cases} y, \text{if } y > 0 \\ -y, \text{if } y < 0 \end{cases}$,

which we accept with probability

$$\min\left\{1, \frac{\widehat{Z}(A_{\mathcal{G}^m}^{(curr)}, \gamma^{(prop)})^{-N} \exp\{-\gamma^{(prop)} \sum_{i=1}^{N} d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}^{(curr)})\} P(\gamma^{(prop)} \mid \alpha_0)}{\widehat{Z}(A_{\mathcal{G}^m}^{(curr)}, \gamma^{(curr)})^{-N} \exp\{-\gamma^{(curr)} \sum_{i=1}^{N} d_{\mathcal{G}}(A_{\mathcal{G}_i}, A_{\mathcal{G}^m}^{(curr)})\} P(\gamma^{(curr)} \mid \alpha_0)}\right\}.$$

Under this scheme, in each iteration of the MCMC algorithm, we draw a new sample from the IS density to calculate $\widehat{Z}$ in the nominator and denominator of the MH ratio. In the next section we discuss the choice of the parameters $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$ for the CER model.

4.3.5 Tuning of the IS density

In this section we discuss the tuning of the parameters of the IS densities considered in this study. Specifically, we discuss the specification of the centroid and the dispersion in the case of a single CER model, as well as in the case of a mixture of CER models.

For the case of a single CER model being the IS density, we consider two alternative ways to specify the parameters $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$. Under the first construction, we develop an MCMC scheme in which the parameters of the IS density remain fixed throughout the iterations. We call this scheme a non-adaptive MCMC scheme. Under the second construction, we allow the values of the parameters $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$ to vary throughout the iterations, depending on the current values of the SNF model parameters. We call this scheme an adaptive MCMC scheme. Below we discuss in more detail the two alternative schemes.

A1 **Non-adaptive MCMC scheme**

Under the non-adaptive MCMC scheme, we determine $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$ by fitting the data to the CER model to obtain the posterior mean of $\tilde{\alpha}$ and posterior mode of $\tilde{A}_{\mathcal{G}^m}$.

In this way, we encode information about the data at hand, that may allow a better approximation of the normalising constant. We investigate this potential in the simulation studies following in Section 4.4.2.

A2 **Adaptive MCMC scheme**

Under the adaptive MCMC scheme, we specify the parameters $\tilde{\alpha}$ and $\tilde{A}_{\mathcal{G}^m}$ depending on the current values of the parameters of the SNF model. To determine $\tilde{A}_{\mathcal{G}^m}$ from the current value of the centroid network of the CER model, is straightforward, as both represent centroid networks and no transformation is required.

However, for the specification of $\tilde{\alpha}$, we consider a transformation of γ under which the CER model is the SNF model with the Hamming distance metric specification, as seen in Lunagómez et al. [2021]. Under this case, the transformation of γ into α is the following:

$$\alpha = \frac{1}{1+e^{\gamma}}. \tag{4.31}$$

We note here that this transformation is applicable for the HS distance metric only, as it involves the Hamming distance metric.

For the mixture of CER models IS density, we determine a single centroid $\tilde{A}_{\mathcal{G}^m}$ resulting from fitting the data to the CER model, while for $\tilde{\alpha}$ we consider a range of values within its support $[0, 0.5]$.

4.3.6 Bias with N

Despite the advantages of the implementation of an IS step within the MCMC scheme for the approximation of the normalising constant, there is one limitation with respect to its implementation for the likelihood of the SNF model. As seen in equation (4.29), the likelihood of the SNF model for a set of network observations N, results in the exponentiation of the normalising constant to the power of N. However, under our framework we approximate Z instead of Z^N, as otherwise the implementation of the IS would not be feasible. This leads to an increasing bias of the estimate, as the number of observations N increases.

Vitelli et al. [2017] who also formulate an Importance Sampler for the Mallow's model that has the same functional form with the SNF, acknowledge this limitation and prove that in order for the approximate posterior to converge to the true posterior, the IS sample size K must grow faster than N^2. Thus for large N we should significantly increase the IS sample size K to achieve convergence. In the case of rank data and the Mallow's model, Vitelli et al. [2017] consider only right-invariant distances that allow them to reduce the expression of the normalising constant into an expression that depends only on the dispersion parameter as seen in Section 4.2.3. This allows an Off-line approximation of the normalising constant on a grid of the dispersion parameter α, thus they do not need to draw an IS sample in each iteration of the MCMC as in our case. In this regard, Vitelli et al. [2017] can obtain very large K samples, without a significant computational cost.

On the other hand, for network data and the SNF model, the right-invariance property does not hold for the majority of network distance metrics, as discussed in Section 4.2.3, thus an off-line approximation of the normalising constant cannot be achieved. Therefore, for large N we need to obtain significantly larger IS samples K, which would lead to a significant increase in computational time. To address this issue, we forge an error adjustment for the estimator of the normalising constant under two different approaches.

First, we explore the error of approximating Z^N by $\hat{Z}^N$ using a second order Taylor expansion of Z around $\hat{Z}$, with $f(x) = x^N$. In this section, we write Z instead of $Z(A_{\mathcal{G}^m}, \gamma)$ for the sake of simplicity. Thence, the Taylor expansion for Z^N is:

$$Z^N = \hat{Z}^N + N\hat{Z}^{N-1}(\hat{Z} - Z) + \frac{1}{2}N(N-1)\hat{Z}^{N-2}(\hat{Z} - Z)^2. \tag{4.32}$$

Taking the expectation for both sides of equation (4.32) we have:

$$\begin{aligned} \mathbb{E}(Z^N) = \mathbb{E}(\hat{Z}^N) + N\hat{Z}^{N-1}\mathbb{E}(\hat{Z} - Z) + \frac{1}{2}N(N-1)\hat{Z}^{N-2}\mathbb{E}((\hat{Z} - Z)^2) \Leftrightarrow \\ \mathbb{E}(Z^N) = \hat{Z}^N\left(1 + \frac{N(N-1)}{2\hat{Z}^2}Var(Z)\right), \end{aligned} \tag{4.33}$$

as $\mathbb{E}(\hat{Z}^N) = \hat{Z}^N$ and $\mathbb{E}(\hat{Z} - Z) = 0$. For small $\frac{N(N-1)}{2\hat{Z}^2}Var(Z)$, equation (4.33) can be written as

$$\mathbb{E}(Z^N) = \hat{Z}^N e^{\frac{N(N-1)}{2\hat{Z}^2}Var(Z)}. \tag{4.34}$$

We further consider an alternative way for adjusting the error of the normalising constant approximation as follows. First, we introduce an estimation error term δ,

$$Z = \hat{Z} + \delta. \tag{4.35}$$

If we raise to the power of N and take the logarithm for both sides we have,

$$\begin{aligned} Z^N &= (\hat{Z} + \delta)^N \Leftrightarrow \\ \log(Z^N) &= \log((\hat{Z} + \delta)^N) \Leftrightarrow \\ N\log Z &= N\log\left(\hat{Z}\left(1 + \frac{\delta}{\hat{Z}}\right)\right) \Leftrightarrow \\ N\log Z &= N\log\hat{Z} + N\log\left(1 + \frac{\delta}{\hat{Z}}\right) \Leftrightarrow \\ N\log Z &= N\log\hat{Z} + N\frac{\delta}{\hat{Z}}, \end{aligned} \tag{4.36}$$

for $\frac{\delta}{\hat{Z}}$ small, and $\delta \sim \mathrm{N}(0, \sigma^2_{\hat{Z}})$ given Monte Carlo draws. Thus,

$$N \log Z \sim \mathrm{N}(N \log \hat{Z}, N^2 \sigma^2_{\hat{Z}} / \hat{Z}^2), \tag{4.37}$$

or equivalently Z^N follows a log Normal distribution with expectation

$$E(Z^N) = \hat{Z}^N \cdot e^{N^2 \sigma^2_{\hat{Z}} / 2\hat{Z}^2} \tag{4.38}$$

We observe that both methods for adjusting the error of the normalising constant approximation, result in a very similar expression as seen in equations (4.34) and (4.38).

4.3.7 Challenges with cycle detection

A motivation for our study has been the research questions arising from the fish data example, presented in Section 4.3.1. Under this data example, the formulation of a distance metric that quantifies differences in cycles among networks is essential to address the research questions. However, cycle detection in networks is a computationally intensive task even for moderate-sized networks, resulting in computationally expensive algorithms with respect to the time and memory required to run.

In our study we perform cycle detection using the R package `igraph` (Csardi and Nepusz [2006]), and specifically the *all simple paths* function that detects all the simple paths connecting two nodes. We note here that a *simple path* is a sequence of nodes connecting an origin node to a destination node, without repeated nodes in that sequence. Thus, path detection can easily be extended to cycle detection, where now the only repeated nodes are the first and the last node in the sequence. If a network is not sparse, the number of paths connecting two nodes can grow exponentially depending on the number of nodes in the graph. Specifically, in the case of a complete graph with n nodes, the time complexity for performing path detection is $O(n!)$ (Csardi and Nepusz [2006]), making calculations infeasible.

Under the MCMC scheme described in section 4.3.4, we draw a new IS sample in each iteration of our algorithm to estimate the normalising constant under equation (4.30). The estimation involves the calculation of the HS distance between each network in the newly drawn IS sample and the corresponding proposed and current centroid, as seen in the nominator of the sum in (4.30). Thus, the cycles in each network of the IS sample need to be identified in every iteration that we draw a new IS sample, which can lead to a computationally heavy algorithm.

To avoid the exact cycle calculation in every iteration of our MCMC, we consider an approximation of the symmetric difference distance between the cycles of two networks. The idea behind this formulation is that other network distance metrics that are less computationally intensive to calculate, can be informative for predicting the symmetric difference distance. In this context, various distance metrics can be thought of as covariates in our problem.

There is a range of different frameworks to perform predictions in the Statistics and Machine Learning literature (Yan and Su [2009], Song and Ying [2015], Biau [2012], Friedman et al. [2001]). In our study, we use supervised learning and specifically the xgboost algorithm (Chen and Guestrin [2016]) to predict the symmetric difference of cycles between networks, using the R package `xgboost` (Chen et al. [2021]). The basis of supervised learning algorithms, such as the xgboost, is the use of a training data set to find the values of the parameters that best fit the data, in order to make predictions for a target variable. Xgboost uses decision trees ensembles to perform prediction, through additive training. Additive training refers to the process of sequentially adding new decision trees with ultimate goal the optimisation of an objective function. The objective function comprises of a loss function and a regularization term which controls the model complexity. Boosted trees use a regularization term that performs well in predicting the target variable, and prevents from overfitting. This attribute of the xgboost along with its fast execution render it as a great approach for prediction. However, we note here that other prediction methods could have been equally applied.

In our framework, we train the xgboost in every iteration of the algorithm in order to make accurate predictions for the newly drawn IS sample. To train the xgboost, we use a single training sample of networks for which we calculate the cycles, thus the cycle detection task is performed only once in our algorithm. Thence, we calculate distances between the training sample of networks and the current or proposed centroid network, and train the model to identify associations between the distance metrics considered as predictors in our problem, and the symmetric difference distance which is the target variable. Notably, the distance metrics considered as covariates in our problem are the Hamming distance, the Jaccard distance and the centrality-betweenness distance. For a description of the Hamming and Jaccard distance metrics refer to Section 1.1, and for the centrality-betweenness distance refer to Appendix B.3. Lastly, to calculate the estimator of the normalising constant for the newly drawn IS sample, we obtain the Hamming, the Jaccard and the centrality betweenness distance between each network in the new IS sample and the current or proposed centroid, and predict the symmetric difference distance using the trained xgboost. In this manner, we avoid the calculation of the cycles of the newly drawn IS sample in every iteration.

In Section 4.4.3, we present the results from our simulation study using the xgboost algorithm to predict the symmetric difference against its exact calculation for the HS distance metric, along with the computational time required in each case.

4.4 Simulation Study

In this Section we investigate the behaviour of the newly proposed HS distance metric, as well as the performance of our proposed MCMC with IS step for inferring the parameters of the SNF model. Specifically, in Section 4.4.1 we investigate the HS distance metric behaviour through synthetic data experiments, in Section 4.4.2 we perform simulation studies to assess the performance of our proposed MCMC with IS step, for both the HS and the Jaccard distance, for small network sizes that allow comparisons to the inferences made under the exact calculation of the normalising constant. In addition in Section 4.4.2 we investigate the performance of the MCMC with IS step for various IS densities and IS sample sizes. Lastly, in Section 4.4.3 we investigate the performance of the MCMC with IS step in inferring the model parameters for moderate network sizes, for both the HS and the Jaccard distance, and explore the performance of the xgboost in predicting the symmetric difference of cycles, along with the computational benefit from its formulation.

4.4.1 Synthetic data experiments for HS distance metric

In section 4.3.1, we introduced a new network distance metric, namely the HS distance, formulated to extract information about differences between networks with respect to their cycles. Inspired by the study of Donnat and Holmes [2018] who explore the properties of various network distance metrics from the network literature, in this simulation study we explore the properties of the HS distance through its application on a set of network observations with diverse structures. Specifically, we simulate network data characterised by different topologies, and obtain distance matrices containing the pairwise distances between the simulated networks. Thence, we analyse the distance matrices obtained in the following ways:

- We map the pairwise distances of the networks in a two dimensional space through an MDS projection. This representation allow us to observe whether networks with similar characteristics are identified to be closer with respect to the distance metric specified.

- We obtain the minimum spanning tree graph and apply a Friedman-Rafsky test, as presented in Donnat and Holmes [2018]. This hypothesis testing framework allows us to examine whether the distance metric under consideration captures similarities between networks with respect to their structure.

First, we generate network observations under three well-known network models that exhibit diverse structures, similarly to the synthetic data experiments presented in Donnat and Holmes [2018]. In our study, we consider networks with $n = 20$ nodes, motivated by the network sizes of the fish data application. The network models used to simulate network data, along with the parameter specification are the following:

1. The Erdos-Renyi (ER) model with probability of connection $p = 0.1$

2. The Preferential Attachment (PA) model with power equal to 1.

3. The Stochastic Block Model (SBM) with two equally sized communities of nodes and probability matrix $\begin{pmatrix} 0.2 & 0.01 \\ 0.01 & 0.3 \end{pmatrix}$.

For each network model specification, we simulate 10 independent network observations, resulting in a set of $N = 30$ networks. We further fix the density of the graphs to be equal to 0.1, so that the variability of the simulated networks originates only from the networks' structure.

As presented in section 4.3.1, the HS distance metric involves a weighting factor λ indicating the importance of the symmetric difference on the overall distance between two networks. In this respect, we consider a range of values for the weighting factor

$\lambda = \{0.1, 0.25, 0.5, 0.75, 1, 1.5, 2, 4, 8\}$, to explore its impact on the distance metric performance. To further understand the behaviour of the HS distance, we also consider two widely used network distance metrics, the Hamming and the Jaccard distance, to serve as a reference for comparing the results obtained from the HS distance. Lastly, we obtain the Symmetric difference distance solely, to observe its behaviour when we do not combine it with the Hamming distance in the HS distance metric.

Figure 4.3 shows the MDS projections for each distance metric considered, where each dot corresponds to a network observation and the colour of the dots correspond to the network model used to simulate the networks. For both the Hamming and the Jaccard MDS projections, we notice that the networks simulated from the same model are positioned closer in the 2-d plane, while for the Symmetric difference of the cycles, the networks are clustered together irrespectively of the network structure, with some networks placed away from that cluster. This finding indicates that it is hard to identify similarities among networks with respect to their structure, considering solely the Symmetric difference distance.

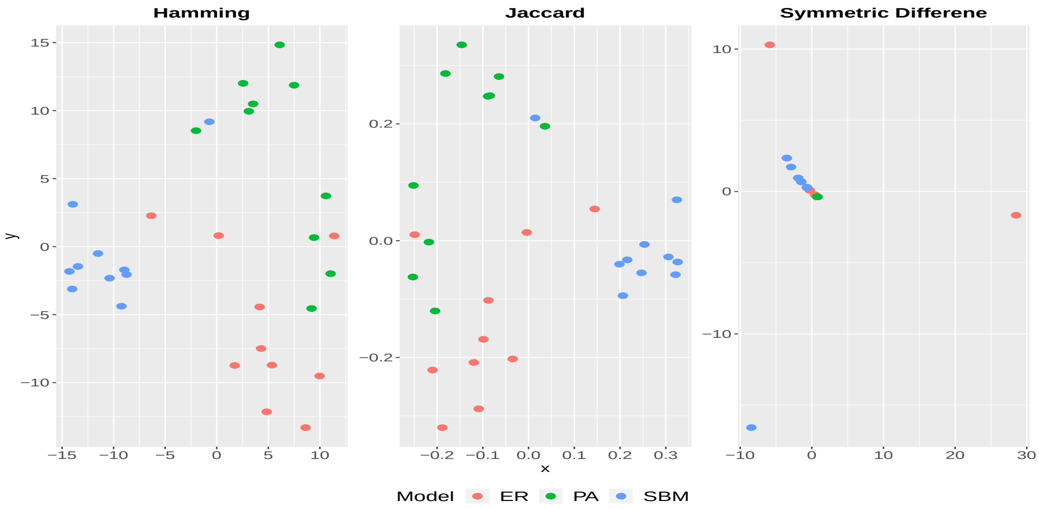

Figure 4.3: MDS for the Hamming distance matrix (left), the Jaccard distance matrix (middle) and Symmetric difference distance matrix (right), for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM).

In Figure 4.4, we show the MDS projections for the HS distance, for different λ weights for the Symmetric difference part of the metric. We notice that as the impact of the Symmetric difference distance increases over the Hamming distance, the networks tend to cluster together in one group. This is an anticipated result considering the

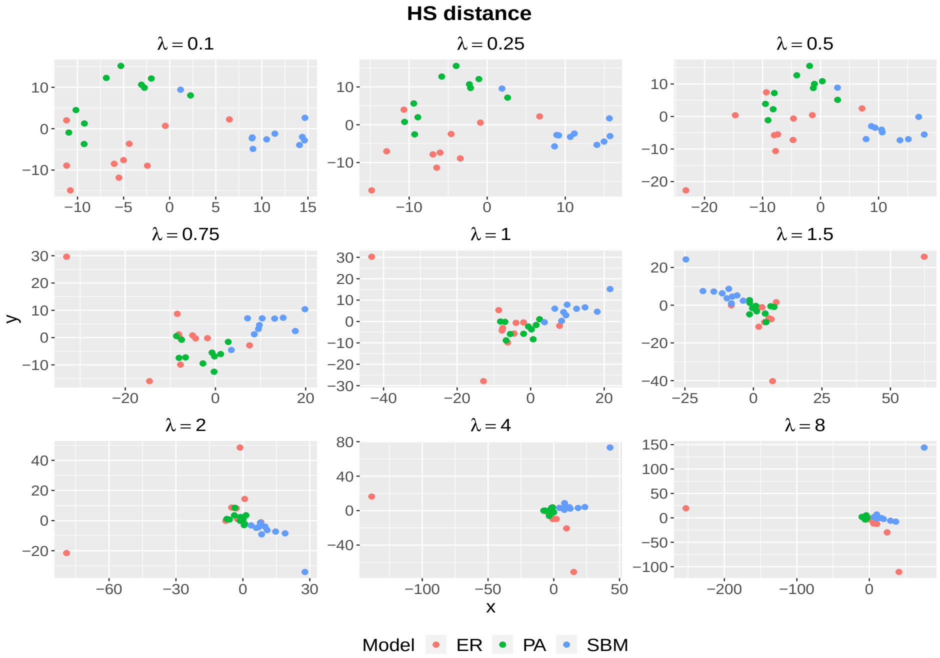

Figure 4.4: MDS for the HS distance matrix for varying sizes of λ, for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM)

behaviours of the Hamming and Symmetric difference distance independently, as seen in MDS projections in Figure 4.3. It is worth noting that as λ increases, the first network data that start to form a cluster are those simulated from the ER and PA models, while the SBM networks differentiate from that cluster, even for greater λ values. This result highlights the importance of combining the Hamming distance with the Symmetric difference to form the HS distance metric, as the Hamming distance captures information about the overall structure of the networks. In addition, the illustration in Figure 4.4 shows the importance of the weight factor λ in the overall behaviour of the HS distance.

We further identify other characteristics of the simulated networks, to investigate whether the distance metrics considered are able to detect similarities among networks with similar characteristics. Network characteristics summarise information about structural properties of the networks on both node and edge level. Maugis et al. [2017] show that two dominant network features that reveal information about a network's topology are trees and cycles. The latter is also a network characteristic that we aim to capture with the HS distance metric formulated. In this regard, we calculate the number of cycles of all sizes and star-trees of size 4 in the simulated networks, and illustrate the results through the MDS projections obtained from the distance metrics.

Figures 4.5 and 4.6 show the MDS projections for the networks, where colours correspond to the number of cycles identified in the networks and shapes correspond to the network model used to simulate the networks. In Figure 4.5, we observe that the Hamming and the Jaccard distance identify similarities among networks that have less cycles, while the networks having 5 up to 25 cycles are not distinguishable. On the other hand, the Symmetric difference identifies as dissimilar the networks with the largest number of cycles, from the rest of the networks. We further notice that the networks with the less cycles are those simulated under the PA model as expected due to their tree-like shape, while the networks simulated under the ER and the SBM enclose varying number of cycles.

For the HS distance metric in Figure 4.6, we notice that as we increase the size of λ the networks with the less cycles start to group together, while the networks involving a large number of cycles are still identified as dissimilar to all the other networks, as seen from the three networks consistently positioned away from all other networks in the 2-d plane. This is a sensible result if we consider that the more cycles two networks have, it is likely that the less cycles will have in common.

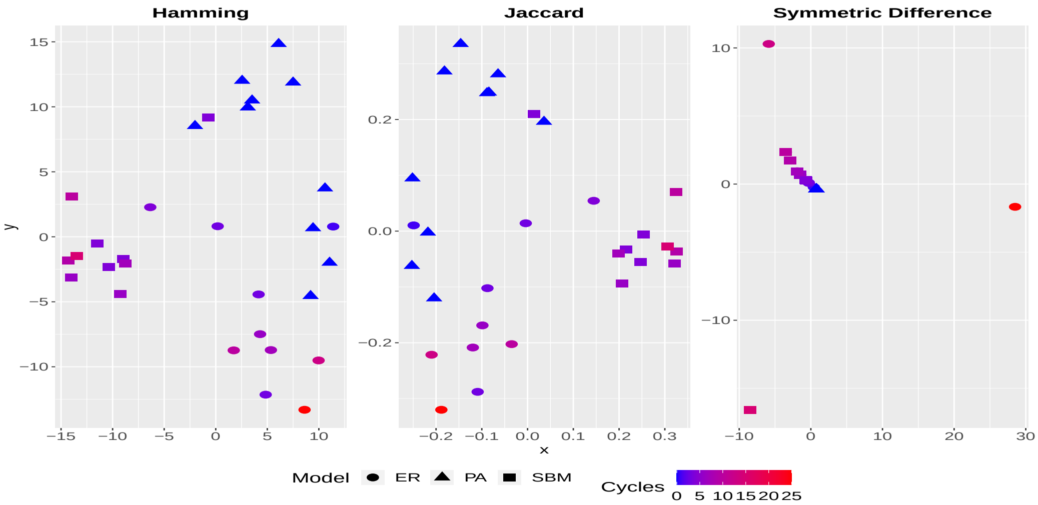

Figure 4.5: MDS for the Hamming distance matrix (left), the Jaccard distance matrix (middle) and Symmetric difference distance matrix (right), for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM), with colours corresponding to number of cycles.

We now detect the 4-node star-tree motifs (Figure 4.7) formed in the simulated networks, and present the results through the MDS projections in Figures 4.8 and 4.9. We observe that the Hamming and the Jaccard distance display a similar behaviour with

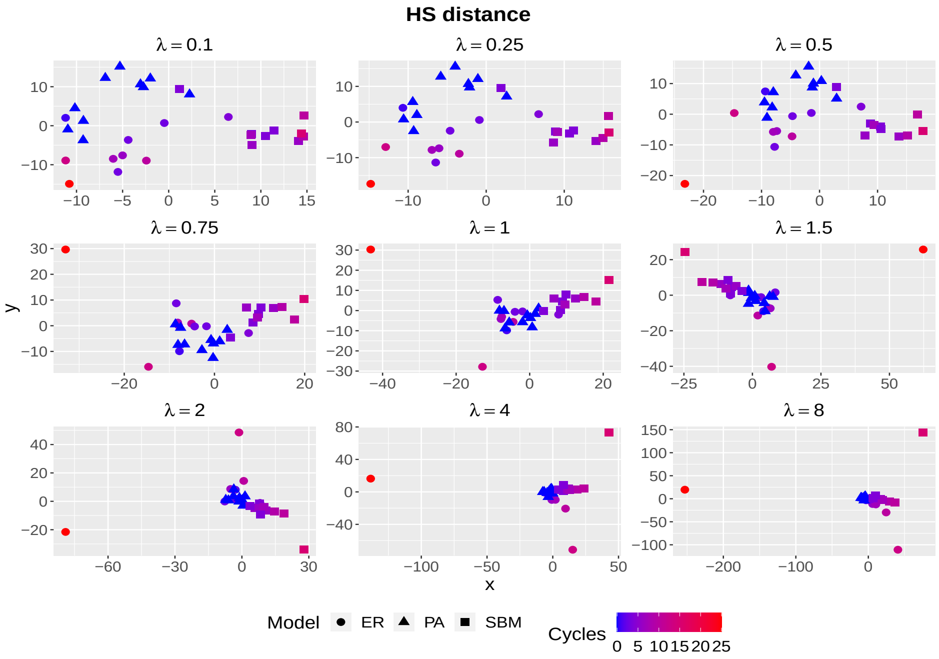

Figure 4.6: MDS for the HS distance matrix for varying sizes of λ, for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM), with colours corresponding to number of cycles

respect to the similarities identified among the networks enclosing varying numbers of star-trees. Overall, we observe that both metrics do not strongly group together networks with similar amount of star-trees. However, we notice that the networks with the largest number of star-trees tend to be identified as more similar, and they are distinguishable from the networks with the less star-trees. Instead, the Symmetric difference identifies as identical the networks with the largest number of stars, as can be deduced by their overlapping positions in the 2-d plane. The latter can also be noticed for the HS distance metric in Figure 4.9, for which we observe that increasing λ leads to the identification of the networks with the largest number of star motifs as identical, as opposed to the rest of the networks which are more dispersed in the 2-d plane. This finding suggests that the HS distance can detect similarities among networks characterised by many star-tree motifs.

Another informative summary about the networks' topology, is the networks' transitivity which expresses the probability that connected triples in a graph close to form triangles, as previously discussed in Section 1.1. In Figures 4.10 and 4.11, we present the MDS projections with respect to the networks' transitivity. We notice that the Hamming and the Jaccard distance tend to detect similarities between networks having close

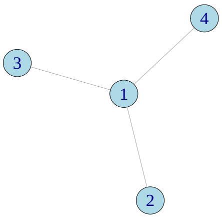

Figure 4.7: 4-node star-tree motif.

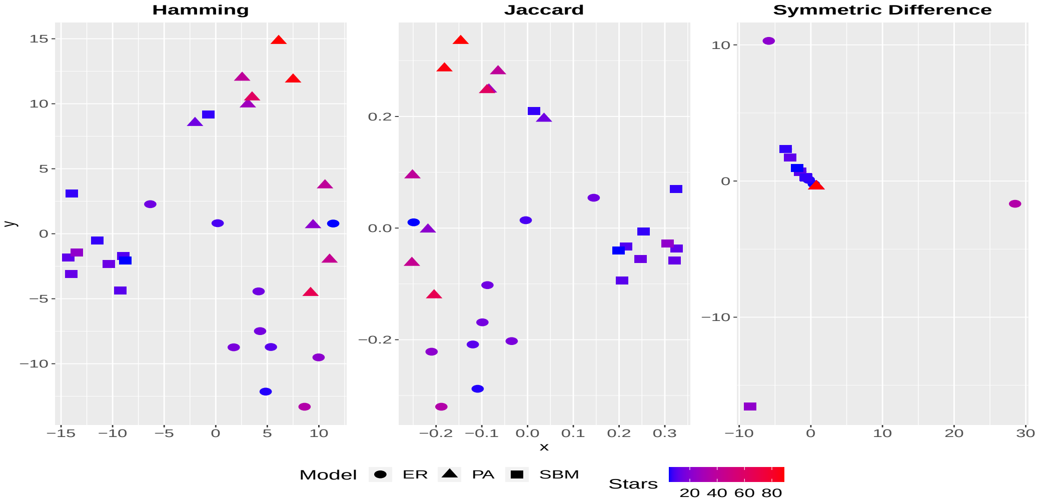

Figure 4.8: MDS for the Hamming distance matrix (left), the Jaccard distance matrix (middle) and Symmetric difference distance matrix (right), for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM), with colours corresponding to number of stars.

transitivity probability. However, Symmetric difference is not consistently grouping together graphs that have close transitivity probability, as can be noticed for the networks positioned away from the group of networks forming a cluster.

Alternatively, the HS distance metric appears to identify similarities among graphs with close transitivity probabilities as seen in Figure 4.11. Specifically, increasing the importance of the Symmetric difference on the overall metric, assists in identifying similarities between networks with small transitivity probability, while simultaneously networks with moderate to higher transitivity also group together. This becomes more prominent for $\lambda = 0.75$ for which we can clearly see networks grouping according to their transitivity.

We now describe the Friedman-Rafsky test performed on the minimum spanning tree induced by the distance matrices, as presented in Donnat and Holmes [2018]. Under this

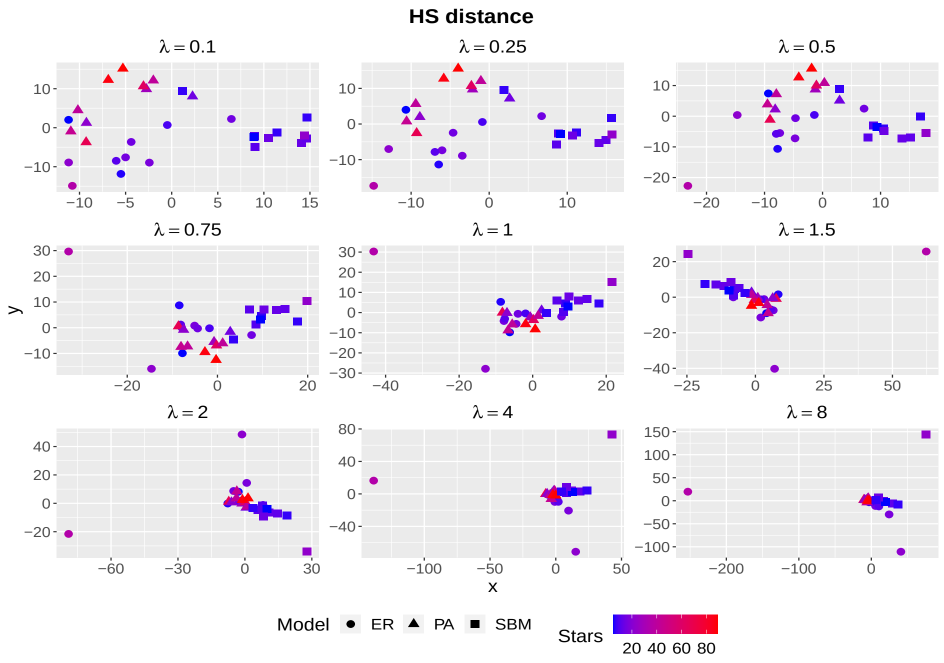

Figure 4.9: MDS for the HS distance matrix for varying sizes of λ, for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM), with colours corresponding to number of stars.

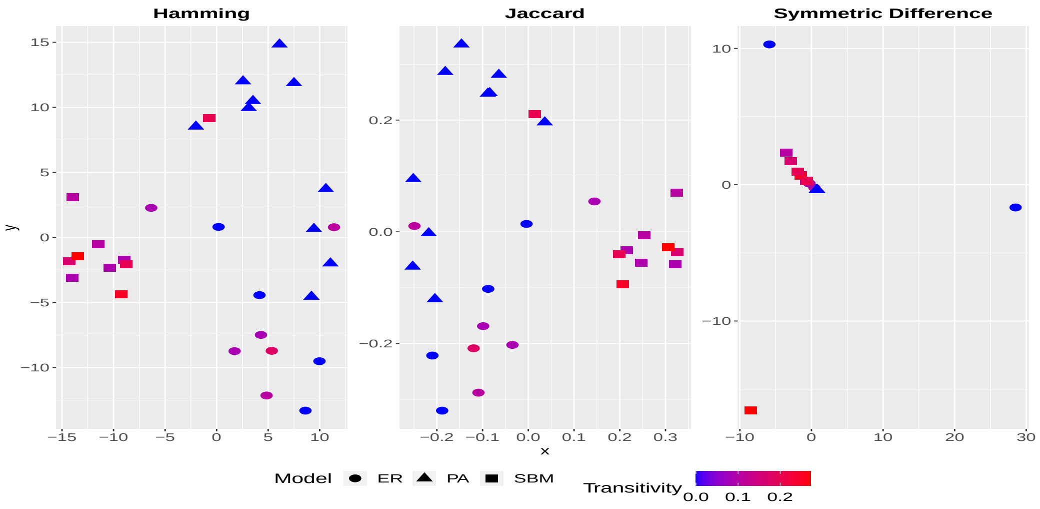

Figure 4.10: MDS for the Hamming distance matrix (left), the Jaccard distance matrix (middle) and Symmetric difference distance matrix (right), for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM), with colours corresponding to transitivity measure.

framework we want to test whether the distance metrics are able to identify groups of networks with respect to their structure. Thus, in our case we have three different groups corresponding to the three different network models used to simulate the networks. By performing this test we are able to quantify the uncertainty of what we observed from the MDS projections.

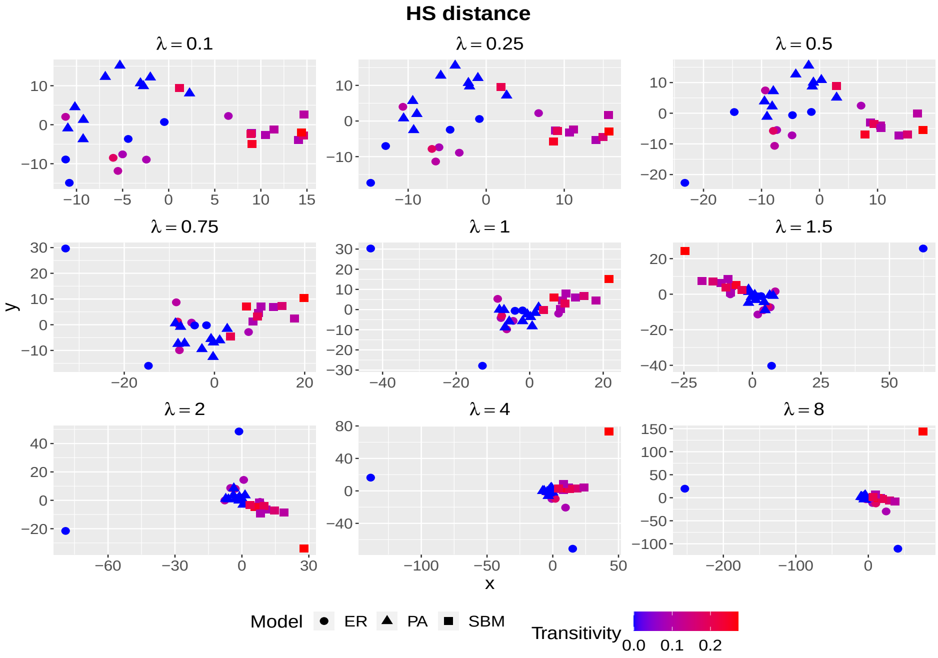

Figure 4.11: MDS for the HS distance matrix for varying sizes of λ, for networks generated under the Erdos-Renyi (ER), the Preferential Attachment (PA) and the Stochastic Block Model (SBM), with colours corresponding to transitivity measure.

As commented above, to perform the Friedman-Rafsky test we first obtain the minimum spanning tree (MST) resulting from a distance matrix. The minimum spanning tree is a connected graph with nodes representing networks, and edges connecting networks with the greatest similarities, such that no cycles are being formed. Figures 4.12 and 4.13 show the minimum spanning trees obtained for the Hamming, the Jaccard, the Symmetric difference and the HS distance matrices, for $\lambda = \{0.25, 1, 2\}$ indicatively. The colours of the graphs' nodes correspond to the three different network models. We observe that the minimum spanning trees obtained for the Hamming, the Jaccard and the HS distance, mostly connect graphs simulated under the same network model, while this is clearly not the case for the Symmetric difference distance.

We further investigate that by performing a Friedman-Rafsky test in the following way. First, we derive the test statistic for our sample, which is the number of edges connecting graphs of the same group (network model), for each minimum spanning tree. Then, we construct the sampling distribution of the test statistic by obtaining 50,000 permutations of the node labels of each minimum spanning tree graph, while keeping fixed the graph's topology. Thence we obtain 50,000 graphs from the node label permutation, and for each graph we calculate the number of edges connecting graphs of

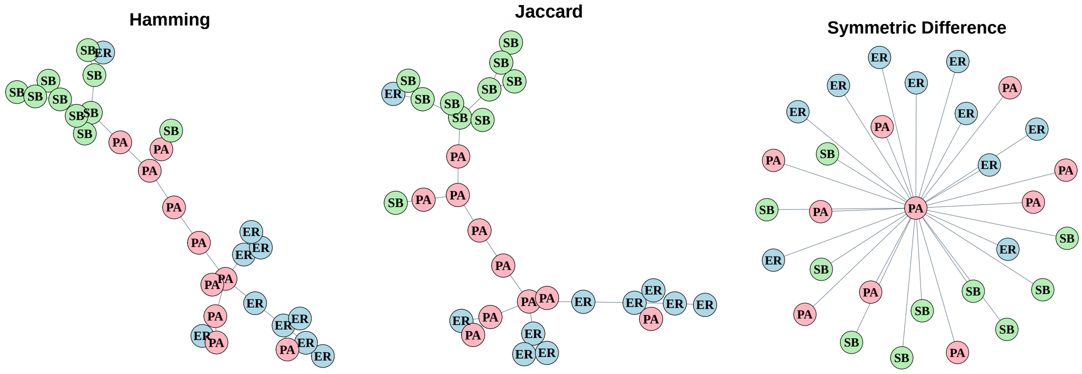

Figure 4.12: MST for the Hamming distance matrix (left), the Jaccard distance matrix (middle) and the Symmetric difference distance matrix (right).

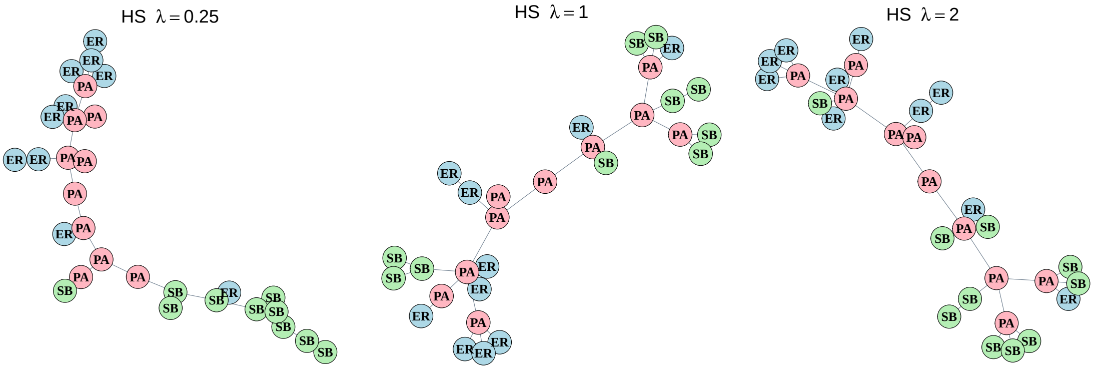

Figure 4.13: MST for the HS distance matrix with $\lambda = 0.25$ (left), $\lambda = 1$ (middle) and $\lambda = 2$ (right).

the same group. This give us the probability of seeing a test statistic at least as extreme as the one observed, namely the p-value of our test. If the p-value is significantly small, then we can claim that there is evidence suggesting that the corresponding distance metric groups together graphs simulated from the same network model.

In Table 4.1, we report the p-values obtained under the Friedman-Rafsky test, formulated for the minimum spanning trees of the Hamming, the Jaccard and the HS distance for $\lambda = \{0.25, 1, 2\}$. We comment here that it is not meaningful to obtain the Friedman-Rafsky test for the minimum spanning tree of the Symmetric difference distance, as any label permutation would result in the same test statistic due to the graph's topology seen in Figure 4.12 (right). We observe that the p-values derived for all the distance metrics are statistically significant, except for the HS distance with $\lambda = 2$. This result supports the findings from the MDS projections which showed that the Hamming and the Jaccard are able to detect similarities among networks of the same group. Moreover, the p-values obtained for various lambda values of the HS distance indicate the impact of the weight on the overall behaviour of the HS distance. Notably, we observe that the HS distance with lambda=0.25 and 1 is able to detect similarities among networks belonging in the same group.

Distance metric	**FR p-values**
Hamming	0.0000
Jaccard	0.0000
HS $\lambda = 0.25$	0.0001
HS $\lambda = 1$	0.0732
HS $\lambda = 2$	0.2584

Table 4.1: p-values for Friedman-Rafsky test applied on the MSTs for the Hamming, the Jaccard and the HS distance.

4.4.2 Performance of MCMC for small network sizes

In this section we investigate the performance of our algorithm in inferring the model parameters, for small networks sizes. Specifically, we consider networks with 5 nodes, as the space of graphs for this size of networks $\{\mathcal{G}_{|5|}\}$ encompasses $2^{10} = 1024$ graphs, rendering the calculation of the true normalising constant feasible. This is an important condition in order to investigate the performance of our MCMC algorithm with the IS step, as it enables us to compare its results against the results obtained from the implementation of an MCMC algorithm with the true normalising constant calculated within the MH ratio.

To explore the behaviour of our MCMC algorithm with IS step for the SNF model, we consider the Jaccard and the HS distance metrics, and determine various regimes of the dispersion parameter γ and network population sizes N. The network population sizes that we consider in this simulation study are $N = \{5, 20, 50\}$. To determine the sizes of γ, we first need to investigate the behaviour of the SNF model under each distance metric specification.

To achieve that we conduct some Exploratory Data Analysis by sampling networks from the SNF model with fixed parameters $\mathcal{G}^0$ and γ_0. Sampling networks from the SNF model is accomplished by implementing an MH algorithm that updates network data in each iteration, while keeping fixed the SNF model parameters, as presented in Lunagómez et al. [2021]. In Section 4.3.3, we presented how sampling networks from

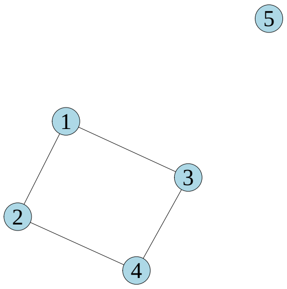

Figure 4.14: 5-node centroid $\mathcal{G}^0$.

the CER model using an MH algorithm is performed, which is similar to the sampling scheme used for the SNF model in this simulation study. We note here that under this set up, the normalising constant cancels from the MH ratio, making the implementation of the MH algorithm straightforward.

Under this formulation, we generate a 5-node network playing the role of the centroid $\mathcal{G}^0$ and simulate network data from the SNF model for various sizes of the dispersion parameter γ_0. In Figure 4.14 we show the 5-node centroid $\mathcal{G}^0$ simulated. Subsequently, we obtain the distance between the simulated network data and the centroid $\mathcal{G}^0$, for each regime of γ_0 considered, with respect to the distance metric specified. We visualise the results using boxplots representing the distribution of the distances for each size of γ_0. This visualisation allows us to observe how the distribution of the distance changes for different sizes of γ_0, and for each distance metric specified. In Figures 4.15 and 4.16, we present the EDA results for the HS and the Jaccard distance, respectively.

We observe that the average distance $\mathbb{E}[d(\mathcal{G}, \mathcal{G}^0)]$ scales differently with γ_0, for each distance metric specification. Notably, $\mathbb{E}[d(\mathcal{G}, \mathcal{G}^0)]$ reaches 0 for different sizes of γ_0 for each distance metric considered. In addition different step sizes for γ_0 impose more or less drastic changes in the distribution of distance according to the distance metric specified. For example, step sizes similar to that obtained for the HS distance (e.g. step equal to 0.1), would impose less drastic changes to the distance distribution for the Jaccard distance metric.

In the following subsections, we demonstrate the simulation regimes for the HS and the Jaccard distance informed by the EDA, and present the results from our simulation study.

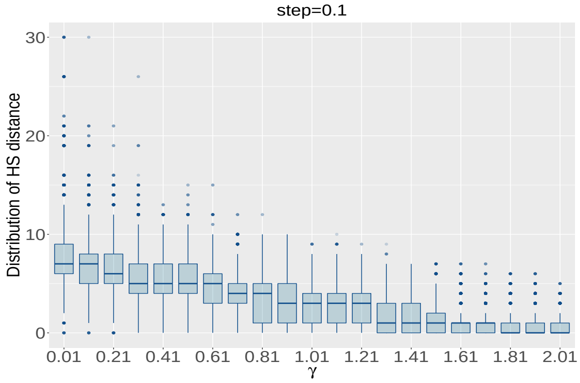

Figure 4.15: Distribution of the HS distance between networks generated from the SNF($\mathcal{G}^0$,γ_0) for various sizes of γ_0, for 5-node networks.

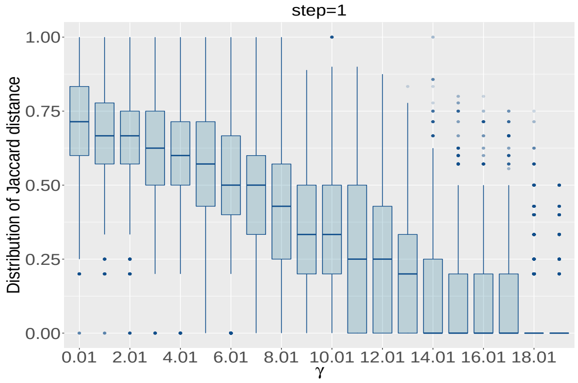

Figure 4.16: Distribution of the Jaccard distance between networks generated from the SNF($\mathcal{G}^0$,γ_0) for various sizes of γ_0, for 5-node networks.

Simulation study for HS distance metric

Under the HS distance metric specification in 4.15, we observe that the distances between the network data simulated and the centroid are close to 0 for $\gamma = 1.6$. Thus, network data simulated for $\gamma_0 > 1.6$ are closer to the centroid, resulting in less variability in the network population generated for those regimes. Considering the EDA results, we set up our simulation study for $\gamma = \{0.01, 0.6, 1.1, 1.6\}$, to investigate the performance of our algorithm in inferring the model parameters for a range of network populations, with respect to their variability from the centroid. In addition, we consider two sample sizes K for the IS step, equal to 2000 and 4000, to explore the effect of the IS sample size in the inference task for the 5-node network case. In Table 4.2, we present the simulation regimes considered for the HS distance metric for the 5-node networks.

We generate network data for each simulation regime specified, and fit them to the SNF model under three different MCMC schemes, (i) an adaptive MCMC scheme for the specification of the parameters of the IS density, (ii) a non-adaptive MCMC scheme,

n	N	K	γ
5	5	2000	0.01
5	20	2000	0.01
5	50	2000	0.01
5	5	4000	0.01
5	20	4000	0.01
5	50	4000	0.01
5	5	2000	0.6
5	20	2000	0.6
5	50	2000	0.6
5	5	4000	0.6
5	20	4000	0.6
5	50	4000	0.6
5	5	2000	1.1
5	20	2000	1.1
5	50	2000	1.1
5	5	4000	1.1
5	20	4000	1.1
5	50	4000	1.1
5	5	2000	1.6
5	20	2000	1.6
5	50	2000	1.6
5	5	4000	1.6
5	20	4000	1.6
5	50	4000	1.6

Table 4.2: Simulation regimes for 5 node networks for the HS distance metric.

in which the parameters of the IS density are held fixed throughout the iterations, and (iii) an MCMC scheme in which the true normalising constant is calculated. For the adaptive scheme (i), we implement the MCMC with an IS step as presented in section 4.3.5. For the non-adaptive scheme (ii), we set the parameters of the CER model to the posterior mean of α and the posterior mode of $\mathcal{G}^m$ after fitting the simulated network data to the CER model. The formulation of scheme (iii) is straightforward as we are able to identify the networks in the space of 5-node graphs $\{\mathcal{G}_{|5|}\}$.

We further specify a Gamma prior distribution for the dispersion γ, with hyperparameters informed by the corresponding scale of γ, as proposed in Lunagómez et al. [2021]. The prior specified for the centroid $\mathcal{G}^m$ is the SNF model as presented in section 4.3.4, with hyperparameters $\mathcal{G}_0$ and γ_0. We set $\mathcal{G}_0$ to the network from the simulated network data that minimises the distance from all the other networks, and γ_0 to a low value so that the we allow high variation to be encoded.

In Figure 4.17, we visualise the posterior distribution for the dispersion parameter γ obtained after running our MCMC for 10,000 iterations with 1,000 burn-in. We notice that the results obtained from the MCMC with IS step under the two different schemes, are similar to the results obtained for the MCMC with the true normalising constant calculated in each iteration. This is an interesting result, suggesting that the IS step implemented provides a good approximation of the normalising constant. In addition, we observe similar inferential results under the two different IS schemes formulated, except for the regime of $\gamma = 1.6$ and $N = 50$, where the adaptive scheme seems to perform better considering the results from the MCMC with the exact calculation of the normalising constant.

In Figures 4.18, 4.19 and 4.20, we further obtain the traceplots for γ to probe the mixing of the chains under each inferential scheme, and for the diverse simulation regimes. Specifically, each subfigure corresponds to the posterior draws obtained for γ under each simulation regime, while the colours of the traceplots correspond to the different inferential schemes implemented. Notably, we observe that for each simulation regime, all MCMC chains are mixing well and explore the same region of values for γ, close to its true value.

We investigate the performance of the MCMC with IS step algorithm in inferring the centroid network $\mathcal{G}^m$, by obtaining the posterior mode from the MCMC draws for $\mathcal{G}^m$, calculating the posterior mass that the mode concentrates and computing the Hamming distance between the posterior mode and the true centroid to detect how accurately the centroid has been inferred. We summarise the results through the plots seen in Figures 4.21, 4.22, 4.23 and 4.24, for each regime of γ considered and $K = 2000$.

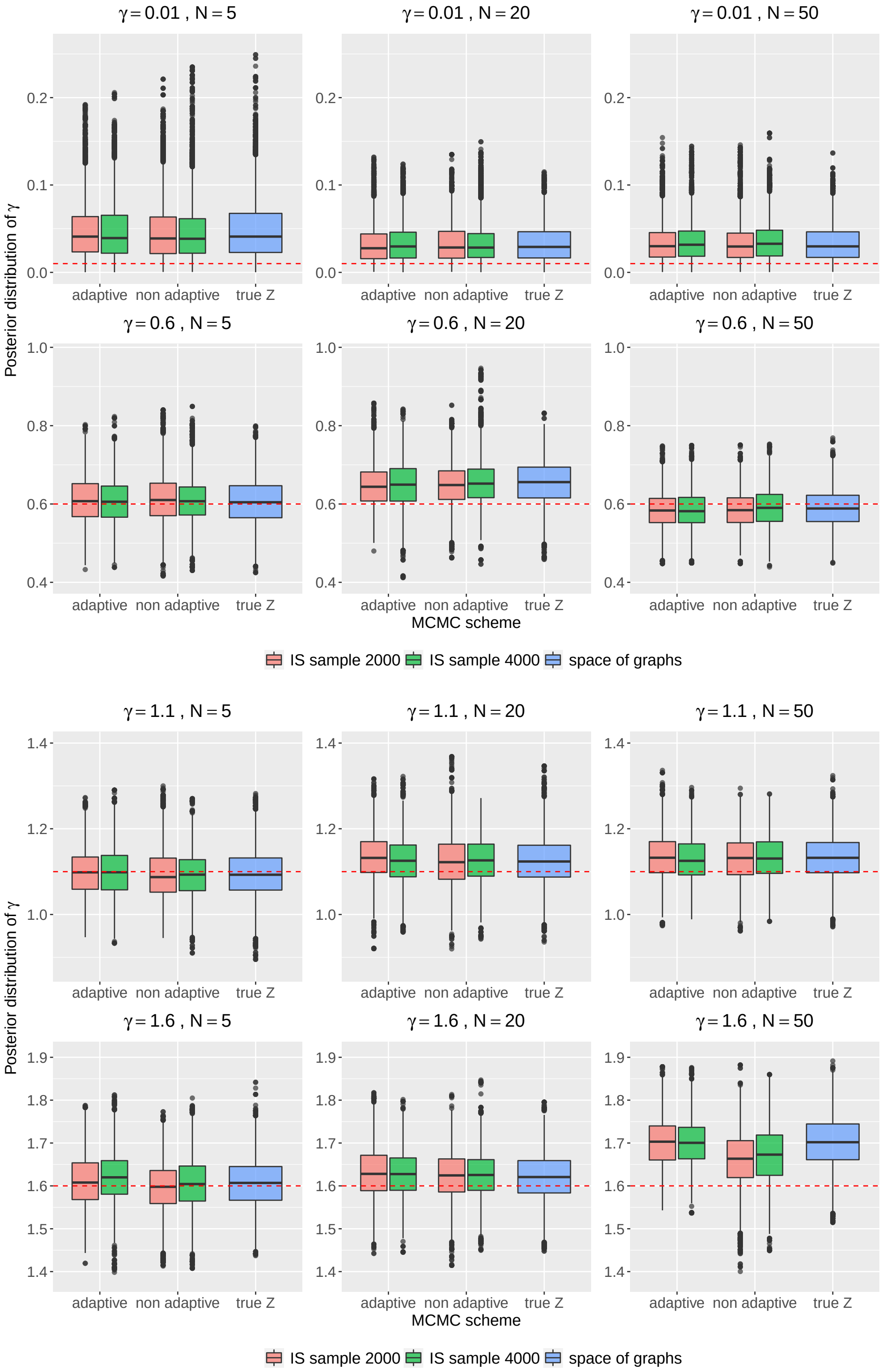

Figure 4.17: Posterior distribution of γ for simulation regimes presented in Table 4.2, after applying the adaptive MCMC scheme, the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant. The red dashed line indicates the true value of γ.

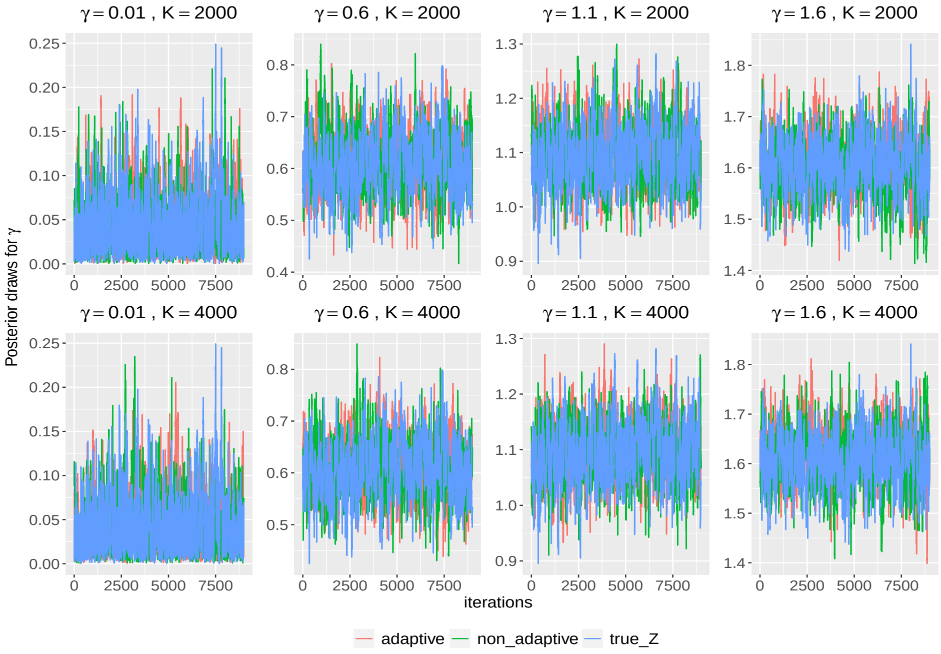

Figure 4.18: Traceplots for γ for simulation regimes presented in Table 4.2 with sample size $N = 5$, after applying the adaptive MCMC scheme, the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant.

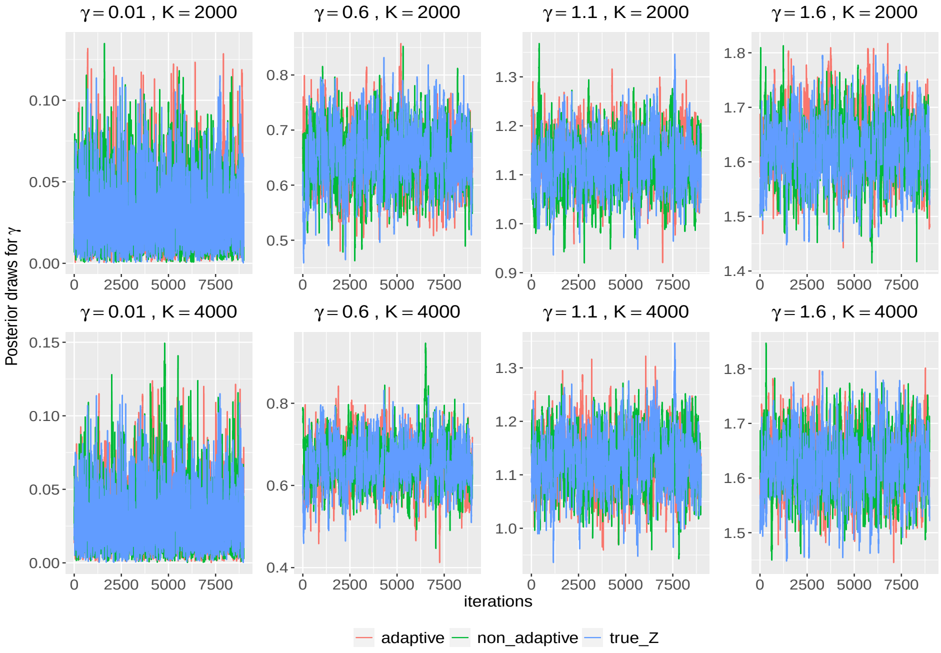

Figure 4.19: Traceplots for γ for simulation regimes presented in Table 4.2 with sample size $N = 20$, after applying the adaptive MCMC scheme, the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant.

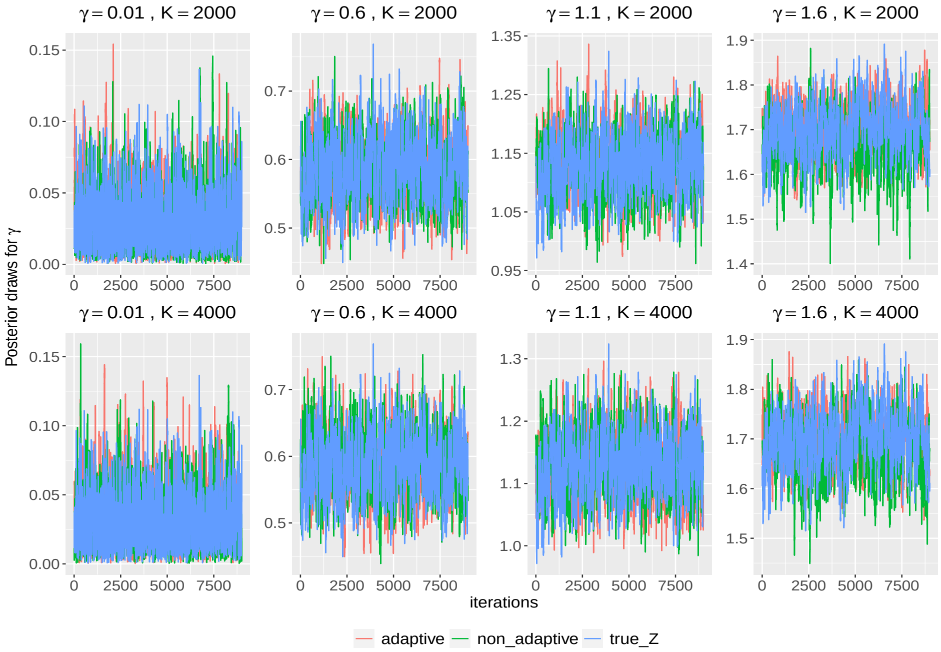

Figure 4.20: Traceplots for γ for simulation regimes presented in Table 4.2 with sample size $N = 50$, after applying the adaptive MCMC scheme, the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant.

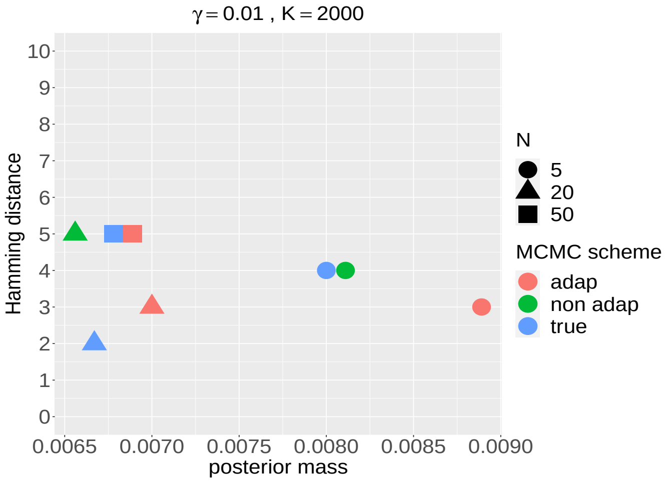

Figure 4.21: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 0.01$ and $K = 2000$, and various sample sizes N and MCMC schemes.

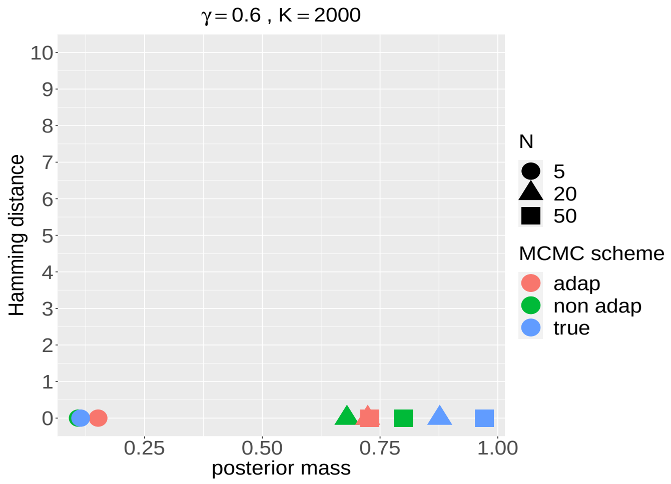

Figure 4.22: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 0.6$ and $K = 2000$, and various sample sizes N and MCMC schemes.

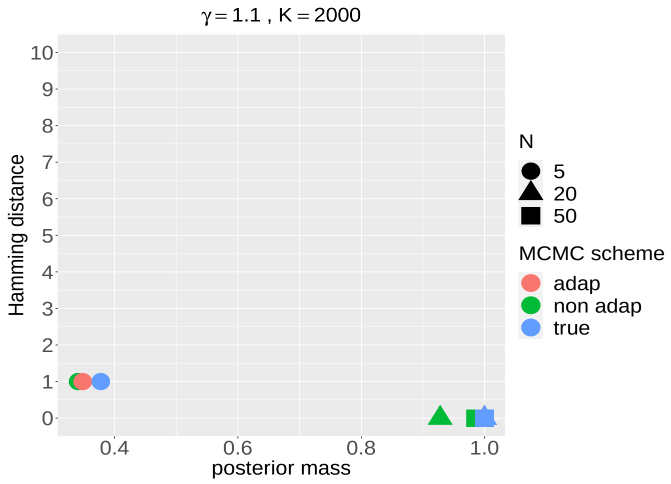

Figure 4.23: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 1.1$ and $K = 2000$, and various sample sizes N and MCMC schemes.

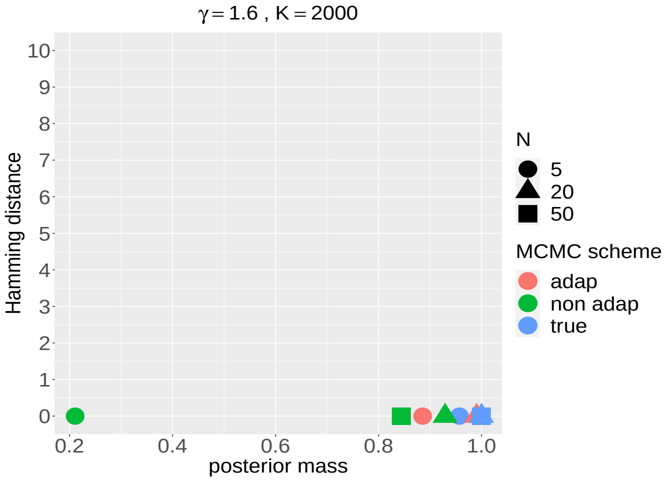

Figure 4.24: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 1.6$ and $K = 2000$, and various sample sizes N and MCMC schemes.

For the regime of $\gamma = 0.01$ corresponding to the populations of networks with the greatest variability from the centroid, we note that the centroid is not fully identified and the posterior modes concentrate low masses ranging from 0.006 to 0.009, for all the different MCMC schemes implemented. On the other hand, for the remaining simulation regimes of γ, the true centroid is accurately inferred from the MCMC, with the posterior modes having a Hamming distance from the true centroid equal to 0. Nonetheless, the posterior masses concentrated by the modes are substantially lower for the smallest population size $N = 5$. We further note here that the inferential results are the same between the MCMC with the IS step and the MCMC with the calculation of the true normalising constant.

In Appendix B.4.1, we further present the autocorrelation plots for γ under the simulation regimes $\gamma = \{0.01, 0.6, 1.1, 1.6\}$ and $N = 50$, for both IS sample sizes $K = \{2000, 4000\}$ and the two distinct IS schemes (adaptive and non-adaptive scheme).

In the next section we set up the simulation study for the 5-node network case and the SNF model with the Jaccard distance metric specified, and present the simulation results after applying the MCMC with the IS step and the MCMC with the true normalising constant obtained in each iteration.

Simulation study for Jaccard distance metric

We now specify the simulation regimes for the 5-node networks case and the Jaccard distance metric specification, utilising the information obtained from the boxplots in Figure 4.16. Similarly to the HS distance, we consider a range of values for γ, such that each network population generated has different variability from the centroid. Notably, we consider $\gamma = \{1, 7, 11, 15\}$, as presented in Table 4.3.

n	N	K	γ
5	5	2000	1
5	20	2000	1
5	50	2000	1
5	5	4000	1
5	20	4000	1
5	50	4000	1
5	5	2000	7
5	20	2000	7
5	50	2000	7
5	5	4000	7
5	20	4000	7
5	50	4000	7
5	5	2000	11
5	20	2000	11
5	50	2000	11
5	5	4000	11
5	20	4000	11
5	50	4000	11
5	5	2000	15
5	20	2000	15
5	50	2000	15
5	5	4000	15
5	20	4000	15
5	50	4000	15

Table 4.3: Simulation regimes for 5-node networks for the Jaccard distance metric.

In this simulation study, for each network population generated according to the simulation regimes seen in Table 4.3, we implement the following two MCMC schemes: (i) a non-adaptive MCMC scheme with an IS step, where we keep fixed the parameters of the IS density, as seen in the simulation study for the HS distance, and (ii) an MCMC scheme in which we calculate the true normalising constant in each iteration. We note here that the adaptive scheme presented in Section 4.3.5 and implemented in the simulation study for the HS distance is not implemented for the Jaccard distance as well, as the transformation applied involves the Hamming distance metric, which is not relevant to the Jaccard distance considered in this section.

We tune the prior distributions for γ and $\mathcal{G}^m$ in the same way as presented in the simulation study for the HS distance. However, in this simulation study we also consider a non-informative prior for the dispersion parameter γ, in order to demonstrate that our algorithm performs equally well under both set ups. We note that obtaining Jeffreys prior is not straightforward for the SNF model, due to the sum over the space of graphs that the likelihood involves. Alternatively, we consider a Uniform prior over an interval $[a, b]$, specified according to the EDA conducted for the SNF model. In this regard, for the Jaccard distance, we consider a Uniform distribution over the interval $[0.01, 18]$. In Appendix B.4.2, we present the results under the non-informative prior specification, for $N = 50$ indicatively.

We run each MCMC algorithm for 10,000 iterations with a burn-in of 1,000 iterations, and visualise the results for γ in Figures 4.25, 4.26, 4.27, 4.28 and 4.29. As commented previously, our ability to compare the results from the MCMC with the IS step to the MCMC with the true normalising constant calculated is key in order to validate the performance of the IS in approximating the normalising constant. We notice that the posterior distribution obtained under the two distinct MCMC schemes are similar, suggesting that our MCMC with IS step performs well in inferring γ.

We summarise the inferential results for $\mathcal{G}^m$ in Figures 4.30, 4.31, 4.32 and 4.33. For each MCMC scheme and each simulation regime considered, we demonstrate the posterior mass concentrated by the mode of $\mathcal{G}^m$ along with its Hamming distance from the true centroid. We notice that for the simulation regime of $\gamma = 1$, the true centroid is not fully identified and the posterior modes obtained concentrate low masses, which can be attributed to the high variability of the network population generated. In contrast to this, for $\gamma = \{7, 11, 15\}$ the posterior modes are identical to the true centroid and concentrate high posterior masses.

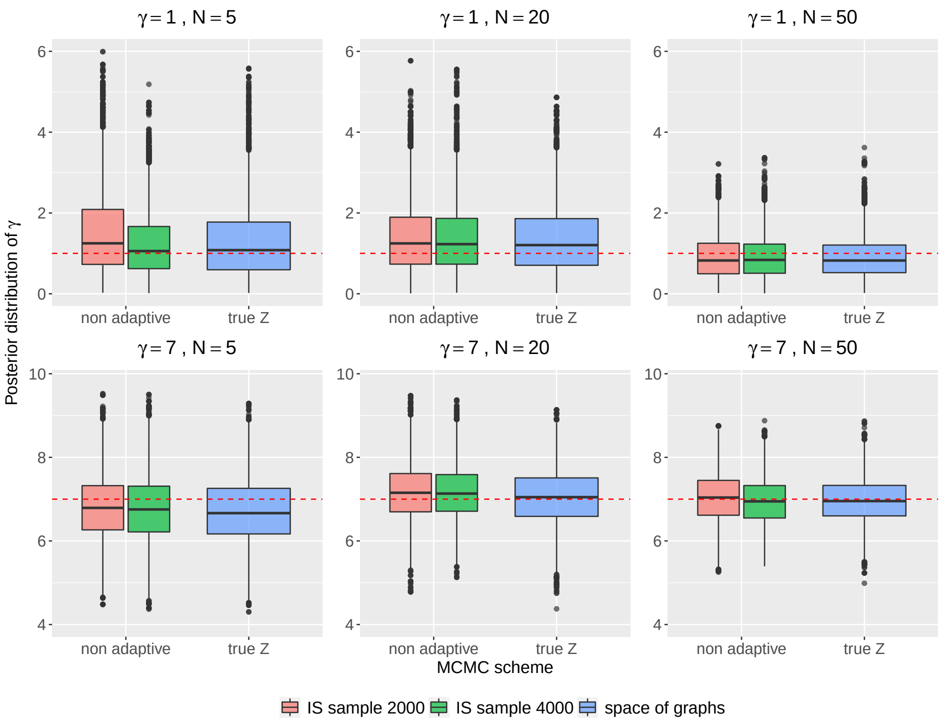

Figure 4.25: Posterior distribution for γ for simulation regimes presented in Table 4.3, after applying the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant. The red dashed line indicates the true value of γ.

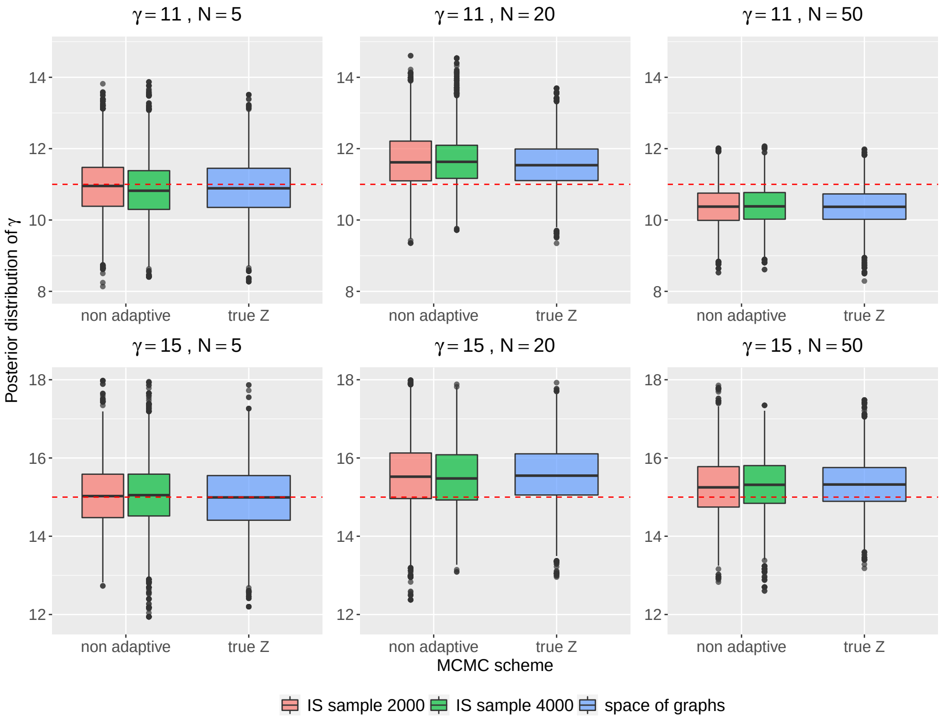

Figure 4.26: Posterior distribution for γ for simulation regimes presented in Table 4.3, after applying the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant. The red dashed line indicates the true value of γ.

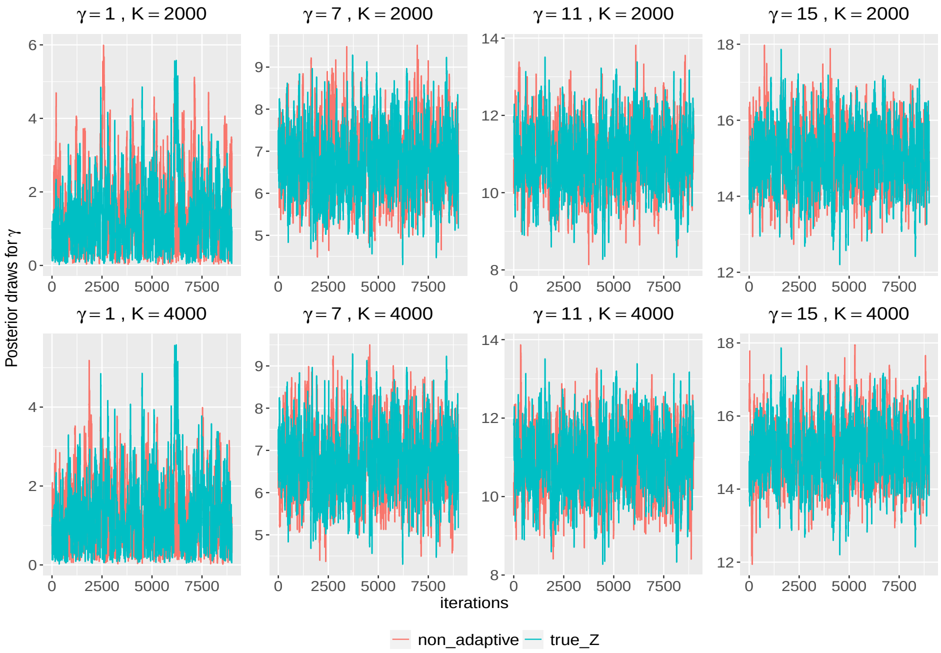

Figure 4.27: Traceplots for γ for simulation regimes presented in Table 4.3 with sample size $N = 5$, after applying the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant.

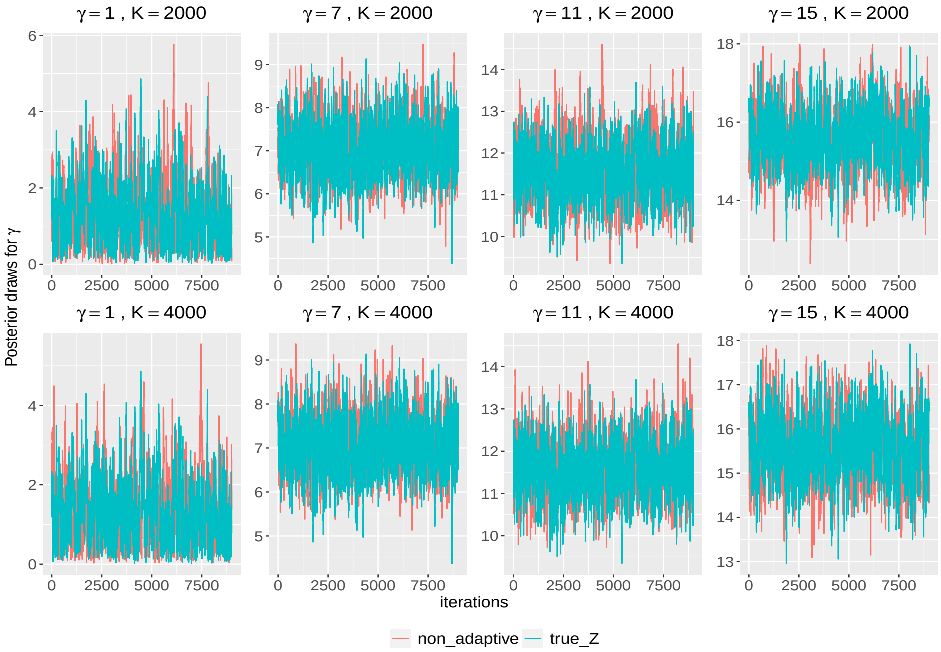

Figure 4.28: Traceplots for γ for simulation regimes presented in Table 4.3 with sample size $N = 20$, after applying the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant.

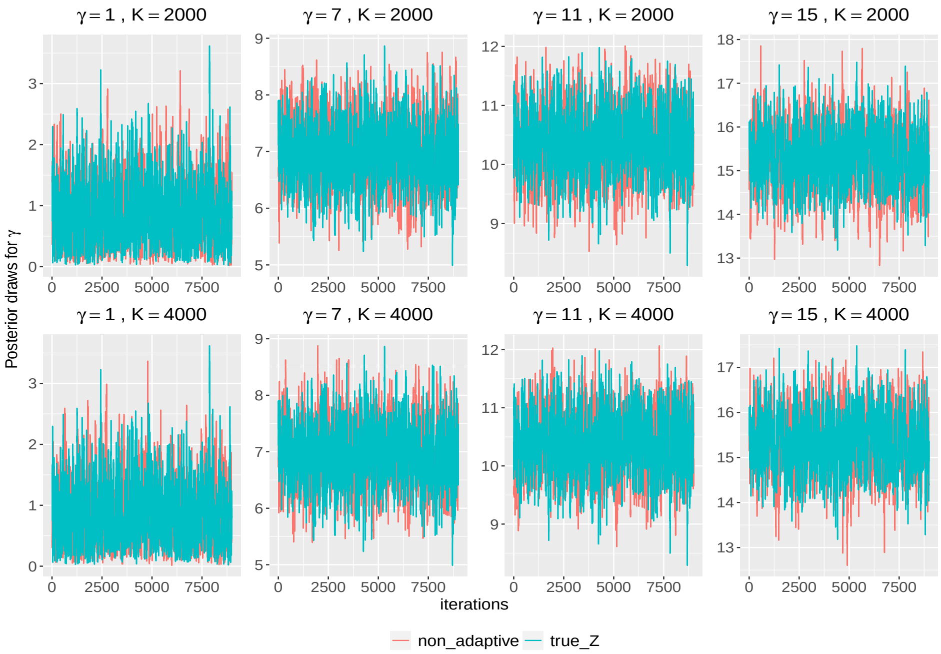

Figure 4.29: Traceplots for γ for simulation regimes presented in Table 4.3 with sample size $N = 50$, after applying the non-adaptive MCMC scheme and the MCMC scheme with the calculation of the true normalising constant.

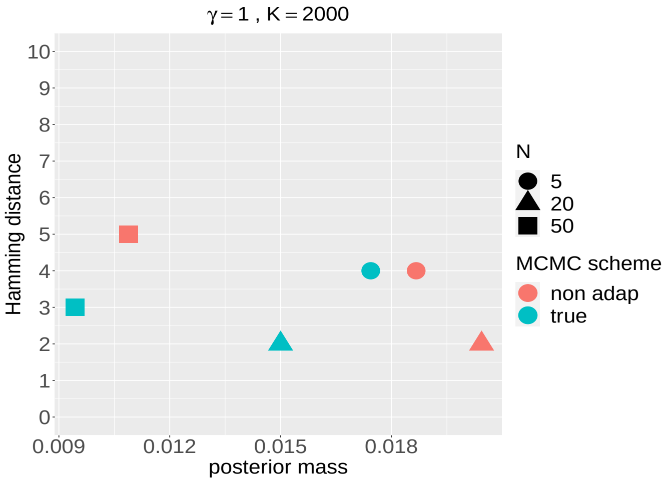

Figure 4.30: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 1$ and $K = 2000$, and various sample sizes N and MCMC schemes.

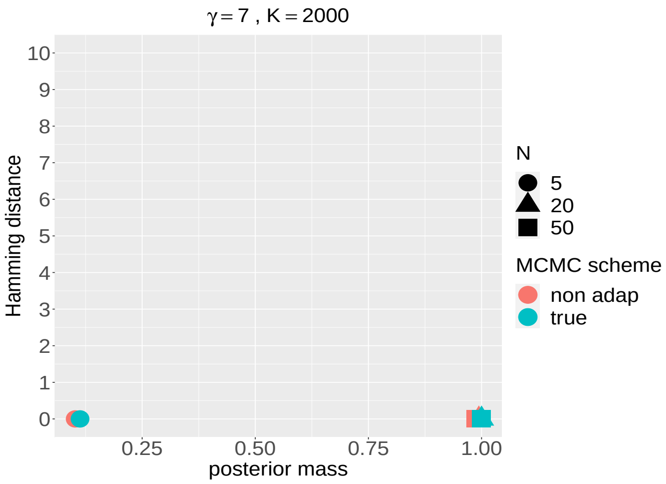

Figure 4.31: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 7$ and $K = 2000$, and various sample sizes N and MCMC schemes.

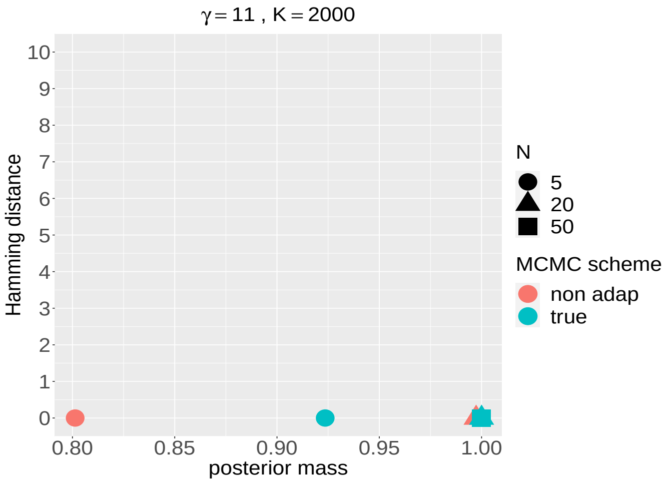

Figure 4.32: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 11$ and $K = 2000$, and various sample sizes N and MCMC schemes.

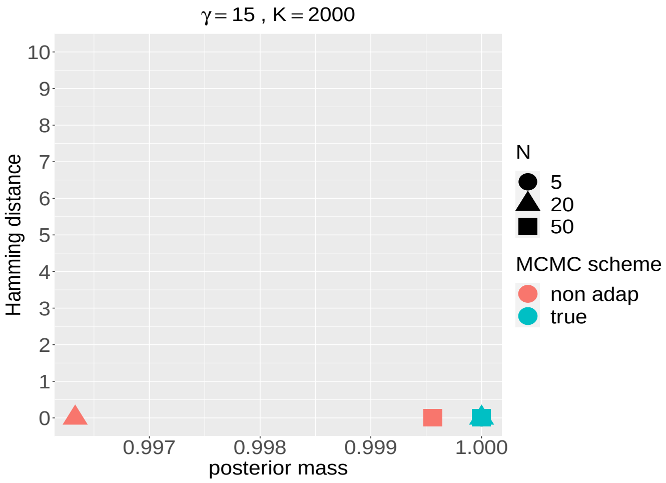

Figure 4.33: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 15$ and $K = 2000$, and various sample sizes N and MCMC schemes.

Investigation for various IS sample sizes and different IS densities

In Section 4.3.6, we discussed how the bias in estimating Z^N increases as a function of N, when we estimate Z under an IS scheme. As shown in section 4.3.6, we expect this bias to increase for increasing network population sizes N. However, in the previous simulation studies for 5-node networks, the effect of the bias was not obvious from the simulation results which can be attributed to the large IS sample sizes K obtained, as discussed in Vitelli et al. [2017]. In this simulation study, we explore the bias occurring with N considering a range of IS sample sizes K, and investigate how alternative formulations of our algorithm perform in dealing with the bias.

We first consider one of the simulation regimes introduced for the Jaccard distance in this section. Notably, we consider the population of $N = 50$ networks generated from the SNF model with $\gamma = 11$. Thence, we apply our MCMC algorithm with IS step, for various IS sample sizes ranging from 100 up to 900 samples, and different formulations of the IS step. With respect to the IS step formulations, we consider two alternative ways to sample from the CER model, namely the MCMC and the Monte Carlo sampling schemes, as introduced in Section 4.3.3. In addition, we specify a mixture of CER models as the IS density and lastly, we apply an error adjustment as obtained in section 4.3.6 for correcting the bias in our estimate.

For the single CER IS density, we tune its parameters similarly to the tuning introduced in the previous simulation studies in this section. Specifically, we obtain the posterior mode for $\mathcal{G}^m$ and posterior mean for α, after fitting the simulated network population to the CER model. For the mixture of CER IS density, we specify a common centroid being the posterior mode obtained for the single CER model, and determine $J = 3$ mixture components with probabilities $\boldsymbol{\beta} = (1/3, 1/3, 1/3)$ corresponding to three different sizes of $\tilde{\boldsymbol{\alpha}} = (0.2, 0.3, 0.4)$.

In Figure 4.34 we present the results for the posterior distribution of γ under the four different formulations of our algorithm. We run each algorithm for 10,000 iterations with a burn-in of 1,000. We now see the effect of the IS sample size K on the performance of our algorithm in inferring the model parameters. Notably, for K ranging from 100 to 600 networks our algorithm with IS density the CER model and the MCMC sampling scheme, consistently underestimates the dispersion parameter γ. A similar behaviour is identified for the algorithm with the error adjustment, indicating that this formulation does not correct the bias in our estimation. However, we observe that the formulation of a Monte Carlo sampling scheme for the CER model (red boxplots) provides more accurate inferential results for small IS sample sizes K. Similarly, the specification of a mixture of CER models for the IS density (blue boxplots) leads to more accurate inference of γ. This result indicates that both the IS density specified and the sampling scheme used to sample from the IS density, play a significant role in the performance of the MCMC.

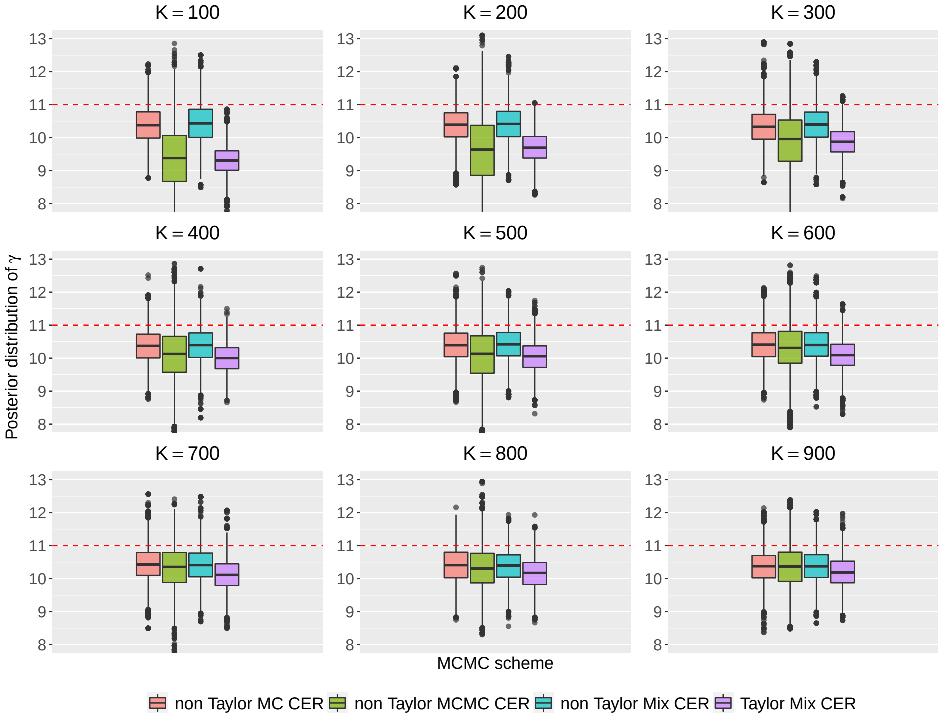

Figure 4.34: Posterior distribution of γ under the four different formulations of our algorithm (colours), for varying IS sample size of K.

In the next section we further probe the performance of our algorithm in inferring the parameters of the SNF model for larger network sizes, under both the Jaccard and the HS distance metric specification. For the latter, we also explore the performance of the MCMC with the xgboost algorithm formulated to perform prediction of the symmetric difference in the HS distance, as presented in section 4.3.7.

4.4.3 Performance of MCMC for moderate network sizes

We now introduce the simulation study performed to explore the performance of the MCMC with IS step for moderate network sizes. Specifically, we consider networks with $n = 20$ nodes, motivated by the network sizes of the ecological application presented in

Section 4.3.1. In contrast to the 5-node case, for the 20-node networks the computation of the true normalising constant is not feasible, as now the size of the space of 20-node graphs is $|\{\mathcal{G}_{|20|}\}| = 2^{190}$. Thence, in this simulation study it is not feasible to compare the results of our algorithm to the results of an MCMC which involves the exact calculation of the normalising constant. However, synthetic data experiments still allow a controlled environment to draw conclusions about the performance of our algorithm.

Analogously to the 5-node case, we set up the simulation regimes with respect to varying sizes of the dispersion γ and the population size N. We examine the performance of our MCMC for both the Jaccard and the HS distance metric, thus, the sizes for γ are determined after conducting EDA for each distance metric. The network population sizes considered are $N = \{5, 20, 50\}$, which are common for both the simulation study for the Jaccard and the HS distance.

To determine the sizes of the dispersion γ, we conduct EDA similarly to the EDA performed for the 5-node networks in section 4.4.2. In this respect, we simulate network populations from the SNF model with centroid $\mathcal{G}^0$ presented in Figure 4.35 and various sizes of γ_0, and obtain the distances between the simulated networks and the centroid with respect to the distance metric considered. Figures 4.36 and 4.37 show the distribution of the distance obtained for each size of γ_0, for the 20-node networks case and for the Jaccard and the HS distance metric, respectively.

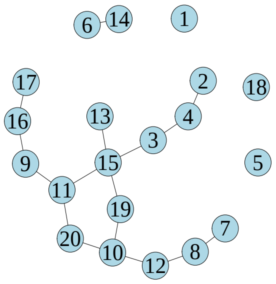

Figure 4.35: 20-node centroid $\mathcal{G}^0$.

As previously discussed, we expect that different distance metric specifications will reveal different relationships between the expected distance $\mathbb{E}[d(\mathcal{G}, \mathcal{G}^0)]$ and the dispersion γ_0. To begin with, for the Jaccard distance metric, we observe a significant drop in the average distance of the simulated networks from the centroid when gamma ranges from 80 to 100, while for ranges of gamma outside of this interval, the average distance

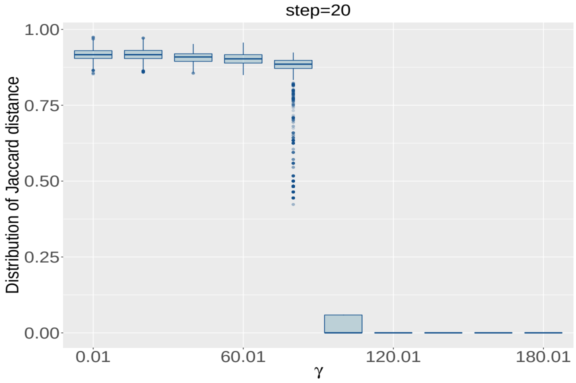

Figure 4.36: Distribution of the Jaccard distance between networks generated from the SNF($\mathcal{G}^0$,γ_0) for various sizes of γ_0, for 20-node networks.

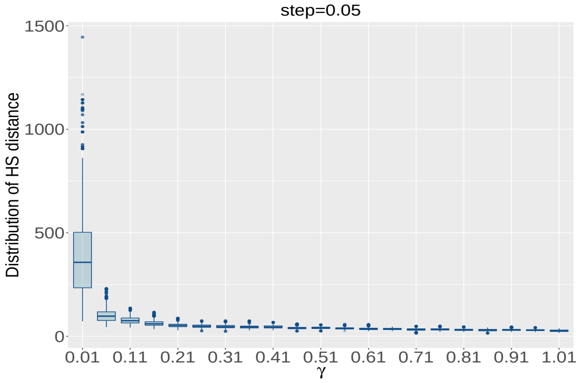

Figure 4.37: Distribution of the HS distance between networks generated from the SNF($\mathcal{G}^0$,γ_0) for various sizes of γ_0, for 20-node networks.

scales gradually with gamma. On the other hand, we observe a smoother behaviour of the distribution of the distance for the HS distance metric for increasing γ_0.

In the following sections we set up the simulation regimes for which we generate populations of 20-node networks under the SNF model, and implement the MCMC with IS step to explore whether our algorithm accurately infers the true sizes of the parameters.

Simulation study for Jaccard distance

We now introduce the simulation regimes for the Jaccard distance metric, utilising the information captured by the EDA. Various sizes of γ imply various levels of variability between the simulated network population and the centroid, as seen in Figure 4.36. In our simulation study, we consider a range of values for γ such that network populations with different levels of variability are simulated. Specifically, we consider $\gamma = \{30, 90, 100, 120\}$ and specify two IS sample size $K = \{2000, 6000\}$. The simulation regimes for the 20-node case and the Jaccard distance, are summarised in Table 4.4.

n	N	K	γ
20	5	2000	30
20	20	2000	30
20	50	2000	30
20	5	6000	30
20	20	6000	30
20	50	6000	30
20	5	2000	90
20	20	2000	90
20	50	2000	90
20	5	6000	90
20	20	6000	90
20	50	6000	90
20	5	2000	100
20	20	2000	100
20	50	2000	100
20	5	6000	100
20	20	6000	100
20	50	6000	100
20	5	2000	120
20	20	2000	120
20	50	2000	120
20	5	6000	120
20	20	6000	120
20	50	6000	120

Table 4.4: Simulation regimes for 20-node networks for the Jaccard distance metric.

For each regime, we simulate a network population and apply solely a non-adaptive MCMC with IS step, as the implementation of an MCMC involving the calculation of the true normalising constant is not feasible for 20-node networks. We tune the prior for γ and $\mathcal{G}^m$ similarly to the tuning conducted for the simulation study in section 4.4.2, and run our MCMC algorithm for 10,000 iterations with a burn-in of 1,000 iterations.

The boxplots in Figures 4.38 and 4.39 show the posterior distribution for γ, for each simulation regime. We observe that our algorithm performs well in inferring γ for $N = 5$, as the posterior mean of the distribution is close to the true value of γ. However, we notice that our algorithm consistently underestimates γ for some simulation regimes, and specifically for increasing population sizes N and dispersion sizes γ equal to 30, 90 and 100. As discussed in section 4.3.6, the poor performance of our algorithm under these regimes can be attributed to the bias increasing with N. In Figures 4.40 and 4.41, we also present the traceplots for γ, with colours corresponding to the two different sizes of K.

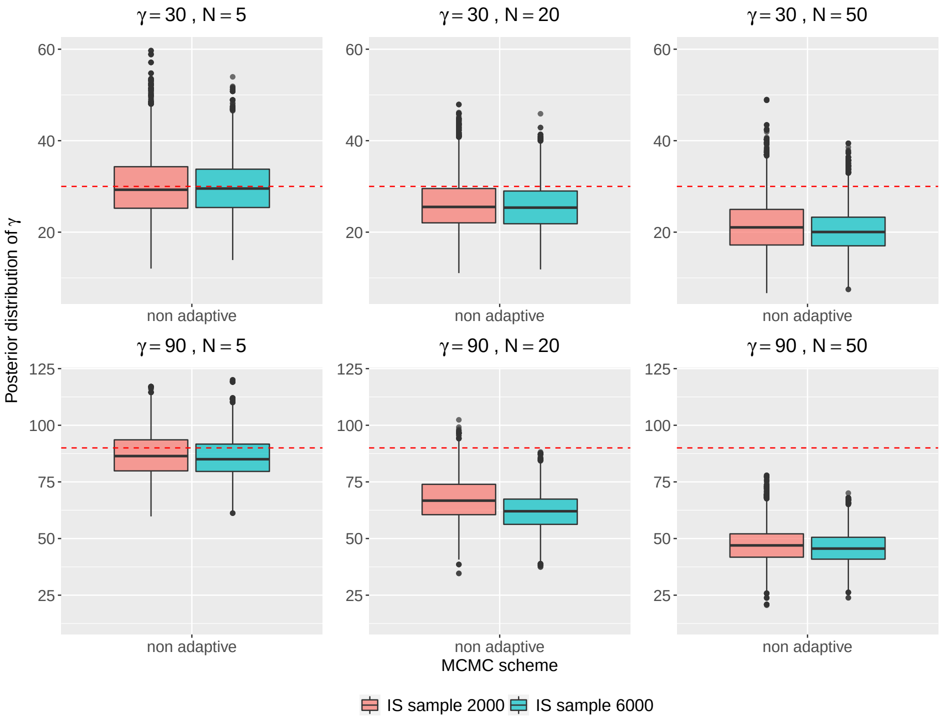

Figure 4.38: Posterior distribution for γ for simulation regimes presented in Table 4.4 after applying the non-adaptive MCMC scheme. The red dashed line indicates the true value of γ.

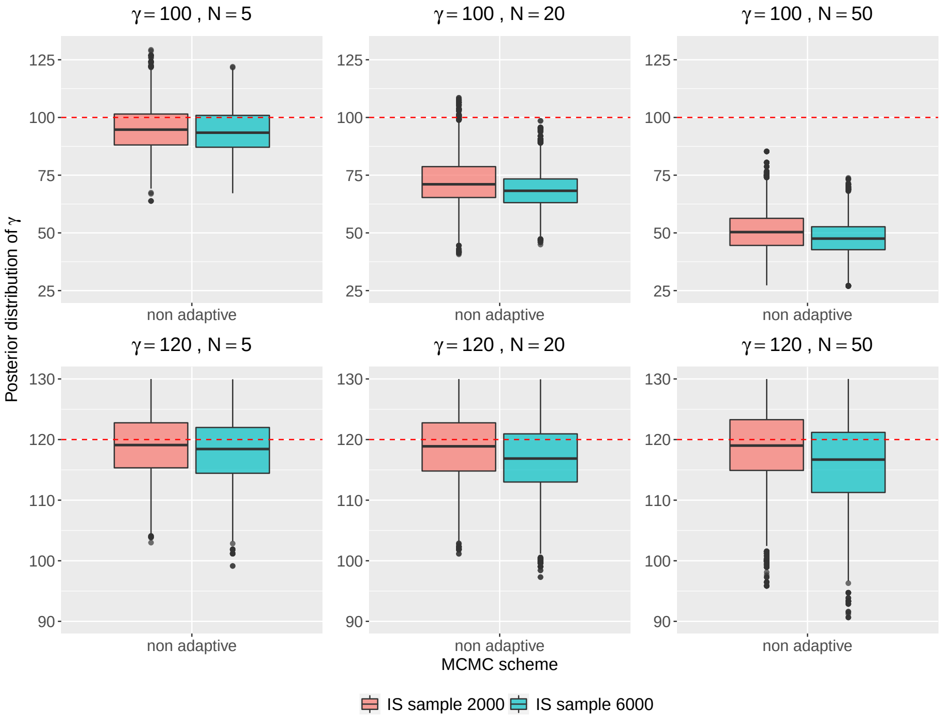

Figure 4.39: Posterior distribution for γ for simulation regimes presented in Table 4.4 after applying the non-adaptive MCMC scheme. The red dashed line indicates the true value of γ.

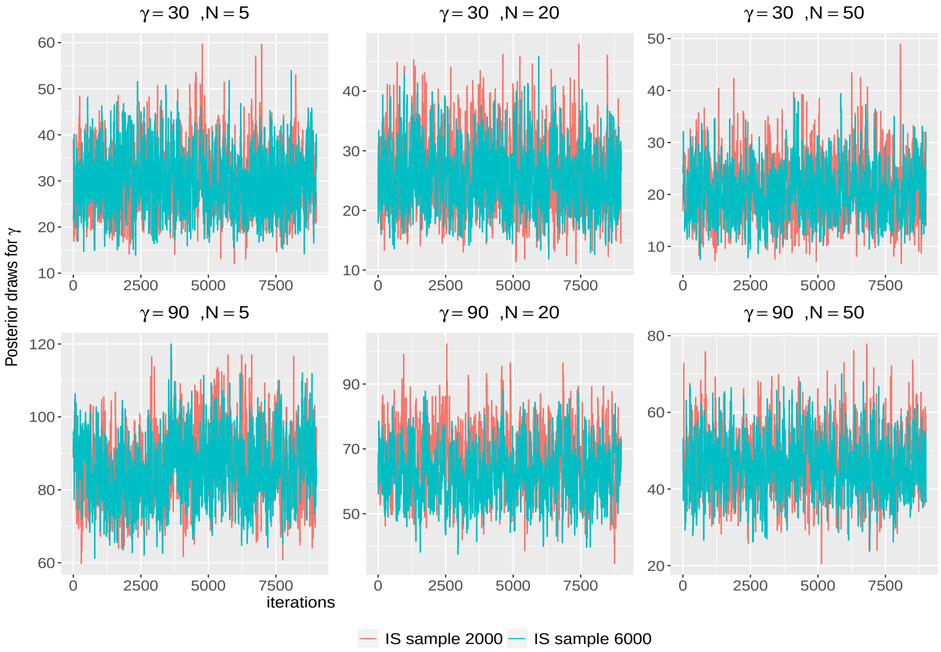

Figure 4.40: Traceplots for γ for simulation regimes presented in Table 4.4 after applying the non-adaptive MCMC scheme.

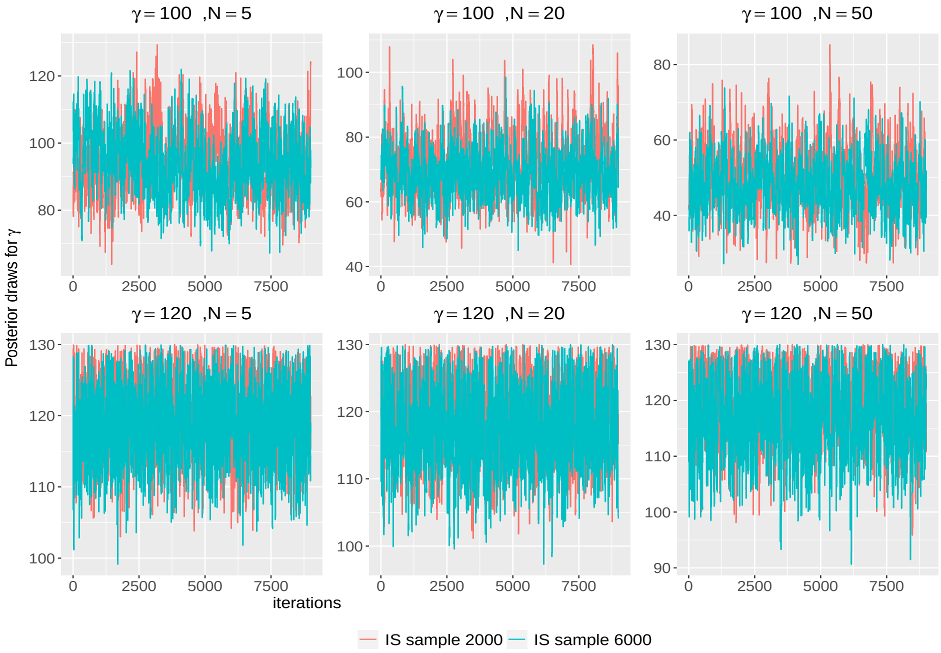

Figure 4.41: Traceplots for γ for simulation regimes presented in Table 4.4 after applying the non-adaptive scheme.

In Section 4.4.2, we investigated the impact of the IS density specification on the bias occurring with large N through simulation studies. We observed that a mixture of CER models IS density outperformed a single CER model IS density in inferring the true size of γ. Utilising this result, we investigate whether the mixture model IS density specification in our MCMC scheme achieves more accurate results for the 20-node network case, and further increase the IS sample size K to 10,000 networks.

Figures 4.42 and 4.43 show the posterior distribution of γ for the non-adaptive MCMC algorithm, with IS density specification a mixture of CER models. We notice that the inferential results obtained are similar to those obtained for the single CER IS density. Thus, the effect of the IS density on the bias occurring with N is not evident for the 20-node networks case. However, we note that the posterior distribution for the regime of $\gamma = 120$ is concentrated on the upper bound of γ equal to 130, where $\mathbb{E}[d(\mathcal{G}, \mathcal{G}^0)] = 0$, as seen from the EDA. This can be explained by the similar behaviour of the distribution of the distance for $\gamma \geq 120$, as seen in figure 4.36.

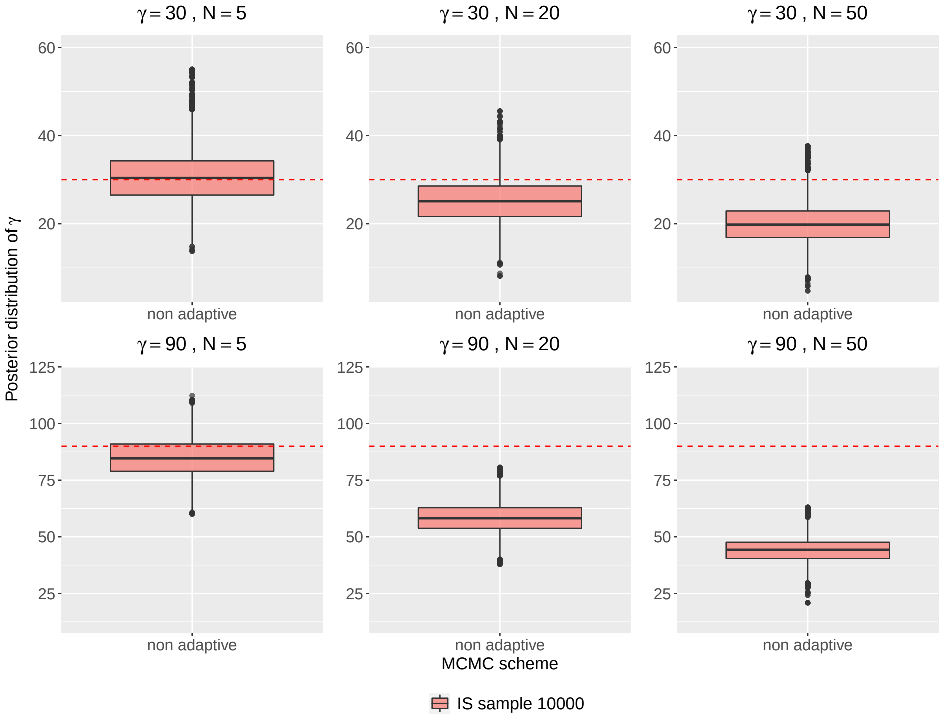

Figure 4.42: Posterior distribution for γ for simulation regimes presented in Table 4.4 after applying the non-adaptive MCMC scheme, with IS density the mixture of CER models and IS sample size $K = 10,000$. The red dashed line indicates the true value of γ.

In addition, we obtain summaries for the posterior draws for the centroid $\mathcal{G}^m$, for both IS density specifications, and present the results in Figures 4.44, 4.45, 4.46 and 4.47. We observe that the centroids inferred are dissimilar to the true centroid with respect to the Hamming distance and they also gather small posterior masses, under both the mixture model and the single CER model IS density. However, for the simulation regime with the less variation in the simulated network population, we notice that the algorithm with the mixture model IS density specification accurately infers the true centroid, and the posterior mode concentrates a high mass.

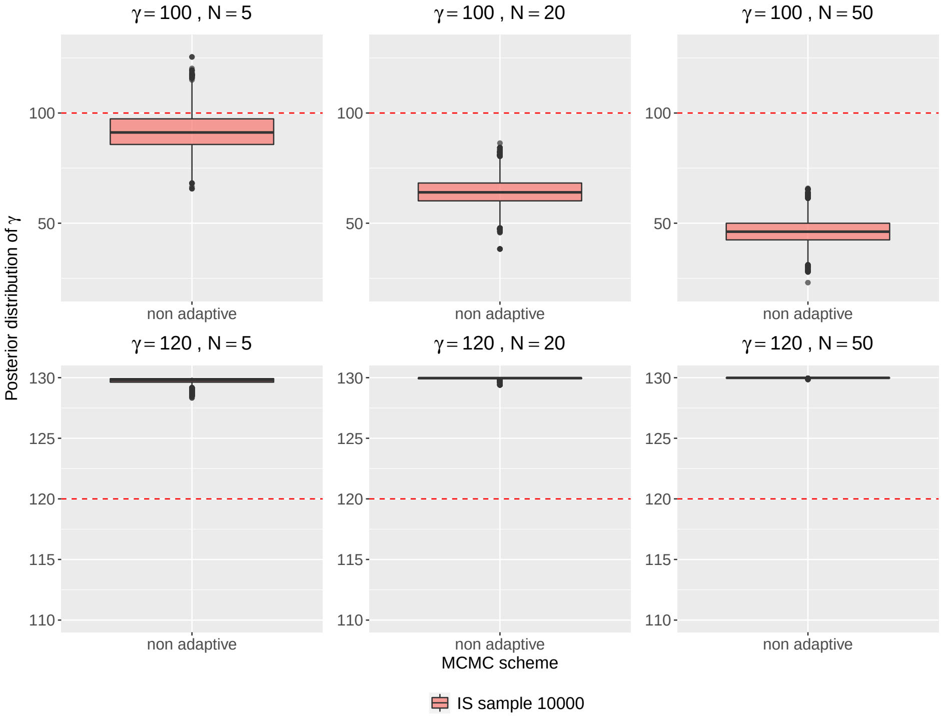

Figure 4.43: Posterior distribution for γ for simulation regimes presented in Table 4.4 after applying the non-adaptive MCMC scheme, with IS density the mixture of CER models and IS sample size $K = 10,000$. The red dashed line indicates the true value of γ.

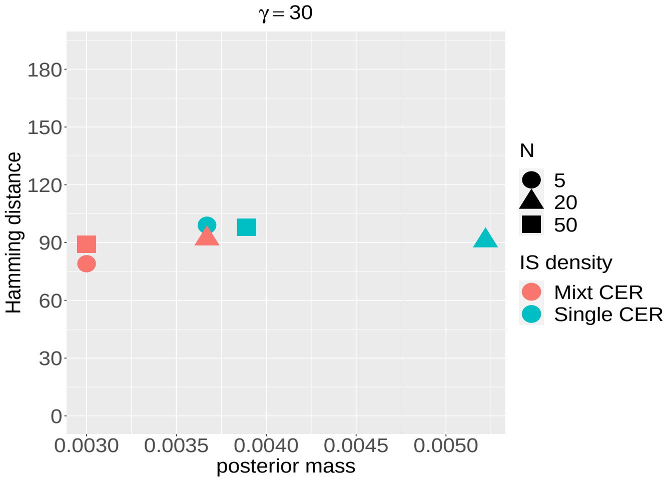

Figure 4.44: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 30$ and $K = 6000$, and various sample sizes N and MCMC schemes.

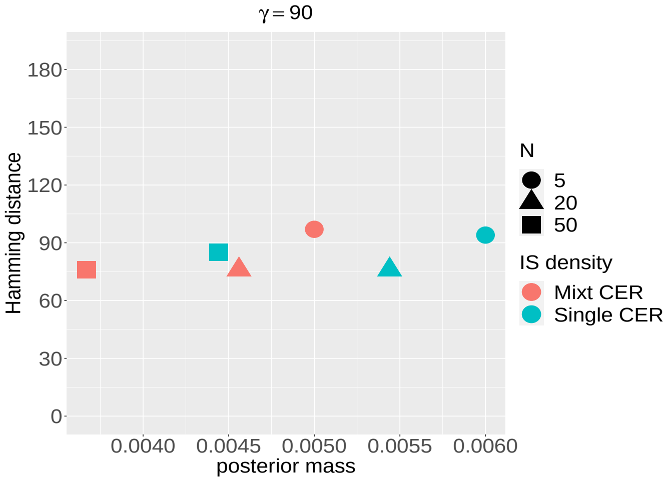

Figure 4.45: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 90$ and $K = 6000$, and various sample sizes N and MCMC schemes.

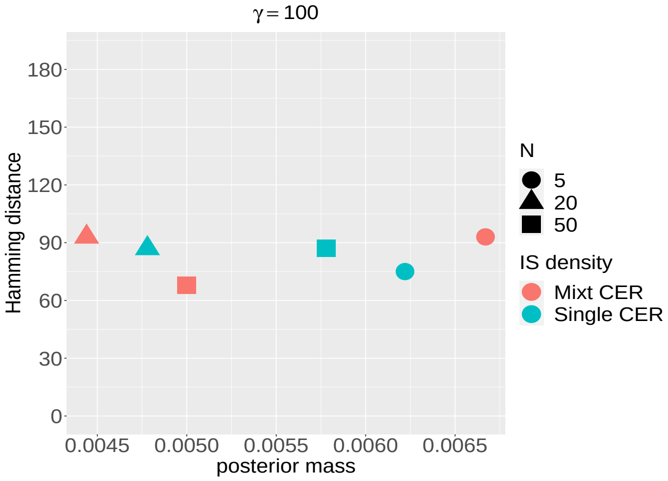

Figure 4.46: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 100$ and $K = 6000$, and various sample sizes N and MCMC schemes.

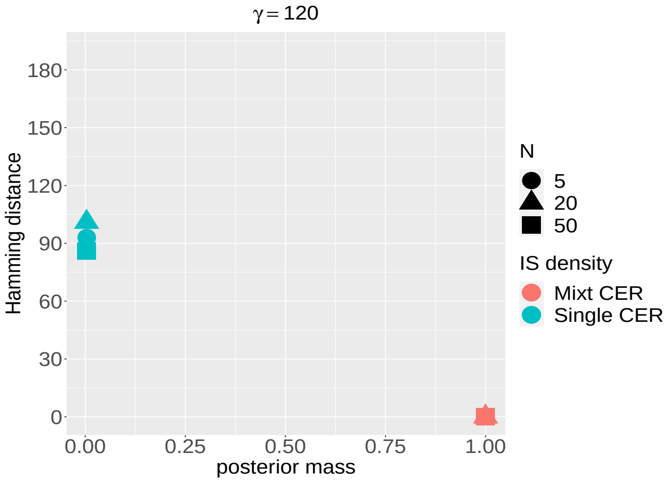

Figure 4.47: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for $\gamma = 120$ and $K = 6000$, and various sample sizes N and MCMC schemes.

In Appendix B.5.1, we further present the simulation results of our MCMC for a greater size of $K = 18,000$ networks, for one of the challenging regimes where $\gamma = 90$ and $N = 20$. However, we do not notice any change in the inferential results.

We note here that despite the limitation of our algorithm to accurately infer the parameters of the SNF model for large populations sizes N, we are still able to implement our algorithm on multiple network data sets that involve only few network observations. This is the case with the ecological application that has motivated our study, which involves date sets with less than 5 network observations each.

In the next section we explore the performance of our MCMC algorithm for the HS distance metric specification, for network sizes and population sizes similar to the sizes of the real data example.

Simulation study for HS distance

In the previous Sections, we examined the performance of the IS step for approximating the normalising constant within the MCMC algorithm, for a range of population sizes $N = \{5, 20, 50\}$ and two network sizes $n = \{5, 20\}$. Motivated by the ecological application, in this section we aim to assess the performance of the MCMC with IS step for the SNF model whith the HS distance metric, for network and sample sizes similar to the sizes of the fish multiple network data. Notably, the network populations sizes from the ecological application are $N = \{3, 4\}$, and the number of networks' nodes are $n = \{18, 19, 22\}$. Motivated by the application, in this simulation study we consider populations of networks with $N = 5$ networks, and $n = 20$ nodes to assess the performance of our algorithm.

A key challenge associated with the HS distance metric is the cycle detection, which is a computationally intensive task for 20-node networks. Hence, we consider only two simulation regimes involving two alternative sizes of $\gamma = \{0.06, 0.6\}$. We then simulate two network populations, corresponding to two different levels of variability from the centroid, as seen from the EDA boxplots in Figure 4.37. Similarly to the 20-node simulation study for the Jaccard distance, the centroid $\mathcal{G}^m$ specified in this simulation study is as seen in figure 4.35.

In Section 4.3.7, we introduced an alternative way to obtain the HS distance between networks avoiding the cycle detection step for every new IS sample drawn, through the implementation of an xgboost algorithm. In this simulation study, we explore the performance of the xgboost within our MCMC algorithm and compare its results to the results obtained from the MCMC with the exact calculation of the HS distance. Moreover, we demonstrate the computational time benefit from applying the xgboost algorithm, instead of calculating the exact HS distance in each iteration.

The IS density specified for this simulation study is a single CER model and the tuning performed is similar to that presented in Section 4.4.2. We consider two alternative schemes for sampling from the CER model, namely the MCMC scheme and the Monte Carlo scheme, and obtain IS samples of $K = 6,000$ networks in each iteration of our main algorithm. We run our MCMC for 5,000 iterations for each simulation regime considered.

Figure 4.48 shows the posterior distributions for γ for the network population generated from the SNF model with $\gamma = 0.06$. We observe that the posterior draws for γ are similarly distributed for both the case of the MCMC with the exact calculation of the HS distance and the case of the MCMC with the xgboost algorithm. For the IS sampling scheme using MCMC and the xgboost predictions of the HS distance, we notice that the posterior distribution covers a wider range of values for γ, and the 50% confidence interval encloses the true value of $\gamma = 0.06$. This can also be noticed from the traceplots obtained in Figures 4.49 and 4.50, that show the posterior draws for γ for the MCMC with the exact and the MCMC with the predicted HS distance, and the two alternative IS sampling schemes. Notably, the traceplots under the MC IS sampling scheme explore the same region of values for γ under both the MCMC with the exact calculation of the HS distance and the MCMC with the xgboost implementation. In addition, the chains mix well in both cases.

For the second simulation regime involving $\gamma = 0.6$, the inferential results for γ are more perturbed compared to the the first simulation regime of $\gamma = 0.06$. Figure 4.51 illustrates the posterior distribution of γ under each MCMC implementation. In this case, the posterior distribution under the exact calculation of the HS distance, entails a range of values for γ closer to its true size. However, the traceplots presented in Figures 4.52 and 4.53, show that the chains for the exact and the xgb scheme perform similarly, having mixing and stationarity issues. One way to explain the chains' behaviour under this simulation regime, is through the EDA boxplots obtained in figure 4.37. Notably,

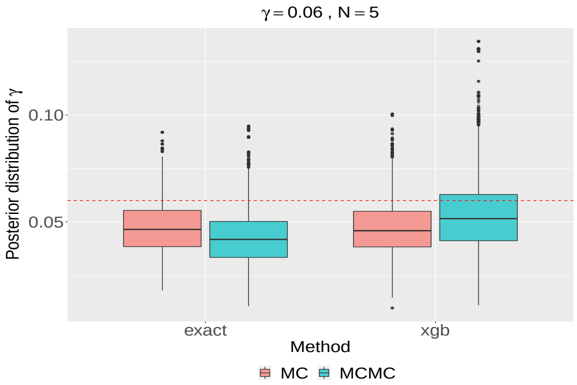

Figure 4.48: Posterior distribution of γ for simulation regime $\gamma = 0.06$ resulting from the MCMC scheme with the exact calculation of the HS distance versus the MCMC scheme with the implementation of the xgboost algorithm (x axis), and two alternative schemes for sampling from the IS density (colours). The red dashed line indicates the true value of γ.

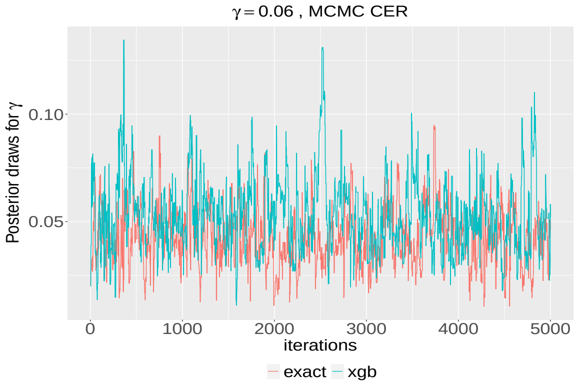

Figure 4.49: Traceplots for γ for simulation regime $\gamma = 0.06$ resulting from the MCMC scheme with the exact calculation of the HS distance versus the MCMC scheme with the implementation of the xgboost algorithm (colours) and sampling from the IS density using an MCMC scheme.

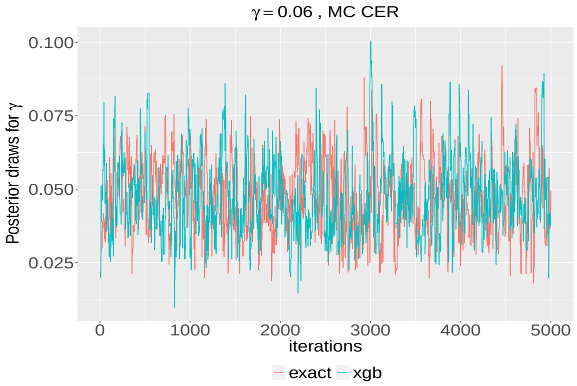

Figure 4.50: Traceplots for γ for simulation regime $\gamma = 0.06$ resulting from the MCMC scheme with the exact calculation of the HS distance versus the MCMC scheme with the implementation of the xgboost algorithm (colours) and sampling from the IS density using Monte Carlo scheme.

for the range of γ values explored by the MCMC chains, specifically for $0.3 < \gamma < 0.8$, the distribution of the distance does not change considerably. This may affect the convergence of the MCMC chains, due to the non distinguishable regimes of γ being explored.

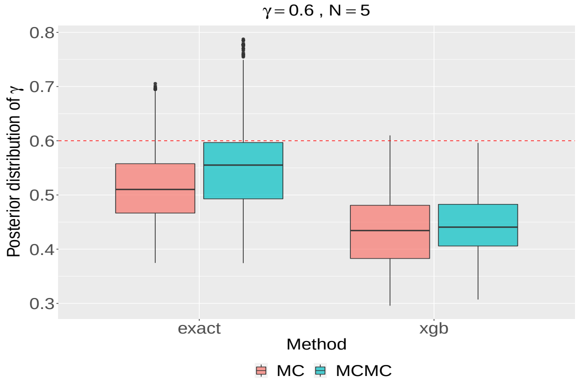

Figure 4.51: Posterior distribution of γ for simulation regime $\gamma = 0.6$ resulting from the MCMC scheme with the exact calculation of the HS distance versus the MCMC scheme with the implementation of the xgboost algorithm (x axis), and two alternative schemes for sampling from the IS density (colours). The red dashed line indicates the true value of γ.

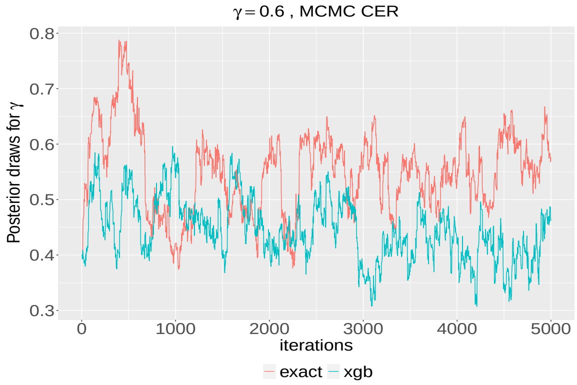

Figure 4.52: Traceplots for γ for simulation regime $\gamma = 0.6$ resulting from the MCMC scheme with the exact calculation of the HS distance versus the MCMC scheme with the implementation of the xgboost algorithm (colours) and sampling from the IS density using an MCMC scheme.

A common result from both the MCMC with the xgboost algorithm and the MCMC with the exact HS calculation, is that the centroids inferred under each scheme are different from the true centroid by approximately 40 edges with respect to the Hamming distance, and they concentrate low masses for both simulation regimes of $\gamma = 0.06$ and $\gamma = 0.6$, as presented in Figures 4.54 and 4.55. In addition, the IS sampling scheme specified does not appear to have a significant effect on the centroid inference.

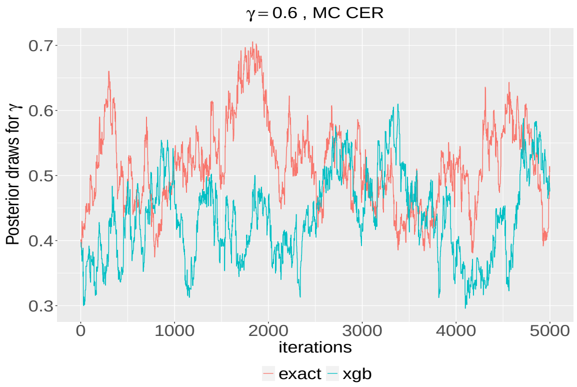

Figure 4.53: Traceplots for γ for simulation regime $\gamma = 0.6$ resulting from the MCMC scheme with the exact calculation of the HS distance versus the MCMC scheme with the implementation of the xgboost algorithm (colours) and sampling from the IS density using a Monte Carlo scheme.

Specifically, we notice that for the MCMC IS sampling scheme the posterior modes for $\mathcal{G}^m$ concentrate only slightly higher posterior masses than the MC IS sampling scheme.

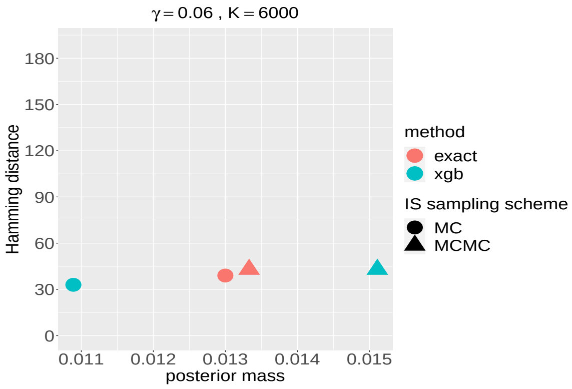

Figure 4.54: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for simulation regime $\gamma = 0.06$.

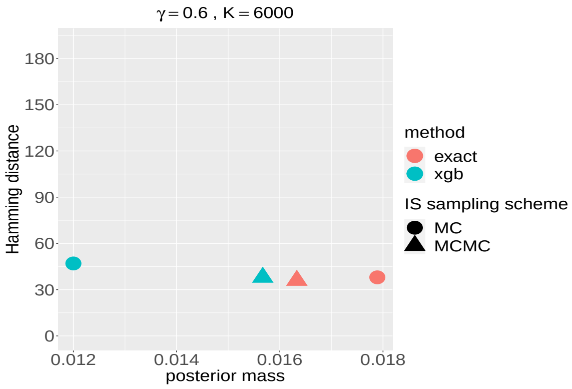

Figure 4.55: Hamming distance between the posterior mode centroid and the true centroid (y axis) versus the posterior mass of the posterior mode centroid (x axis) for simulation regime $\gamma = 0.6$.

As seen in the 20-node simulation study for the Jaccard distance, the specification of the mixture of CER models as the IS density improves the centroid inference for the simulated network populations with the less variability from the centroid. However, for the HS distance metric the use of the mixture of CER models as the IS density can be computationally challenging depending on the sizes of the dispersion $\tilde{\alpha}$ specified. Notably, for greater sizes of $\tilde{\alpha}$ there is greater variability between the simulated networks sampled and the centroid, thus dense networks are also sampled for which the task of cycle detection can be computationally infeasible. We note here that even with the xgboost implementation, we still need to calculate the cycles of the training IS sample.

However, we can explore the performance of the mixture of CER models IS density, for small sizes of the dispersion, and specifically for $J = 2$ mixture components with probabilities $\beta = (1/2, 1/2)$ corresponding to two dispersion parameters $\tilde{\alpha} = (0.02, 0.07)$, resulting in a sample with less variability from the centroid. The traceplots of the posterior draws for γ after running the MCMC with xgboost algorithm are presented in Appendix B.5.2. We note here that the inferential results for the dispersion γ and the centroid $\mathcal{G}^m$, are similar to the results obtained from the MCMC with IS density specification the single CER model, presented above.

Lastly, in Figure 4.56 we demonstrate the computational benefit of formulating an xgboost algorithm to predict the HS distance between networks, as opposed to its exact calculation. We observe that under both the MC and the MCMC sampling scheme for the IS density, the MCMC algorithm with the xgboost formulation is remarkably faster than the MCMC with the exact calculation of the HS distance. Specifically, for the MC IS sampling scheme, 5,000 iterations of the MCMC with the xgboost formulation complete in approximately half a day, while 5,000 iterations of the MCMC with the exact HS distance calculation need 16 days to be completed. Thus, the xgboost formulation is 26 times faster than the exact calculation of the HS distance. Respectively, under the MCMC IS sampling scheme, the xgboost is 16 times faster than the exact calculation of HS within the MCMC.

It is further worth noting that the MC IS sampling scheme, is considerably more computationally intensive compared to the MCMC IS sampling scheme, particularly for the MCMC algorithm with the exact calculation of the HS distance. This is an anticipated result considering that the MC IS sampling scheme resembles to sampling uniformly from the space of graphs as discussed in Section 4.3.3, thus a wider range of 20-node networks is sampled resulting in denser networks that can potentially enclose many cycles.

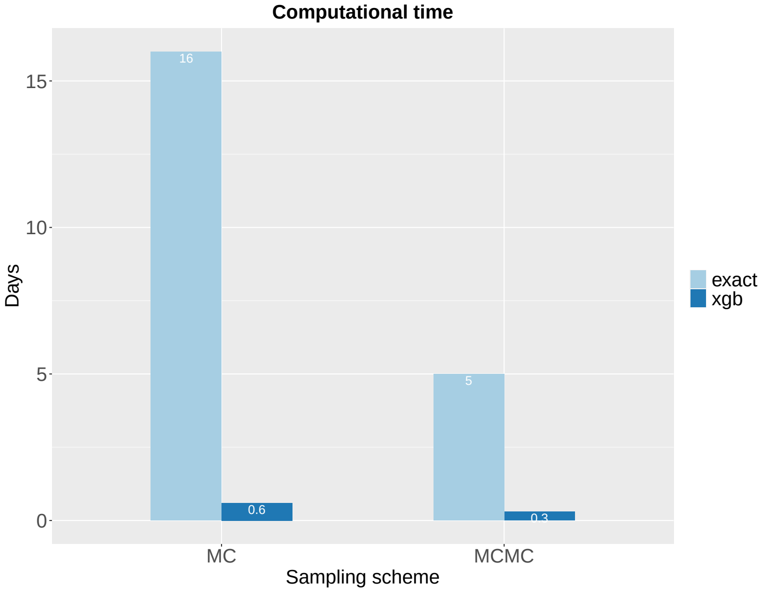

Figure 4.56: Computation time in days (y axis) required for 5,000 iterations of the MCMC with the exact calculation of the HS distance (light blue) versus the MCMC with the xgboost implementation (dark blue), under two alternative IS sampling schemes (x axis).

4.5 Real data examples

We now return to the ecology application that motivated our study. In this section we analyse three multiple network data sets, corresponding to data collected at three different regions in the Indo-Pacific ocean. In each multiple network data set, each network observation represents the aggressive interactions between species of fish at a specific reef of the corresponding region. Specifically, in our study we consider the Bali region, the Christmas Island and the Aceh region. In table 4.5, we present details about the network sizes n and the population sizes N for each region, and in table 4.6 we present all the fish species recorded at the three regions, along with their node labels. Thence, each region involves a subset of the total 29 species of fish (nodes) presented in table 4.6.

The goal of our analysis is to fit the SNF model to obtain posterior draws for the centroid network $\mathcal{G}^m$ and the dispersion parameter γ, assuming the HS distance metric

Region	**N**	**n**
Bali	3	22
Christmas Island	3	18
Aceh	4	19

Table 4.5: Network sizes n and sample sizes N for three different regions in the Indo-Pacific ocean.

Fish Species	**Label**	**Fish Species**	**Label**
adiergastos	1	melannotus	16
andamanensis	2	meyeri	17
auriga	3	ornatissimus	18
baronessa	4	punctatofasciatus	19
citrinellus	5	rafflesii	20
collare	6	semeion	21
decussatus	7	speculum	22
ephippium	8	triangulum	23
falcula	9	trifascialis	24
guttatissimus	10	trifasciatus	25
interruptus	11	ulietensis	26
kleinii	12	unimaculatus	27
lineolatus	13	vagabundus	28
lunula	14	xanthocephalus	29
lunulatus	15		

Table 4.6: Node labels assigned to species of fish observed at the Bali, the Christmas Island and the Aceh region.

that captures information about the networks' cycles, as cycles indicate a form of competition among species of fish. We note here that originally the edges of the networks are directed, denoting the direction of the aggressive encounter, and weighted, denoting the probability of the aggressive encounter. However, in this analysis we consider the undirected and unweighted version of the networks, as otherwise a different formulation of our MCMC algorithm would be required.

In the following subsections, we present summaries of the results obtained after applying the SNF model with HS distance metric specification on each of the three multiple network data sets.

4.5.1 Bali region

We first fit the SNF model to the multiple network data set corresponding to the Bali region. The data were collected at three different reefs of the Bali region, resulting in a multiple network data set with $N = 3$ network observations, that share the same set of $n = 22$ nodes. In Figure 4.57 we illustrate the network observations collected at the region of Bali.

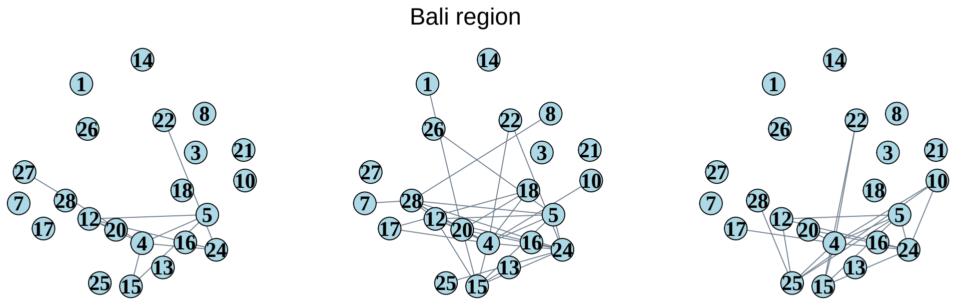

Figure 4.57: Networks representing interactions of fish at three different reefs of the Bali region.

We centre the prior for the centroid network at the network observation that minimises the HS distance from the rest of the network observations, as proposed by Lunagómez et al. [2021]. In our case, the network that minimises the HS distance from the other two networks is the left network in figure 4.57. We also specify a Gamma prior distribution for the dispersion parameter, and centre it with respect to the information from the EDA boxplots obtained for the HS distance in Figure 4.37 of section 4.4.3. Specifically, we obtain a centroid estimate for the data using majority vote, and calculate the HS distance between the centroid estimate and the data to identify the value of γ that the distance corresponds to, using the boxplots in figure 4.37. We then centre the prior of γ to that value identified. Lastly, we centre the single CER model IS density at the posterior mode obtained after fitting the data to the CER model, and specify a small size of the dispersion parameter $\tilde{\alpha}$ for the CER IS density, in order to be computationally feasible to run our MCMC algorithm. We further implement an MCMC scheme to sample from the CER model, and obtain an IS sample size of $K = 6,000$ networks, after a burn-in of 1,000 networks.

In Figures 4.58, 4.59 and 4.60, we present the results after running the MCMC with xgboost algorithm for 5,000 iterations. Specifically, in figure 4.58 we present the traceplot of the posterior draws for γ, in figure 4.59 we present the histogram of the posterior distribution for γ, while in figure 4.60 we present the 5 most commonly drawn networks from the posterior of the centroid. We further highlight in pink colour the edges of the centroid posterior draws that are present in the network data as well, to demonstrate which inferred edges are also observed in the data.

We note that our algorithm explores a region of small values for γ indicating a

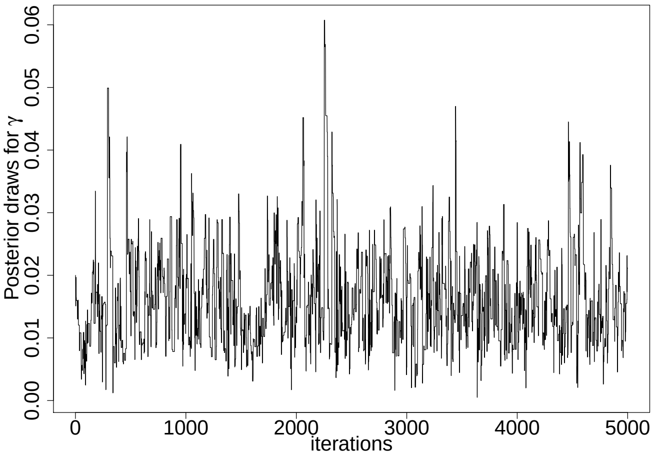

Figure 4.58: Traceplot for γ for 5,000 iterations of the MCMC for data recorded at the Bali region, using the xgboost algorithm to predict the HS distance in each iteration.

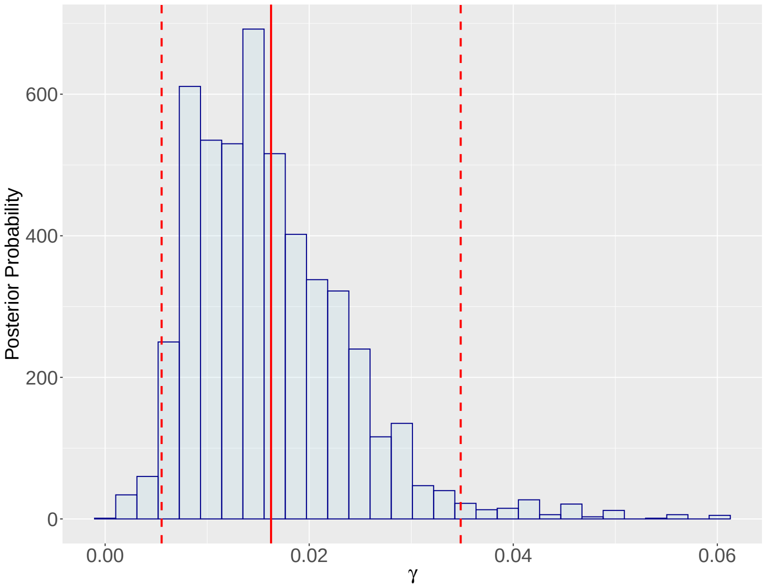

Figure 4.59: Histogram of posterior distribution of γ for data recorded at the Bali region, with red solid line indicating the mean 0.016, and red dashed lines indicating the 95% credible interval (0.01,0.03).

high dispersion of the data from the posterior centroids drawn. Notably, the most commonly drawn centroid networks are concentrating only small posterior masses of approximately 0.02, and most of their edges are not observed in the network data as well. This result suggests that inferring a centroid network for this multiple network data set is challenging, due to the variability of the network data with respect to the HS distance metric. Notably, the network observations corresponding to the Bali region seen in Figure 4.57, enclose 6, 1219 and 154 cycles respectively from left to right, resulting in

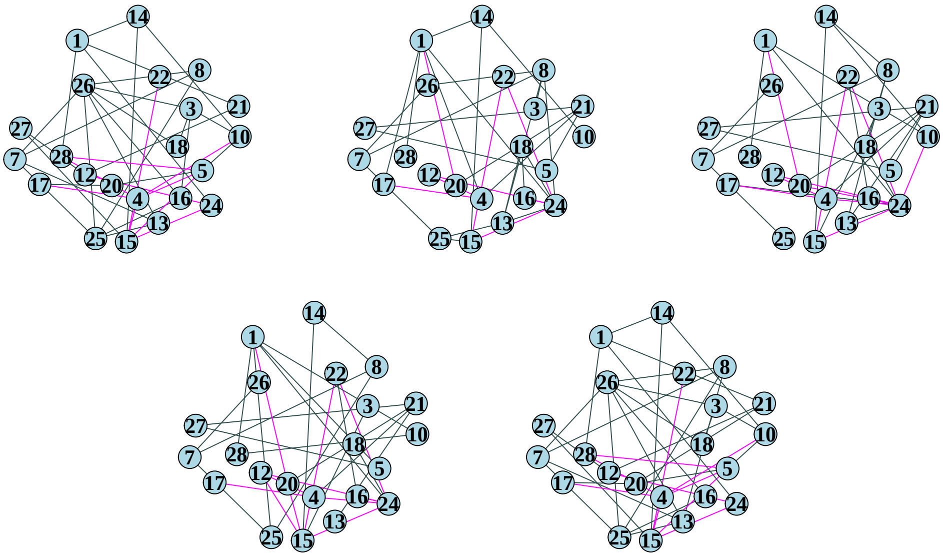

Figure 4.60: Top 5 most commonly drawn centroids with posterior masses from top left to bottom right 0.0262, 0.0234, 0.0232, 0.018 and 0.0176 respectively. Pink edges indicate edges that are also present in the data recorded at the Bali region.

having many not in common cycles between them.

As the posterior draws for the centroid network concentrate only small masses, a single centroid is not encoding all the information for the data. To summarise the information obtained from the posterior draws we investigate whether there are cycles repeatedly found in the posterior centroids drawn, as well as which of the most common cycles are found in the network data as well. In this respect, we identify the 10 most common cycles found in the 5,000 posterior centroids drawn, and count the number of posterior centroid networks that contain each of these cycles. Lastly, we detect which of the cycles identified are also observed in the network data. The results from this computations are gathered in table 4.7.

From table 4.7, we observe that each of the most common cycles are identified in a large proportion of the posterior centroids drawn. Thus, we can deduce that those cycles can be informative about the data. We further note that the only cycle of the most common cycles that is present in the network data as well is the the cycle between the fish species baronessa (node 4), lunulatus (node 15) and citrinellus (node 5), which is found in the 0.33 proportion of centroid networks drawn.

most common cycles	proportion of times identified	observed/inferred
7-26-8-7	0.66	inferred
15-24-22-15	0.39	inferred
13-26-16-25-13	0.38	inferred
5-27-21-5	0.37	inferred
1-28-5-4-15-14-1	0.35	inferred
7-25-13-26-7	0.34	inferred
7-26-13-25-7	0.34	inferred
4-15-5-4	0.33	observed
13-26-25-13	0.33	inferred
7-25-13-8-7	0.3	inferred

Table 4.7: 10 most common cycles detected in posterior draws for centroid network, proportion of times each cycle detected in posterior draws and presence (observed) or absence (inferred) of cycle in the data recorded at the Bali region.

4.5.2 Christmas island

We now analyse the multiple network data set corresponding to the Christmas island. Aggressive interactions of $n = 18$ species of fish were recorded at three different reefs of the Christmas island resulting in $N = 3$ network observations, as shown in figure 4.61. We note that each network observation involves 12, 1834 and 0 cycles respectively, from left to right.

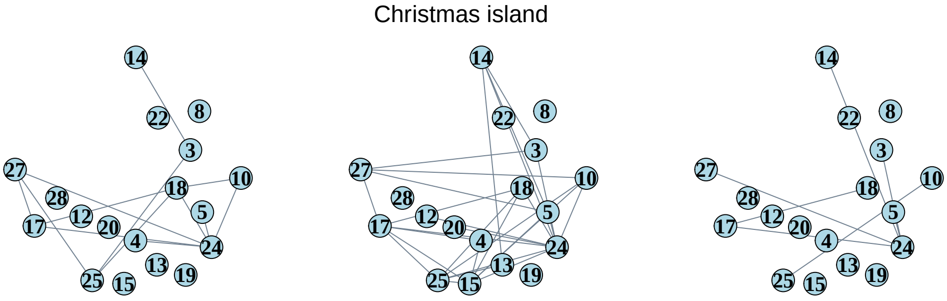

Figure 4.61: Networks representing interactions of fish at three different reefs of the Christmas island.

To fit the SNF model with HS distance, we first tune the prior distributions of the parameters and the IS density. Similarly to the tuning performed for the Bali region in the previous section, we centre the prior of the centroid network at the network observation that minimises the distance from the other network data, being the right network of figure 4.61. The tuning of the prior for γ and the IS density, is performed in the same way as performed for the Bali region. Notably, we specify the single CER model as the IS density of our algorithm, and sample from it using an MCMC sampling scheme. The size of the IS sample drawn is $K = 6,000$ networks, after a burn-in of 1,000 networks.

We run our MCMC with xgboost algorithm for 5,000 iterations and obtain the traceplot of the posterior draws for γ in figure 4.62, the histogram of the posterior distribution for γ in figure 4.63 and the 5 most commonly drawn networks from the posterior of the centroid shown in figure 4.64. We note again that the pink edges of the centroid posterior draws correspond to edges that are also present in the network data.

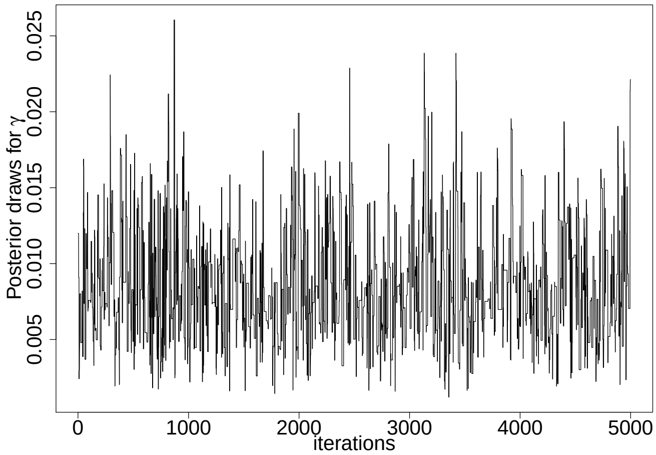

Figure 4.62: Traceplot for γ for 5,000 iterations of the MCMC for data recorded at the Christmas island, using xgboost to predict the HS distance in each iteration.

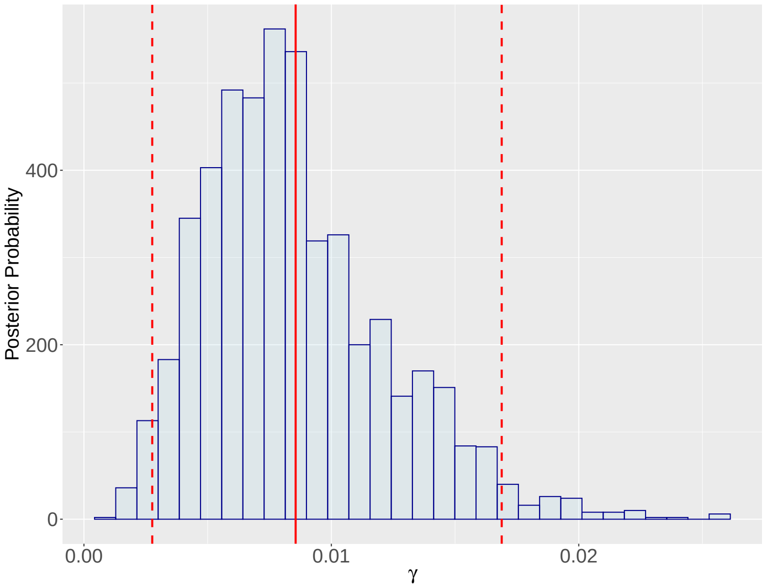

Figure 4.63: Histogram of the posterior distribution of γ for data recorded at the Christmas island, with red solid line indicating the mean 0.0085567, and red dashed lines indicating the 95% credible interval (0.0028,0.017).

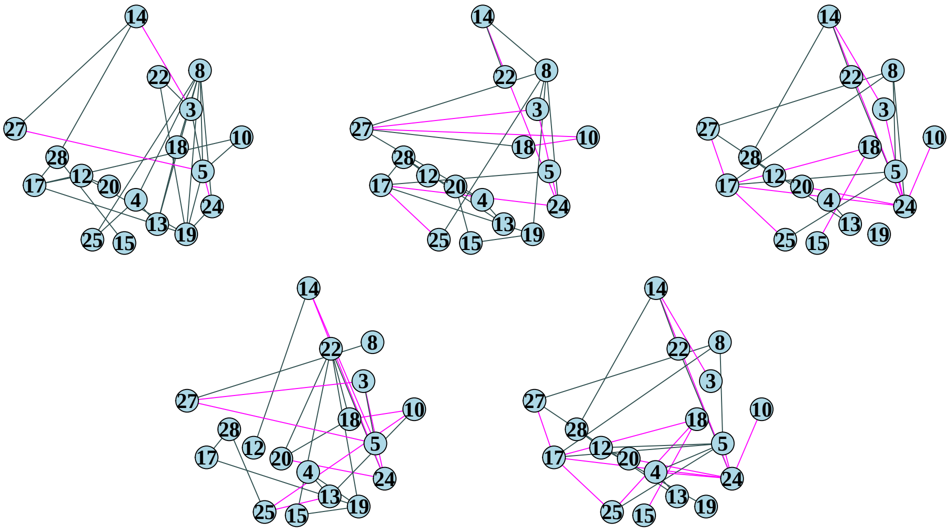

Figure 4.64: Top 5 most commonly drawn centroids with posterior masses from top left to bottom right 0.027, 0.025, 0.023, 0.021 and 0.019 respectively. Pink edges indicate edges that are also present in the data recorded at the Christmas island.

In Figure 4.62, we observe good mixing of the MCMC chain for γ. Similarly to the Bali region, we further notice that the sizes of the posterior draws for γ are small, indicating a high dispersion of the data from the posterior centroids inferred. Moreover, the 5 most commonly drawn centroids involve edges that are not observed in the network data, and concentrate small posterior masses. A reason for the challenging nature of inferring a single centroid for the network data of the Christmas island is the many not in common cycles among the network observations, as the middle network of figure 4.61 contains 1834 cycles while the other two network observations contain only 0 and 12 cycles respectively.

To further investigate the centroid inference, we obtain the 10 most common cycles observed in the posterior centroid draws along with the proportion of the posterior centroids enclosing each cycle, and detect whether these cycles are also observed in the network data. The results for the centroids' cycles are gathered in Table 4.8. We observe that most of the cycles that are frequently observed in the posterior centroids are not present in the network observations as well, except for the cycle between the fish species auriga (node 3), trifascialis (node 24) and lunula (node 14), which is observed in 16% of the posterior centroids drawn. Furthermore, the rest of the cycles presented in Table 4.8 are also observed in a high proportion of the posterior centroids, indicating that those cycles might be meaningful for the network data at hand.

most common cycles	**proportion of times identified**	**observed/inferred**
14-24-22-14	0.2	inferred
5-19-8-5	0.19	inferred
10-25-24-17-10	0.18	inferred
14-27-18-14	0.18	inferred
10-27-18-10	0.17	inferred
15-22-19-15	0.16	inferred
3-24-14-3	0.16	observed
8-24-17-19-8	0.15	inferred
12-28-14-24-20-12	0.15	inferred
3-14-27-5-3	0.14	inferred

Table 4.8: 10 most common cycles detected in posterior draws for centroid network, proportion of times each cycle detected in posterior draws and presence (observed) or absence (inferred) of cycle in the data recorded at the Christmas island.

4.5.3 Aceh region

The third and last real data example that we consider is the multiple network data collected at the Aceh region. The data were recorded at four different reefs of the Aceh region resulting in a collection of $N = 4$ network observations, sharing the same set of $n = 19$ nodes, as depicted in figure 4.65. We notice that the networks of the Aceh region are sparser compared to the Bali region and the Christmas island presented in the previous sections. Accordingly, the number of cycles that each network observation encloses is 3, 0, 6 and 0 respectively, from top left to bottom right of figure 4.65.

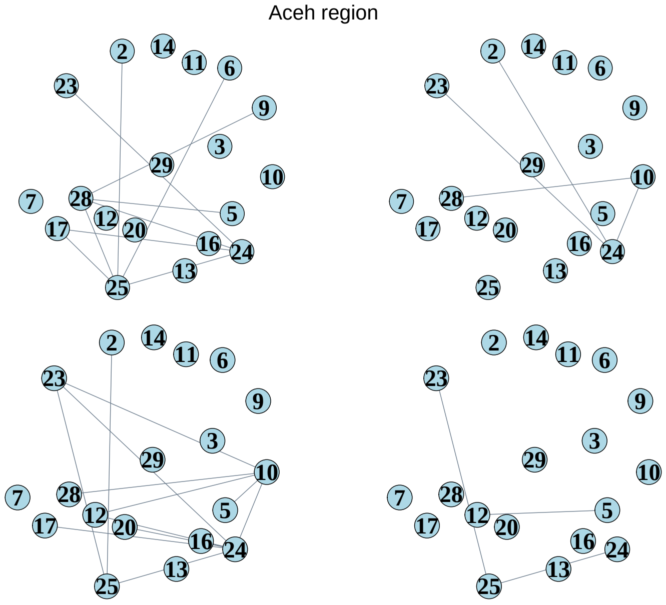

Figure 4.65: Networks representing interactions of fish at four different reefs of the Aceh region.

The tuning of the prior distributions and the IS density is similar to that described in the previous two sections. In this respect, we now centre the prior of the centroid network to the top-right network seen in figure 4.65, as it minimises the HS distance from the rest of the network data. We further specify the CER model as the IS density of our algorithm and sample from it using an MCMC scheme. The IS sample size is $K = 6,000$ networks, after a burn-in of 1,000 networks.

In figures 4.66, 4.67 and 4.68, we present the results after running our MCMC with xgboost algorithm for 5,000 iterations. Figure 4.66 shows the traceplot of the posterior draws for γ, which demonstrates a poorer mixing compared to the MCMC chains of γ obtained for the network data of the two previous regions in sections 4.5.1 and 4.5.2. We further notice that the most frequently drawn centroid networks, depicted in figure 4.68, are denser compared to the network data of the Aceh region. Motivated by this observation, we formulate a restriction on the density of the centroid networks proposed in our MCMC algorithm, to explore whether we can achieve better chain mixing and centroid inference.

We comment here that a similar restriction on the sizes of γ proposals is proposed by Lunagómez et al. [2021]. Specifically, Lunagómez et al. [2021] propose an upper bound for γ according to the information revealed by the EDA boxplots for the SNF model, for each distance metric specification. In this regard, the upper bound of γ is determined by the corresponding size of γ for which the distribution of the distance reaches 0. This restriction is necessary for the MCMC to explore a reasonable region of values for γ.

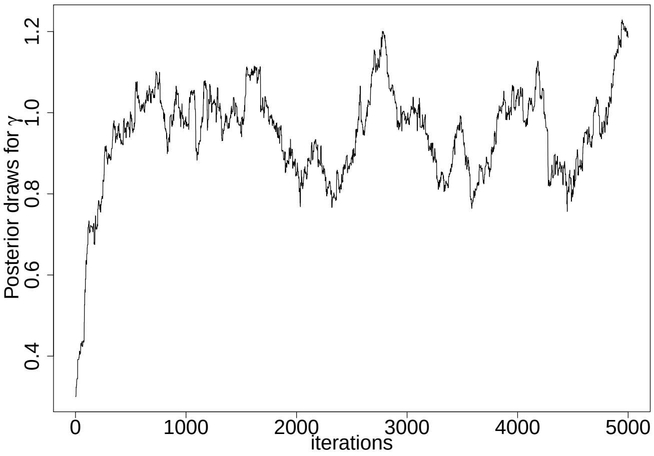

Figure 4.66: Traceplot for γ for 5,000 iterations of the MCMC for data recorded at the Aceh region, using the xgboost algorithm to predict the HS distance in each iteration.

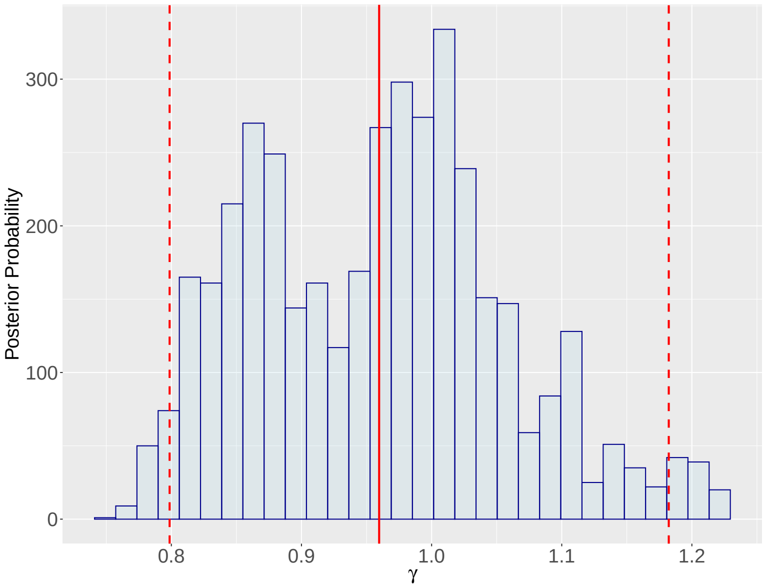

Figure 4.67: Histogram of posterior distribution of γ after a burn-in of 1,000 iterations, for data recorded at the Aceh region, with red solid line indicating the mean 0.96, and red dashed lines indicating the 95% credible interval (0.8,1.18).

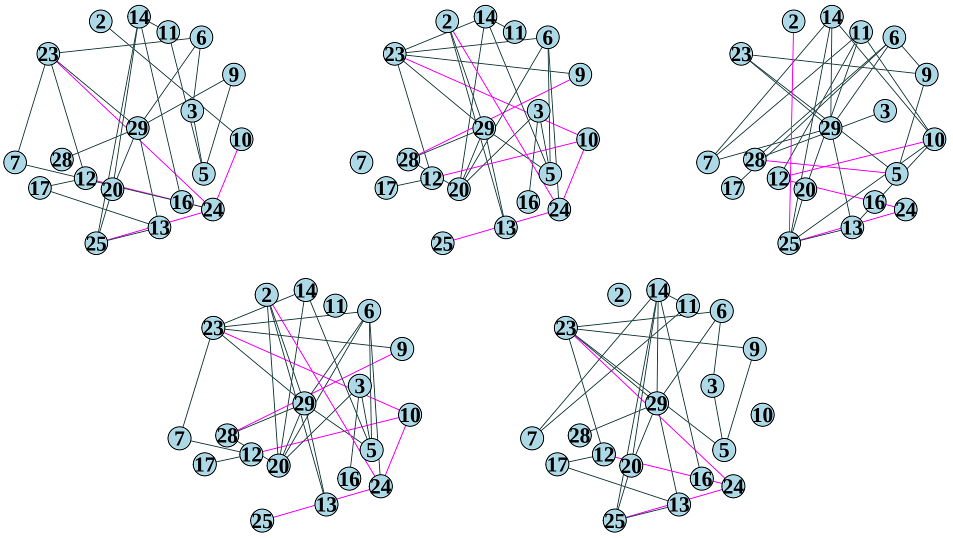

Figure 4.68: Top 5 most commonly drawn centroids with posterior masses from top left to bottom right 0.025, 0.025, 0.024, 0.024 and 0.023 respectively. Pink edges indicate edges that are also present in the data recorded at the Aceh region.

We now present the results of our MCMC with xgboost algorithm and restriction on the density of the centroid proposals. We further specify a mixture of CER models for the IS density and centre it to the same network observation that we previously centred the single CER IS density. Our CER mixture model involves $J = 2$ equally weighted mixture components, thus we specify two alternative sizes of $\tilde{\alpha} = (0.02, 0.07)$. We note here that we specified small values for the two alternative dispersion parameters of the

CER mixture model, as otherwise it would be computationally prohibitive to run our MCMC algorithm. Figures 4.69, 4.70 and 4.71 illustrate the results from 5,000 iterations of our MCMC.

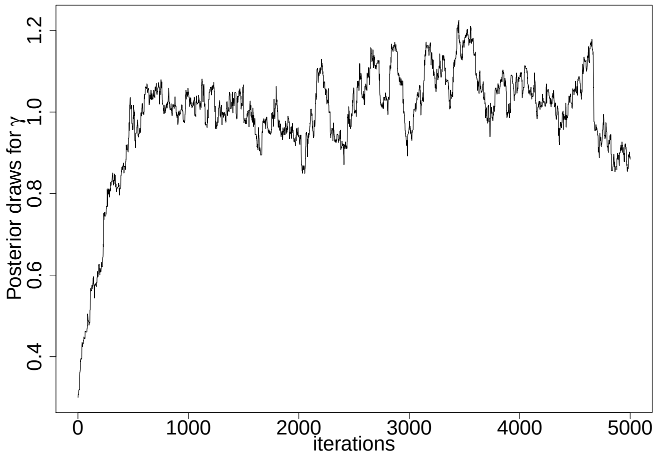

Figure 4.69: Traceplot for γ for 5,000 iterations of the MCMC with restriction, for data recorded at the Aceh region, using xgboost to predict the HS distance in each iteration.

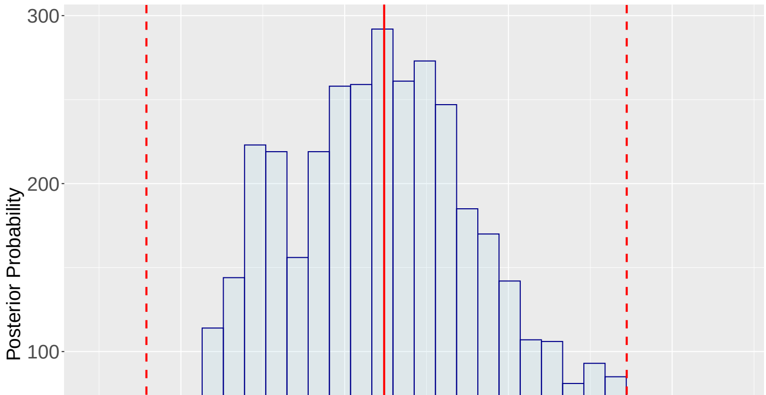

Figure 4.70: Histogram of posterior distribution of γ after a burn-in of 1,000 iterations from MCMC with restriction, for data recorded at the Aceh region, with red solid line indicating the mean 1.02, and red dashed lines indicating the 95% credible interval (0.88,1.17).

Under the restricted MCMC, the mixing of the MCMC chain for γ slightly improved compared to the non-restricted case, and the posterior centroids illustrated in figure 4.71 appear to be more meaningful for the network data of the Aceh region. The region of

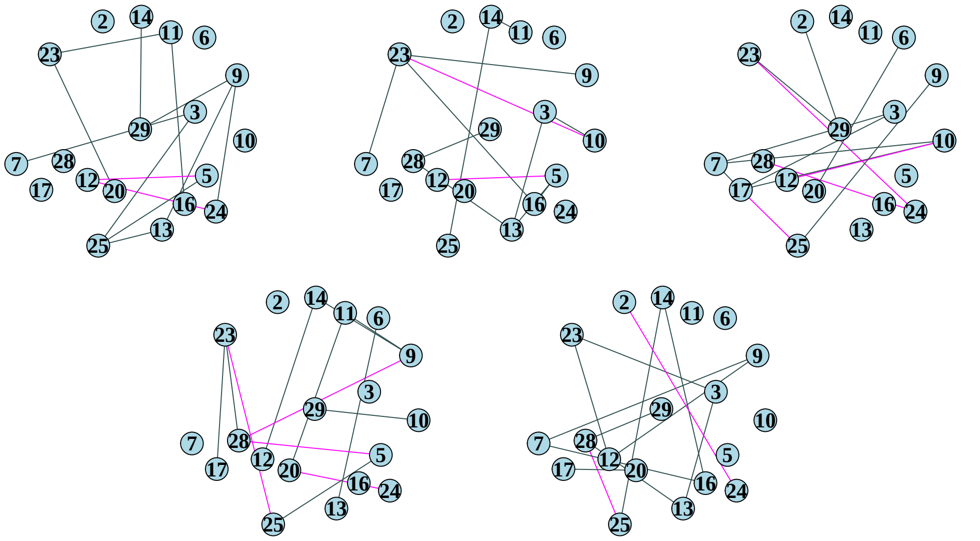

Figure 4.71: Top 5 most commonly drawn centroids with posterior masses from top left to bottom right 0.022, 0.022, 0.021, 0.021 and 0.02 respectively, from MCMC with restriction. Pink edges indicate edges that are also present in the data recorded at the Aceh region.

γ values explored by the MCMC chain indicates small variability of the network data from the centroids drawn, with respect to the HS distance. Notably, both the networks observations of the Aceh region and the posterior centroids involve only a small number of cycles, resulting in a small number of not in common cycles between the network data and each centroid drawn. Specifically, each posterior centroid depicted in figure 4.71 has only 1 cycle, except for the top right centroid draw which encloses 3 cycles.

We note here that another possible direction for exploring the behaviour of our model in inferring the complex posterior distribution of the parameters for the Aceh region data, would be to investigate how the sampling of γ affects the model performance. This could be achieved by treating γ as fixed in our inferential scheme, and perform a search over different sizes of γ to investigate how the performance of our algorithm in inferring a centroid changes. In this way, we would avoid the additional complexity of sampling γ from its posterior distribution, which accordingly affects centroid inference.

Similarly to the results obtained from the previous data examples of sections 4.5.1 and 4.5.2, the posterior masses that the most commonly drawn centroids concentrate are approximately equal to 0.02. Due to this result, we cannot summarise information about the network data from a single centroid inferred. Thence, we investigate which cycles are mostly observed in the posterior centroids drawn, similarly to the previous

most common cycles	**proportion of times identified**	**observed/inferred**
7-17-10-7	0.12	inferred
7-29-9-7	0.11	inferred
5-28-23-25-5	0.11	inferred
3-17-7-3	0.08	inferred
3-25-6-3	0.06	inferred
3-17-10-7-3	0.06	inferred
3-23-9-25-28-13-3	0.05	inferred
5-25-13-9-24-12-5	0.05	inferred
3-29-13-3	0.05	inferred
10-24-23-10	0.05	observed

Table 4.9: 10 most common cycles detected in posterior draws for the centroid network, proportion of times each cycle is detected in posterior draws and presence (observed) or absence (inferred) of cycle in the data recorded at the Aceh region, from the MCMC with restriction.

sections.

From the cycle detection task, we observed that only 48% of the centroid posterior draws involved cycles. The rest 52% of the centroids drawn did not enclose any cycles. In Table 4.9 we present the 10 most common cycles found in the posterior centroids that involved cycles, along with the proportion of times that each cycle was detected in the posterior sample of centroids. In addition, we identify which of the cycles are also observed in the network data.

We observe that only one of the 10 most common cycles found in the centroid posterior draws is also identified in the data, which is the one involving the fish species guttatossimus (node 10), trifascialis (node 24) and triangulum (node 23). The rest of the cycles are not identified in the network observations of the Aceh region, however, they are repeatedly found in the posterior centroid draws, which suggests that these cycles can be informative for the network observations of the Aceh region. Nonetheless, the proportion of centroids inferred involving no cycles is significant.

4.6 Discussion

In this Chapter we proposed an Importance Sampler as an alternative to the Auxiliary Variable technique implemented by Lunagómez et al. [2021], that allows us to make inferences for the SNF model that involves an intractable normalising constant. The IS formulated allowed better chain mixing for distance metric specification for which the

Auxiliary Variable technique provided poor mixing results. In addition, we introduced a new network distance metric, namely the HS distance, that quantifies dissimilarities between networks with respect to their cycles. The motivation for developing the HS distance metric was the ecological application presented in section 4.3.1. Nonetheless, as cycles are a common network feature of interest, we believe that the HS distance can be an informative measure of dissimilarity for other network applications as well.

To better understand the behaviour of the HS distance, we performed synthetic data experiments and examined the performance of the HS distance in identifying different network structures and characteristics. The MDS projections and the FR test performed highlighted the importance of the inclusion of the Hamming distance and the weight factor λ in the HS distance, with respect to the performance of the metric in capturing information about the networks' structure. This is an anticipated result if we consider that the Hamming distance is a structural distance metric, as presented in Donnat and Holmes [2018], while the symmetric difference focuses on specific information about the networks' topology.

From the synthetic experiments performed for the HS distance, we further noticed that the metric is able to detect similarities among networks with similar number of cycles. Nonetheless, an interesting finding is that the networks involving the largest number of cycles were not identified as similar to any of the other simulated networks. This result suggests that the HS distance might fail to detect similarities among networks with many cycles, even when the networks share a similar structure. This is a reasonable finding if we consider that the more cycles two networks have, the more likely it is that they will have less cycles in common. Thence, it is important to adjust the weight λ according to the characteristics we want to reveal for the networks.

In Section 4.4, we further explored the performance of the MCMC with IS step for the Jaccard and the HS distance, which are both metrics for which the implementation of the MCMC with the Auxiliary variable technique presented in Lunagómez et al. [2021], resulted in poor chain mixing. The simulation results led to a number of interesting findings. First, for the 5-node network simulation study, we observed that our MCMC with IS step performed well in inferring the dispersion parameter γ, and this was further supported by the similar inferential results obtained under the formulation of an MCMC with the exact calculation of the normalising constant in each iteration.

However, a key observation arising from the simulations performed is the bias for large sizes of N, which was also discussed in Section 4.3.6. As previously commented, Vitelli et al. [2017] propose that a large IS sample can adjust for the bias with increasing N. The impact of the size of the IS sample in the bias was also noticed by the investigation we performed in section 4.4.2. In this respect, the bias for increasing N observed for the 20-node case could be potentially corrected by a very large IS sample. However, for the case of the SNF model with the HS distance specification, drawing a drastically large IS sample is computationally prohibitive.

Another finding from the investigation performed about the bias of our IS estimator in Section 4.4.2, was the impact of the choice of the IS density and the IS sampling scheme. In this respect, we observed that the mixture of CER models IS density provided more accurate inferential results, regardless of the small IS sample size K and large population N. A similar finding resulted from the Monte Carlo sampling scheme implemented to sample from the IS density being the CER model. The better performance of the algorithm under these two cases can be attributed to the greater variability of the networks sampled, resulting in heavier tailed IS densities which is an important condition for the good performance of the IS (Robert and Casella [2013]). However, for the 20-node networks the mixture of CER models IS density did not improve the results with respect to the bias. Nonetheless, we observed that our method can still provide accurate results for small populations of networks.

With respect to the inferential results for the centroid network, we observed that for the simulation regimes with smaller values of γ indicating a higher variability in the network population simulated, the true centroid was harder to be identified. This is a reasonable finding as the networks simulated are significantly different from the centroid. Equivalently, small populations of networks N made the inference of the centroid harder, which can be justified by the less information encoded in the data due to the fewer observations.

For the SNF model with HS distance, another triggering factor for the challenging nature of inferring a centroid is the behaviour of the HS distance metric with respect to identifying similarities among networks with many cycles as previously noted. Specifically, we noticed from both the 20-node network simulation studies and the real data examples, that different posterior centroid networks concentrate low posterior masses.

From the real data examples we noticed that inferring a single centroid is challenging for a multiple network data set that involves a network observation with a large number of cycles, while the rest of the network observations involving only a small number of cycles. This is reasonable if we consider that a centroid aims to minimise the distance from all the network observations simultaneously.

An extension to our framework would be the incorporation of information about edge weights. As discussed in Section 4.5, the network representations of the fish interactions have weighted edges, corresponding to the multiple interactions observed between species of fish. The incorporation of this network characteristic would require two modifications in our framework. First, we would need to modify our HS distance by replacing the Hamming distance part with a distance metric that quantifies dissimilarities between weighted graphs. As highlighted from the synthetic data experiments the inclusion of the Hamming distance to the HS distance metric is important to capture structural information about the networks. An alternative metric that could be used in the case of weighted networks is the Frobenius distance. In this respect we would need to perform synthetic data experiments to explore the behaviour of the new combination of distance metrics.

The second and most substantial modification that would be required in order to incorporate edge weight information, is the formulation of the MCMC to allow sampling networks from the weighted space of graphs. This would be essential for two reasons, (i) to draw proposal centroids and (ii) to draw IS samples of weighted graphs. For the first case, we would need to modify our MCMC scheme to also update the edge weights of the proposed network. In this case, a Uniform distribution could be an option to sample a new vector of weights, and adjust the weights of the current centroid according to the newly drawn weights. For the second case, we would need the specify an IS density other than the CER model, as the CER is not a model for weighted graphs. We note that the specification of the SNF model as the IS density would not be feasible due to the normalising constant that would appear in the denominator of $\hat{Z}$, thence such a formulation is not straightforward.

As described in this chapter, there is a vast literature on techniques developed to sample from intractable distributions. In our study we proposed the implementation of an IS to tackle the challenges arising with the Auxiliary Variable technique implemented for the SNF model in Lunagómez et al. [2021]. An alternative approach to sample from the posterior of the intractable SNF model would be the Noisy Exchange algorithm reviewed in section 4.2.2. This method requires sampling from the SNF model which is feasible by using an MCMC scheme. In this respect the sample obtained from the SNF model would be used to substitute the ratio of normalising constants by the sum of the ratio of the unnormalised SNF likelihood. However this would not fix the bias due to N.

Chapter 5

Conclusions and Further Work

In this thesis we developed new statistical frameworks that allow inferences for multiple network data sets. In Chapter 3, a new framework was proposed for clustering heterogeneous network populations, while in Chapter 4 we proposed an MCMC scheme to make inferences for the SNF model. In both Chapters 3 and 4, we discussed potential extensions of the work presented. In Sections 5.1 and 5.2 of this Chapter, we further introduce two new directions for future work for the analysis of multiple network data, while in Section 5.3 we conclude with our general view on future research in the context of networks.

5.1 Dependent network data

A common assumption made in the literature for modelling multiple network data is that the network observations in the population are independent. Even though studies focusing on the analysis of single network observations have considered the problem of edge dependence in a network (Hoff [2003], Chen et al. [2020]), the problem of modelling the dependence among network observations has yet to be analysed. A reason why the analysis of dependencies among network observations has not yet been considered, is the recent need in analysing multiple network data sets due to their currently increasing availability.

The real data examples presented in Chapters 3 and 4 can serve as cases where dependencies among the observed networks may be present. First, the Tacita real data presented in Chapter 3 can be analysed as a multiple network data set in various ways.

As presented in Chapter 3, one way to analyse the data recorded by the Tacita application is to consider the aggregated movements of the users recorded during different times of the day and different days of the week, resulting in a set of multiple network observations corresponding to different individuals. Another way to analyse this data would be to consider the movements of each individual depending on specific hours of the day (morning, afternoon, evening, night), during different days of the week. In this case, we would obtain network populations corresponding to movements of the users during a specific time interval of the day, thus the network populations would be comprised by network observations corresponding to the same individuals. One research question arising in the latter set up is whether we can detect different movement patterns of the individuals during different times of the day, thus compare dependent network populations.

Second, the ecological application that motivated the study of Chapter 4 can also be considered under another perspective. Notably, in our analysis we focused on fitting the SNF model with the HS distance metric to three network populations corresponding to three different regions of the Indo-Pacific ocean. However, for some of these regions, data were collected both before and after a coral bleaching event (Hughes et al. [2017], McClanahan et al. [2009]). Thus, for each region, we have two dependent populations of networks: one population of networks resulting from measurements taken before the bleaching event, and one population of networks resulting from measurements taken after the bleaching event. A research question arising in this set up is whether we can detect structural changes between two dependent populations of fish networks, before and after a bleaching event.

One way to address these research questions would be through the formulation of a hypothesis testing framework for comparing dependent network populations. As reviewed in Chapter 2, Ginestet et al. [2017] developed a frequentist hypothesis testing framework for comparing populations of networks, under the assumption that the network populations are independent. Thus, an interesting extension to the hypothesis testing problem would not only be the consideration of dependent network populations, but also the formulation a Bayesian hypothesis testing framework, e.g. using Bayes factor (Robert [2007]).

The research problems described above focus on comparing populations of networks that are assumed to be dependent. An alternative way to think about dependence in the context of networks, would be to model a network population assuming dependence among the network observations therein. The brain network data presented in Chapter 3 can be regarded as a population of dependent network observations. Specifically, the brain network population was constructed from repeated measurements taken over a set of individuals. Thus, in this data set we have network observations corresponding to the same individuals. Our study, along with the studies of Arroyo et al. [2019] and Lunagómez et al. [2021], model this network population without accounting for potential dependencies among the network observations.

In this regard, an interesting extension to our work would be to model dependent observations for a population of networks, building on the studies of Le et al. [2018] and Lunagómez et al. [2021]. Specifically, we may model dependence among network-valued observations by allowing the deviations from the centroid/true network to be correlated. We now introduce some additional notation to make this notion precise. Let $H_k(i,j) = \mathrm{xor}(\mathcal{G}_k(i,j), \mathcal{G}^m(i,j))$, denote the perturbation of network observation $\mathcal{G}_k$ with respect to the centroid/true network $\mathcal{G}^m$, for $k \in \{1,\ldots,N\}$. The measurement error models introduced in Newman [2018a] and Le et al. [2018] assume that H_k are independent for each k. However, this assumption is not realistic for datasets such as the brain multiple network data set presented in Chapter 3. One way to overcome this limitation would be to assume that the perturbations are such that the random vectors $(H_k(i,j), H_l(i,j)),\ k,l \in \{1,\ldots,N\}$, follow a multivariate Bernoulli as discussed by Teugels [1990], accounting for the dependencies among the Bernoulli random variables.

5.2 Anomaly detection

Many recent applications involve networks that evolve over time, and examples include social networks representing relationships among actors changing over time, or computer networks representing interactions between Internet Protocol (IP) addresses. This type of networks are often introduced as dynamic networks in the network literature. Dynamic networks are networks that can exhibit changes in their structure with respect to their nodes, edges or attributes. In light of this, researchers are interested in the problem of anomaly detection in dynamic networks, with goal the detection of changes in the networks' structure over time. Type of anomalous behaviour can emerge from the networks' nodes, edges and/or underlying subgraphs (Ranshous et al. [2015]).

A common characteristic of the methods developed for anomaly detection is the formulation of a two stage approach, where the first step involves mapping the graphs to a common graph representation, while the second step involves scoring the anomalousness of the graph representations obtained in the first step (Ranshous et al. [2015]). In a probabilistic set up, this requires the construction of a model that best describes the "normal" network data and then detect a graph as anomalous with some probability. Such approaches are presented in Heard et al. [2010] who build a Bayesian framework for anomaly detection, Priebe et al. [2005] who employ a scan statistics method on the nodes of the graph to detect anomalous behaviour and the recent work by Lee et al. [2019] who perform anomaly detection on networks assuming a latent space representation.

The anomaly detection problem links to the outlier cluster detection introduced in Chapter 3, which was formulated to identify network data that differ from the majority with respect to their structure. However, under our construction there is no time component characterising the networks. An interesting extension to the outlier cluster detection presented would be its formulation for networks characterised by a time component. The network applications considered in both Chapter 3 and 4 can serve us examples of time dependent networks. Specifically, one way to view the data recorded by the Tacita application, is to consider the movements of individuals over time. In this respect, we could investigate changes in their movements over time, and specifically, an interesting extension would be to consider users' movements recorded before and after the Covid-19 pandemic.

The second application presented in this thesis that can serve as an example of time dependent networks, is the ecological study measuring interactions between different species of fish, introduced in Chapter 4. As previously discussed, this data can be modelled as dependent network data as the recording of the fish interactions took place in two different time stamps, before and after a coral bleaching event. Another way to view the dependence among the fish networks is to index them by the time they were observed. Under this set up, it would be interesting to investigate changes in the structure of the networks after the bleaching event by performing anomaly detection. In this regard, the aim would be to model the distribution of the data before the bleaching event, with respect to the cycles formed in the networks, playing the role of history data, and quantify the probability of a network to be anomalous after the bleaching event. Thus an interesting extension could be developed for the SNF model to perform outlier cluster detection. A relevant work is that introduced by Aggarwal et al. [2011], however they perform outlier detection on graph streams which are series of graphs $\{G_t\}_{t=1}^{T}$ for $T \to \infty$.

5.3 Closing remarks

One of the biggest challenges in the field of statistics and machine learning is the ability to analyse big data (Chauhan and Sood [2021], Espinosa et al. [2019]). In this context, the availability of massive networks involving billions of nodes and edges is continuously increasing (Chung [2018]). The storage and analysis of such big network data require substantial computing resources, however, such computing resources are not accessible to everyone. In light of this, it is essential not only to formulate scalable algorithms for network analysis (Teng [2016]), but also to develop modelling frameworks in such way that they will require less computing resources.

Two application examples which involve the analysis of massive network data, arise from the fields of cyber-security and social network analysis. In cyber-security, a type of network representation commonly analysed is the traffic (edges) recorded among IP addresses (nodes) (Wang and Jones [2020]). Such networks become drastically massive in size and often their online analysis is required. On the other hand, a social network instance can represent interactions (edges) among individuals (nodes) through the use of Internet services (e.g. Facebook, Twitter) (Sapountzi and Psannis [2018]). This type of network data are huge in size considering the billions of Internet users (Chauhan and Sood [2021]).

Common characteristics of the network data in the applications described above among others, are their continuous evolving nature through time (dynamic networks), as well as their multiple views or measurements taken (multiple network data discussed in this thesis), which both give rise to challenges considering the huge number of nodes and edges that they usually involve. In this regard, we believe that the development of modelling frameworks and scalable algorithms for big network data will be an active research area in the forthcoming years.

Appendix A

Appendix for 'Bayesian model-based clustering for multiple network data'

A.1 MCMC scheme

In this Section we provide the full conditional posteriors for the model parameters that are updated through a Gibbs sampler, as discussed in Section 3.3.4.

The full conditional posterior for the probability of a network to belong to cluster $\boldsymbol{\tau}$ is given by

$$P(\boldsymbol{\tau}|\boldsymbol{A}_{\mathcal{G}^*}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) \propto P(\boldsymbol{z}|\boldsymbol{\tau}) \cdot P(\boldsymbol{\tau}|\boldsymbol{\psi})$$

where $P(\boldsymbol{z}|\boldsymbol{\tau}) = \text{Multinomial}(1; \tau_1, \ldots, \tau_C)$ and $P(\boldsymbol{\tau}|\boldsymbol{\psi}) = \text{Dirichlet}(\boldsymbol{\psi})$, as specified in Section 3.3.3 of the main article. Thus, we have

$$\begin{aligned} P(\boldsymbol{\tau}|\boldsymbol{A}_{\mathcal{G}^*}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) &\propto \prod_{j=1}^{N} \tau_{z_j} \cdot \Gamma(\psi C) \cdot \Gamma(\psi)^{-C} \cdot \prod_{c=1}^{C} \tau_c^{\psi-1} \\ &\propto \Gamma(\psi C) \cdot \Gamma(\psi)^{-C} \cdot \tau_{z_1} \cdots \tau_{z_N} \cdot \tau_1^{\psi-1} \cdots \tau_C^{\psi-1} \\ \propto \Gamma(\psi C) \cdot \Gamma(\psi)^{-C} \cdot \tau_1^{\eta_1} \cdots \tau_C^{\eta_C} \cdot \tau_1^{\psi-1} \cdots \tau_C^{\psi-1} &\propto \tau_1^{\eta_1+\psi-1} \cdots \tau_C^{\eta_C+\psi-1} \end{aligned} \tag{A.1}$$

where $\eta_c = \sum_{j=1}^{N} \nVdash_c(z_j)$, $c = 1, \ldots, C$, denotes the number of network data that belong to cluster c. Thence we obtain,

$$P(\boldsymbol{\tau}|\boldsymbol{A_{\mathcal{G}^*}}, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) = \text{Dirichlet}(\psi + \eta_1, \ldots, \psi + \eta_C).$$

The derivation of the full conditional posterior for the vector of the nodes' block-membership probabilities $\boldsymbol{w_c}$ for cluster c, is similar to the derivation of the the full conditional posterior for $\boldsymbol{\tau}$, as already described above, thus we have

$$P(\boldsymbol{w_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{\theta_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) \propto P(\boldsymbol{b_c}|\boldsymbol{w_c}) \cdot P(\boldsymbol{w_c}|\boldsymbol{\chi})$$

where $P(\boldsymbol{b_c}|\boldsymbol{w_c}) = \text{Multinomial}(\boldsymbol{w_c})$ and $P(\boldsymbol{w_c}|\boldsymbol{\chi}) = \text{Dirichlet}(\boldsymbol{\chi})$, as specified in Section 3.3.2. Hence we obtain

$$\begin{aligned} P(\boldsymbol{w_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{\theta_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) &\propto \prod_{i=1}^{n} w_{c,b_i} \cdot \Gamma(\chi K) \cdot \Gamma(\chi)^{-K} \cdot \prod_{k=1}^{K} w_{c,k}^{\chi-1} \\ &\propto \Gamma(\chi K) \cdot \Gamma(\chi)^{-K} \cdot w_{c,b_1} \cdots w_{c,b_n} \cdot w_{c,1}^{\chi-1} \cdots w_{c,K}^{\chi-1} \\ \propto \Gamma(\chi K) \cdot \Gamma(\chi)^{-K} \cdot w_{c,1}^{h_1} \cdots w_{c,K}^{h_K} \cdot w_{c,1}^{\chi-1} \cdots w_{c,K}^{\chi-1} &\propto w_{c,1}^{(h_1+\chi)-1} \cdots w_{c,K}^{(h_K+\chi)-1}. \end{aligned}$$

where h_k denotes the number of the nodes that belong to block k. Thus the full conditional posterior for $\boldsymbol{w_c}$ is

$$P(\boldsymbol{w_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{\theta_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) = \text{Dirichlet}(\chi + h_1, \ldots, \chi + h_K).$$

The full conditional posterior for the vector of the block-specific probabilities of an edge occurrence $\boldsymbol{\theta_c}$, for the network representative of cluster c is

$$P(\boldsymbol{\theta_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{w_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) \propto P(A_{\mathcal{G}_c^*}|\boldsymbol{w_c}, \boldsymbol{b_c}, \boldsymbol{\theta_c}) \cdot P(\boldsymbol{\theta_c}|\boldsymbol{\epsilon}, \boldsymbol{\zeta})$$

where $P(A_{\mathcal{G}_c^*}|\boldsymbol{w_c}, \boldsymbol{b_c}, \boldsymbol{\theta_c}) = \text{SBM}(\boldsymbol{w_c}, \boldsymbol{b_c}, \boldsymbol{\theta_c})$ and $P(\boldsymbol{\theta_c}|\boldsymbol{\epsilon}, \boldsymbol{\zeta}) = \text{Beta}(\boldsymbol{\epsilon}, \boldsymbol{\zeta})$, as specified in Section 3.3.2. Thus,

$$
\begin{aligned}
&P(\boldsymbol{\theta_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{w_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) \\
&\propto \prod_{(i,j):i<j} \theta_{c,b_ib_j}^{A_{\mathcal{G}_c^*}(i,j)} (1-\theta_{c,b_{c,i}b_{c,j}})^{1-A_{\mathcal{G}_c^*}(i,j)} \cdot \prod_{k=1}^{K}\prod_{l=1}^{K} \theta_{c,kl}^{\epsilon_{kl}-1}(1-\theta_{c,kl})^{\zeta_{kl}-1} \\
&\propto \prod_{k=1}^{K}\prod_{l=1}^{K} \theta_{c,kl}^{A_{\mathcal{G}_c^*}[kl]}(1-\theta_{c,kl})^{n_{c,kl}-A_{\mathcal{G}_c^*}[kl]}\theta_{c,kl}^{\epsilon_{kl}-1}(1-\theta_{c,kl})^{\zeta_{kl}-1} \\
&\propto \prod_{k=1}^{K}\prod_{l=1}^{K} \theta_{c,kl}^{A_{\mathcal{G}_c^*}[kl]+\epsilon_{kl}-1}(1-\theta_{c,kl})^{n_{c,kl}-A_{\mathcal{G}_c^*}[kl]+\zeta_{kl}-1},
\end{aligned}
$$

where $A_{\mathcal{G}_c^*}[kl] = \sum_{(i,j):b_{c,i}=k,b_{c,j}=l} A_{\mathcal{G}_c^*}(i,j)$ represents the sum of the entries for the pairs of nodes of network representative of cluster c, that have block membership k, l respectively, and $n_{c,kl} = \sum_{(i,j):i\neq j} \mathbb{I}(b_{c,i} = k, b_{c,j} = l)$ representing the number of the pair of nodes of representative of cluster c that have membership k, l accordingly. Hence we obtain

$$
P(\boldsymbol{\theta_c}|A_{\mathcal{G}_c^*}, p_c, q_c, \boldsymbol{z}, \boldsymbol{\tau}, \boldsymbol{b_c}, \boldsymbol{w_c}, A_{\mathcal{G}_1}, \ldots, A_{\mathcal{G}_N}) = \text{Beta}(A_{\mathcal{G}_c^*}[kl] + \epsilon_{kl}, \zeta_{kl} + n_{c,kl} - A_{\mathcal{G}_c^*}[kl]).
$$

A.2 Simulation study for moderate network sizes

We now demonstrate additional results for the Simulation study of Section 3.4.1. Specifically, in Figures A.1, A.3 and A.5, we present the traceplots for simulation regime 3 of Table 3.1 for θ. We demonstrate the last 50,000 iterations due to the traceplots being very dense. In addition, we present the histograms for θ after burn-in of 150,000 iterations in Figures A.2, A.4 and A.6.

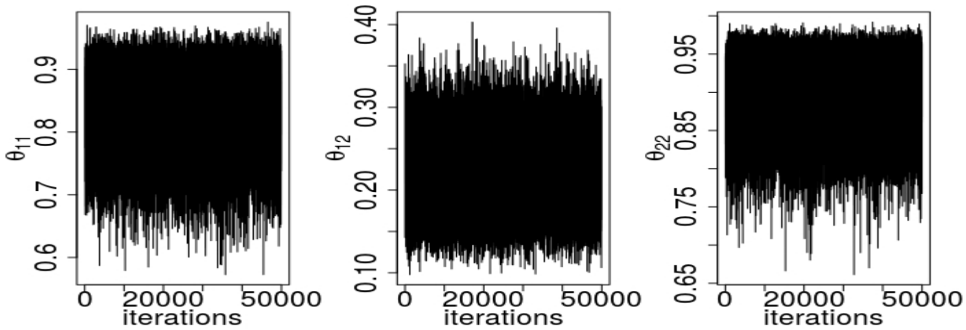

Figure A.1: Traceplots of the posterior draws for θ_{11}, θ_{12}, θ_{22} block probabilities of an edge occurrence for cluster 1.

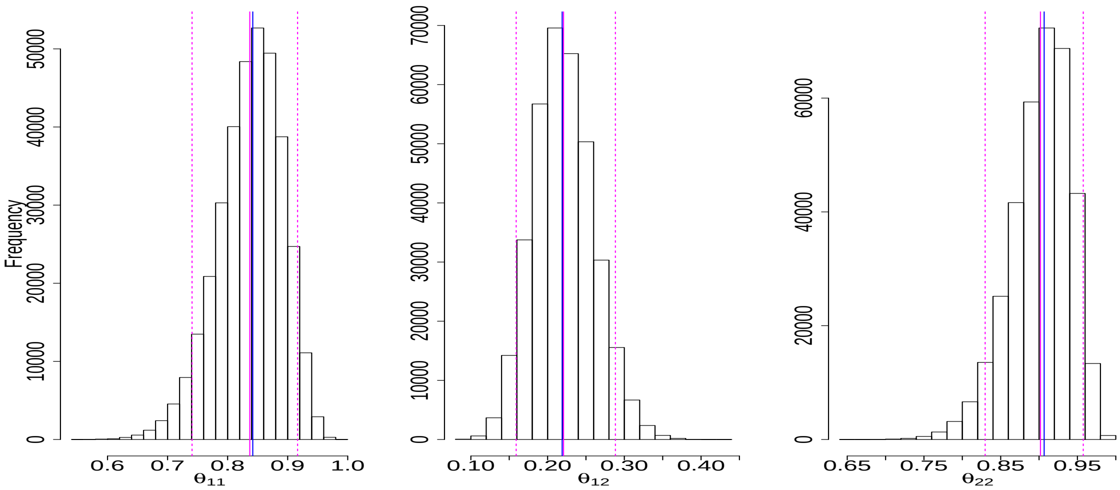

Figure A.2: Histograms for θ_{11}, θ_{12}, θ_{22} block probabilities of an edge occurrence for cluster 1.The pink solid line indicates the posterior mean, the blue solid line the posterior median and the pink dashed lines indicate the 95% credible interval.

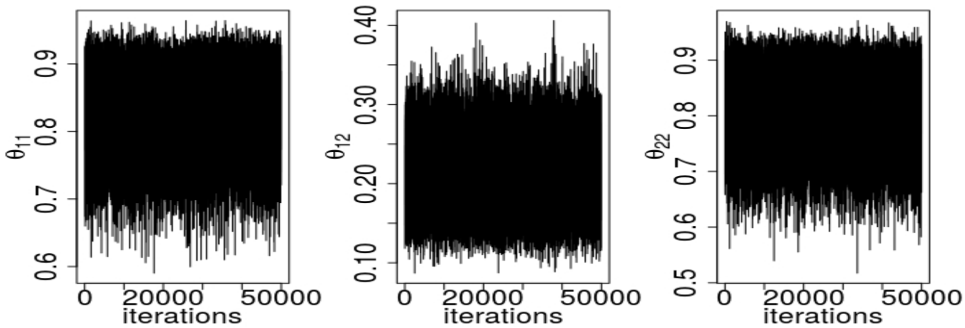

Figure A.3: Traceplots of the posterior draws for θ_{11}, θ_{12}, θ_{22} block probabilities of an edge occurrence for cluster 2.

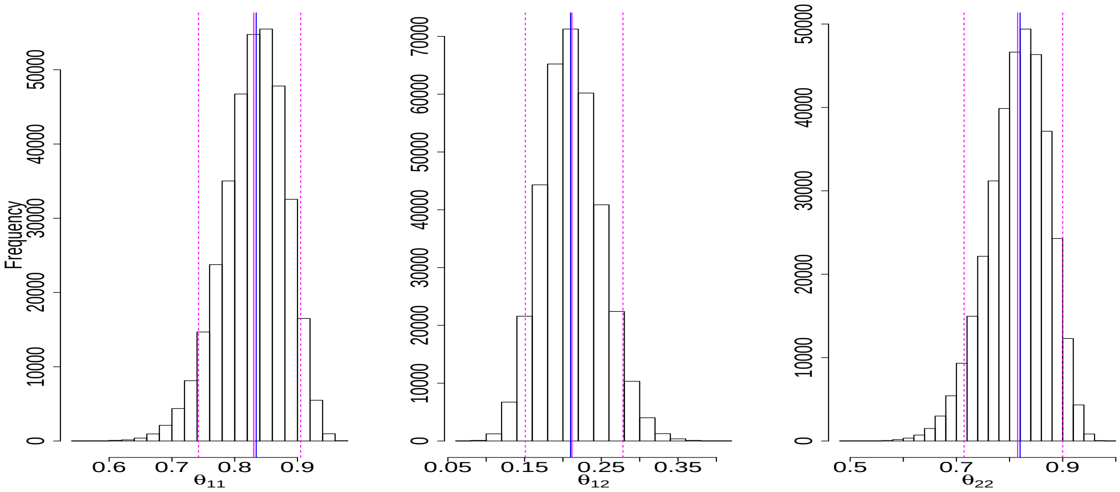

Figure A.4: Histograms for θ_{11}, θ_{12}, θ_{22} block probabilities of an edge occurrence for cluster 2. The pink solid line indicates the posterior mean, the blue solid line the posterior median and the pink dashed lines indicate the 95% credible interval.

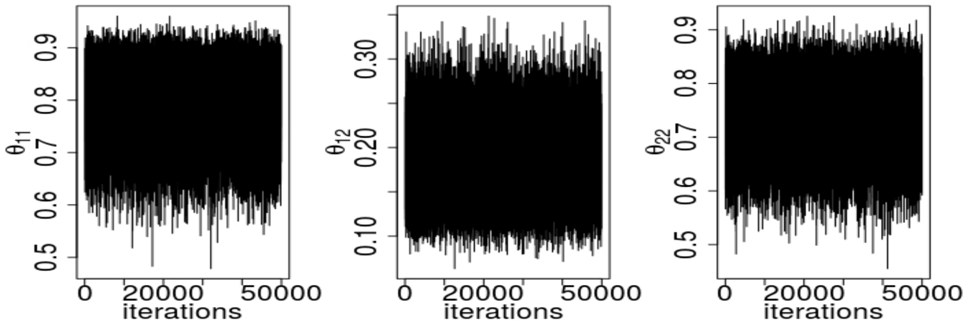

Figure A.5: Traceplots of the posterior draws for θ_{11}, θ_{12}, θ_{22} block probabilities of an edge occurrence for cluster 3.

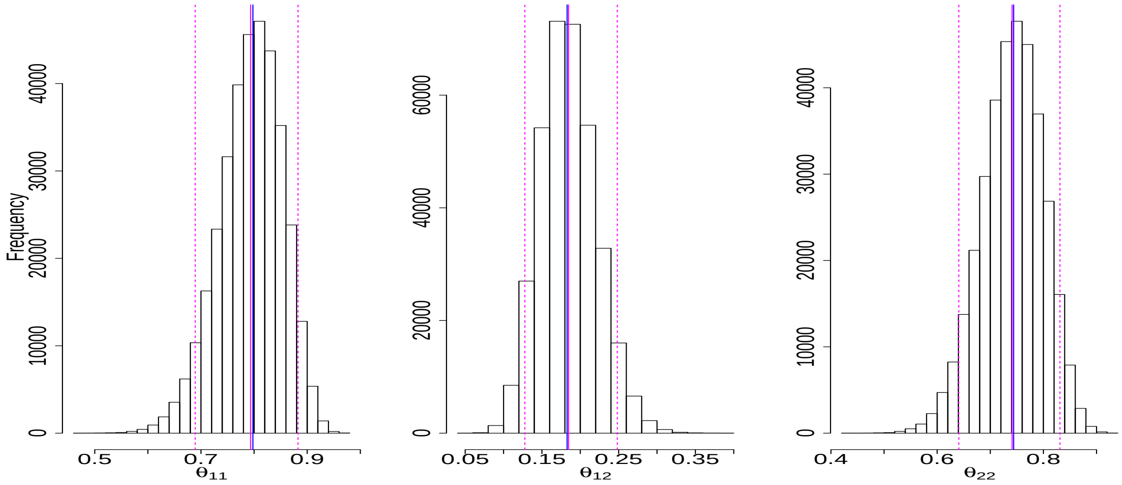

Figure A.6: Histograms for θ_{11}, θ_{12}, θ_{22} block probabilities of an edge occurrence for cluster 3. The pink solid line indicates the posterior mean, the blue solid line the posterior median and the pink dashed lines indicate the 95% credible interval.

A.3 Real data examples

We now present the distance metrics used to perform EDA and initialise the algorithms for both the Tacita data and the brain networks data.

A spectral distance which utilises the eigenvalue decomposition of some matrix representation of the graph (e.g. the adjacency matrix), is the l_p distance. The general form of the l_p distance between graphs $\mathcal{G}$ and $\tilde{\mathcal{G}}$, as introduced in Donnat and Holmes [2018] is

$$d(\mathcal{G}, \tilde{\mathcal{G}})^p = \sum_{i=0}^{R-1} |f(\lambda_i + \epsilon_i) - f(\lambda_i)|^p,$$

where $\lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_R - 1$ are graph's eigenvalues and $f(\cdot)$ is any (almost everywhere) differentiable function.

In the EDA performed for both applications of Section 3.5, we implemented the l_2 distance and specified the low-pass filters $f(\lambda) = e^{-0.1\lambda}$, similarly to the formulation seen in Donnat and Holmes [2018] for the application on Microbiome data.

Another metric discussed in Donnat and Holmes [2018] that we adapted for the real data analysis, is the distance that uses heat spectral wavelets. This metric captures neither too global nor too local changes between two graphs $\mathcal{G}$ and $\tilde{\mathcal{G}}$, and has the form

$$d(\mathcal{G}, \tilde{\mathcal{G}}) = \frac{1}{n}\mathrm{Tr}[\Delta^T \Delta]$$

where $\Delta = Ue^{-\tau\Lambda}U^T - \tilde{U}e^{-\tau\tilde{\Lambda}}\tilde{U}^T$, with $U\Lambda U^T$ and $\tilde{U}\tilde{\Lambda}\tilde{U}^T$ being the eigenvalue decomposition of the laplacian matrices of graphs $\mathcal{G}$ and $\tilde{\mathcal{G}}$ respectively.

A.3.1 Tacita data

In Figure A.7, we present the results for $\boldsymbol{\theta_2}$ for representative of cluster $c = 2$, for clustering algorithm applied on whole data set and $C = 2$.

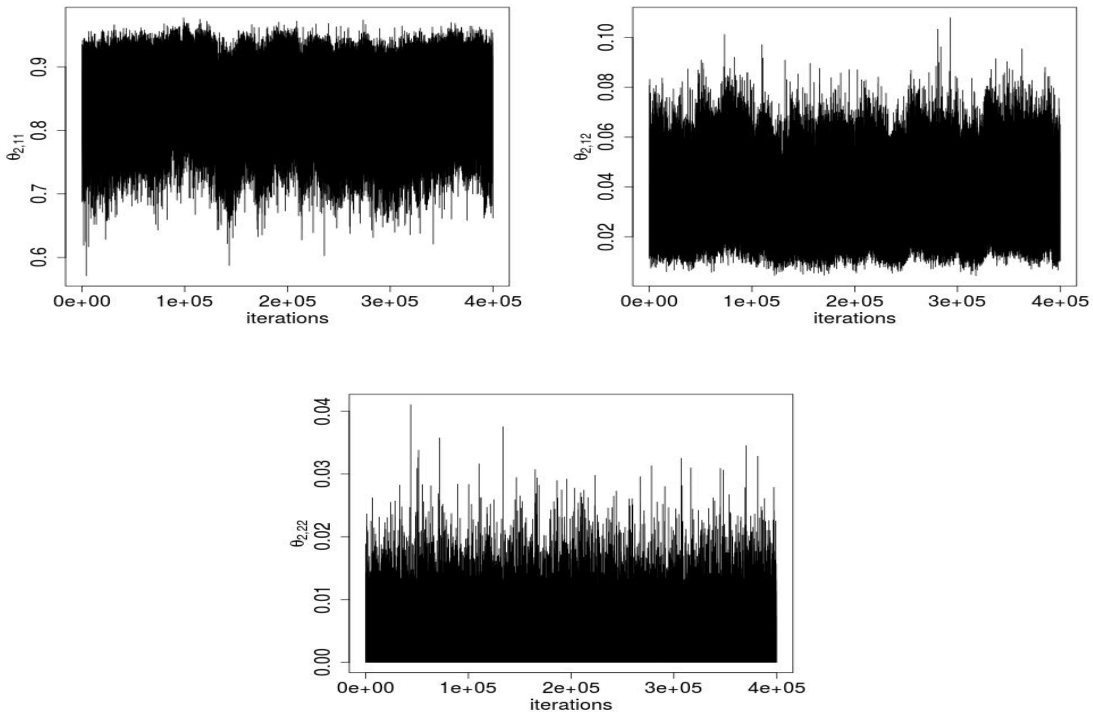

Figure A.7: Trace plots for block specific probabilities of observing an edge $\boldsymbol{\theta_2}$ for representative of cluster $c = 2$, for 400,000 iterations after a burn-in of 100,000.

In Figures A.8 and A.9, we present the results for $\boldsymbol{\theta_c}$ for representatives of cluster

$c = 1$, and 2, for clustering algorithm applied on the denser network data set and $C = 2$.

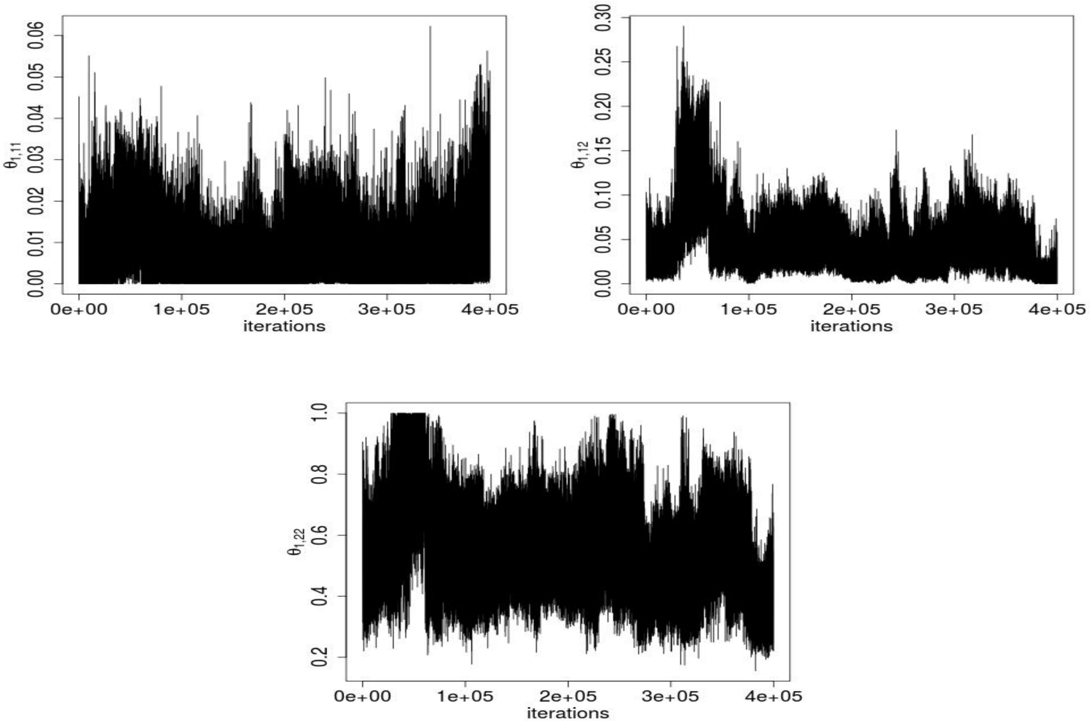

Figure A.8: Trace plots for block specific probabilities of observing an edge $\boldsymbol{\theta_1}$ for representative of cluster $c = 1$, for 400,000 iterations after a burn-in of 100,000.

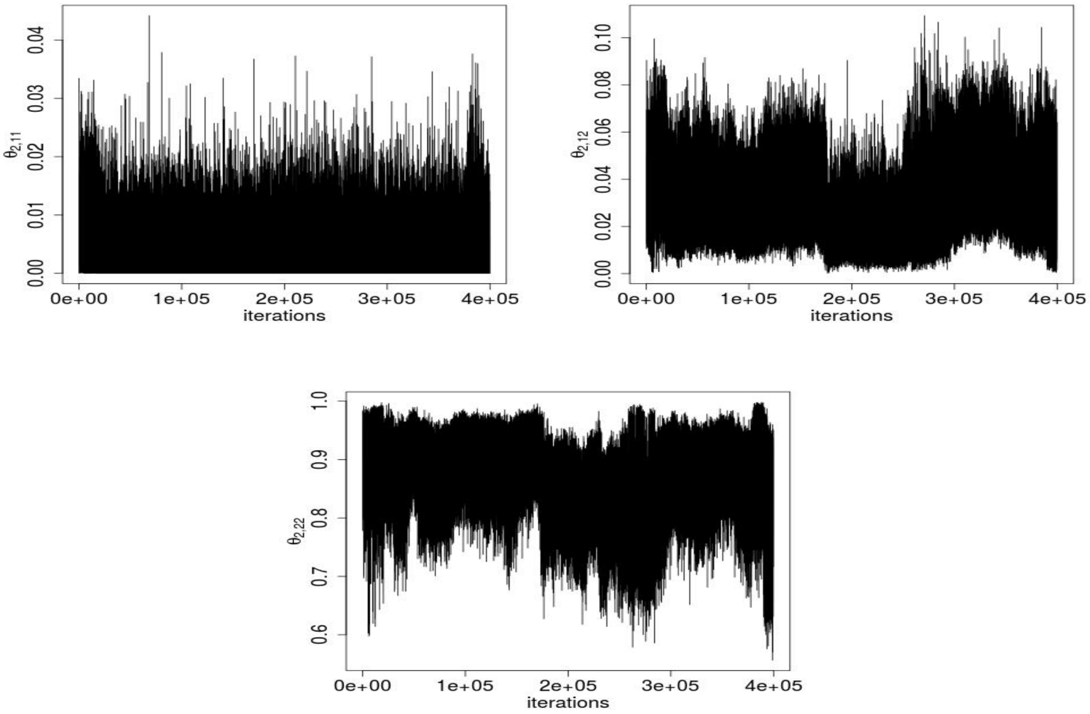

Figure A.9: Trace plots for block specific probabilities of observing an edge $\boldsymbol{\theta_2}$ for representative of cluster $c = 2$, for 400,000 iterations after a burn-in of 100,000.

The stationarity issue of the chains for $\boldsymbol{\theta}$ can be explained by the small posterior mass (6% and 8%) that the posterior modes for the representatives concentrate, affecting the inference of the SBM parameters.

In Figure A.10, we present the results for $\boldsymbol{\theta}$ for single network representative after implementing the outlier clustering algorithm on the denser network data.

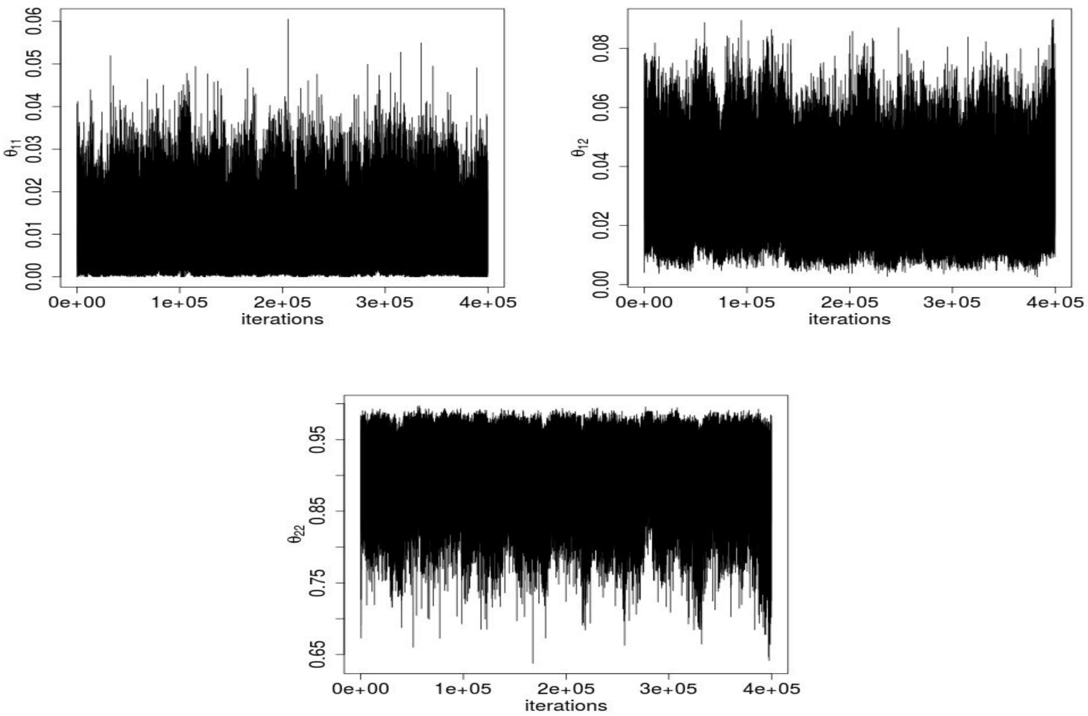

Figure A.10: Trace plots for block specific probabilities of observing an edge $\boldsymbol{\theta}$ for single network representative, for 400,000 iterations after a burn-in of 100,000.

In Figures A.11, A.12, A.13 and A.14, we present the results for the outlier cluster model with the Erdös-Rényi model assumed for the sole representative of the population.

In Figures A.15, A.16, A.17, A.18 and A.19 we present the results for $C = 2$ with two representatives with the Erdös-Rényi model assumed.

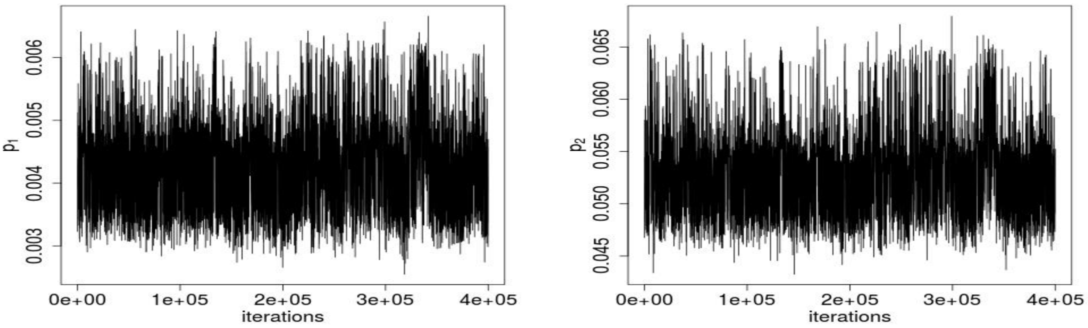

Figure A.11: Trace plots for false positive probabilities p_c for majority cluster labelled by $c = 1$ (left) and outlier cluster labelled by $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000, for the Erdös-Rényi model specification.

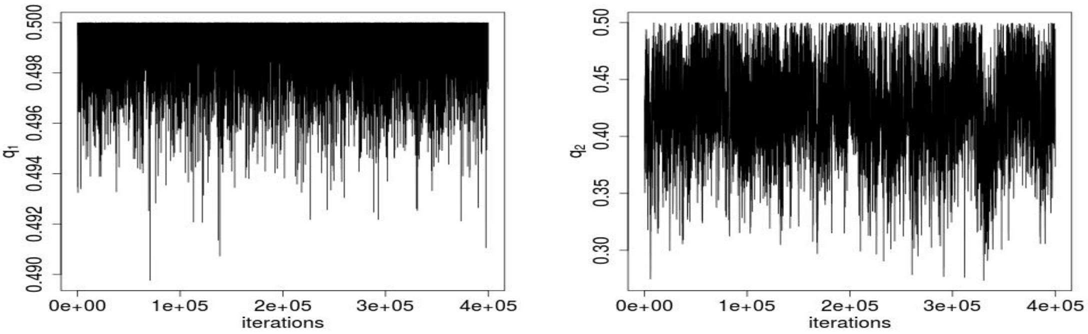

Figure A.12: Trace plots for false negative probabilities q_c for majority cluster labelled by $c = 1$ (left) and outlier cluster labelled by $c = 2$ (right), for 400,000 iterations after a burn-in of 100,000, for the Erdös-Rényi model specification.

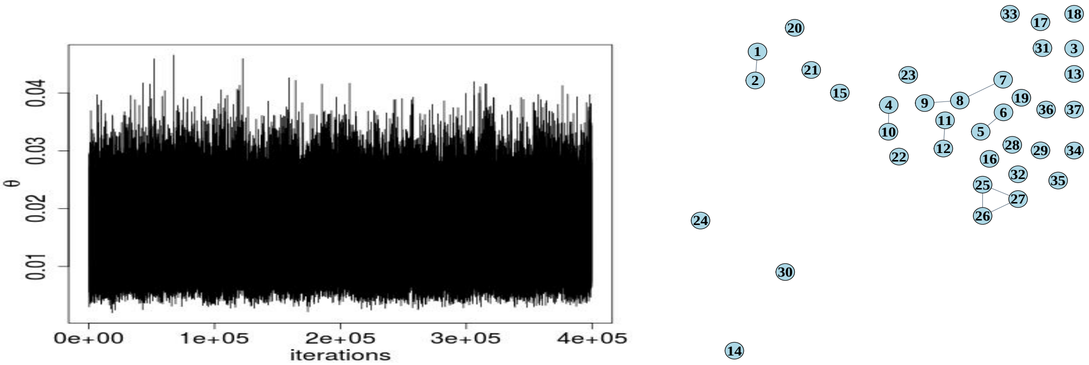

Figure A.13: Left: Trace plot for probability of an edge θ for network representative, for 400,000 iterations after a burn-in of 100,000, for the Erdös-Rényi model specification. Right: Posterior mode of representative network with posterior mass 52%, for last 100,000 iterations, for the Erdös-Rényi model specification.

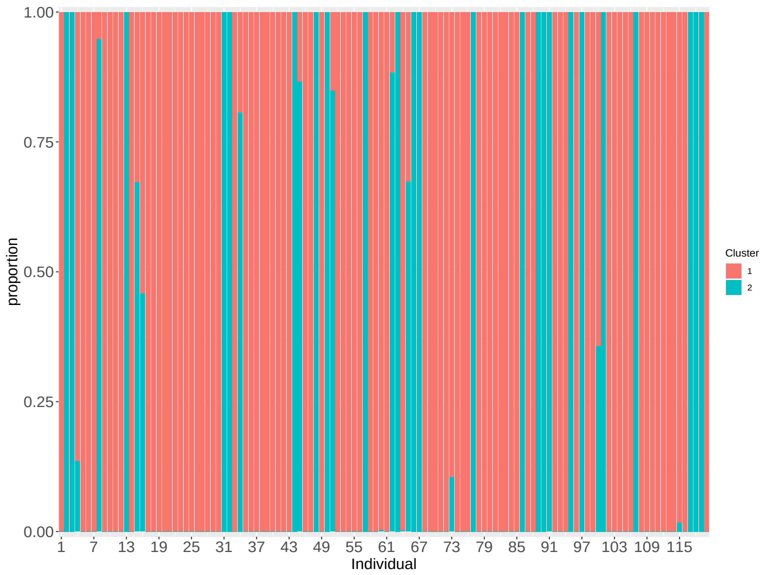

Figure A.14: Proportion of times (y axis) an individual (x axis) is allocated to each of the 2 clusters, after a burn-in of 100,000 iterations, for the Erdös-Rényi model specification.

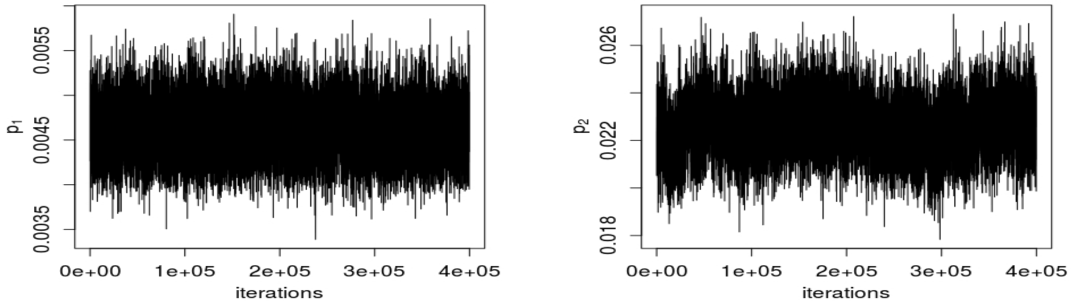

Figure A.15: Trace plots for false positive probabilities p_c for clusters $c = 1$, and 2, for 400,000 iterations after a burn-in of 100,000, for the Erdös-Rényi model specification.

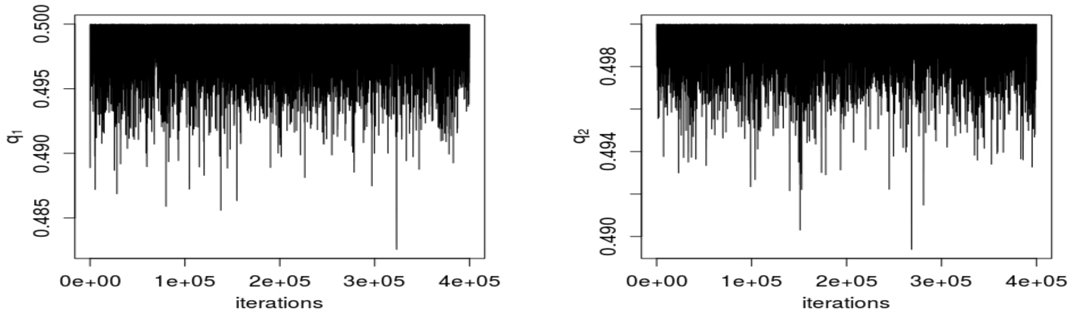

Figure A.16: Trace plots for false negative probabilities q_c for clusters $c = 1$, and 2, for 400,000 iterations after a burn-in of 100,000, for the Erdös-Rényi model specification.

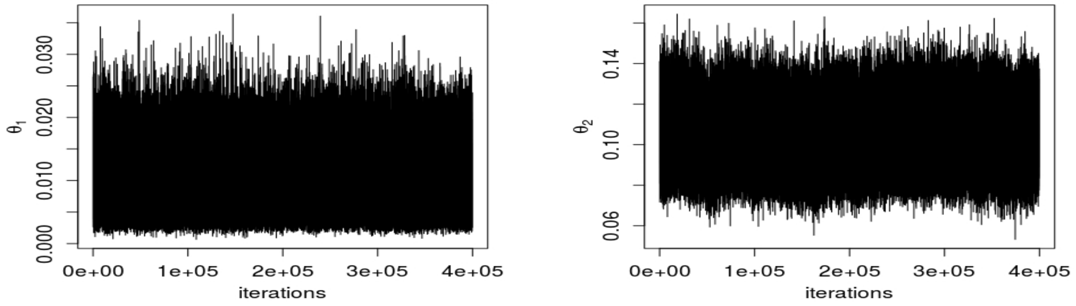

Figure A.17: Left: Trace plot for probability of an edge θ_1 for network representative of cluster $c = 1$, for 400,000 iterations after a burn-in of 100,000, for the Erdös-Rényi model specification. Right: Trace plot for probability of an edge θ_2 for network representative of cluster $c = 2$, for 400,000 iterations after a burn-in of 100,000, for the Erdös-Rényi model specification.

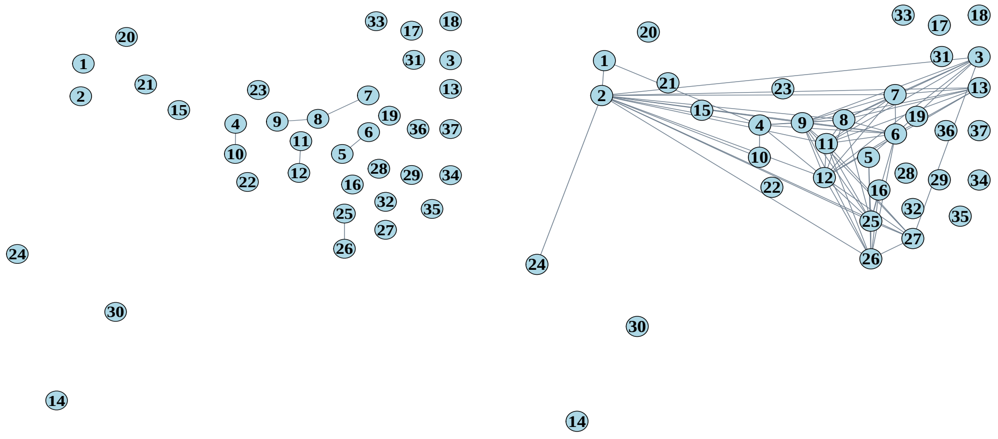

Figure A.18: Left: Posterior mode of representative network for cluster $c = 1$, with posterior mass 100%, for last 100,000 iterations, for the Erdös-Rényi model specification. Right: Posterior mode of representative network for cluster $c = 2$, with posterior mass 17% for last 100,000 iterations, for the Erdös-Rényi model specification.

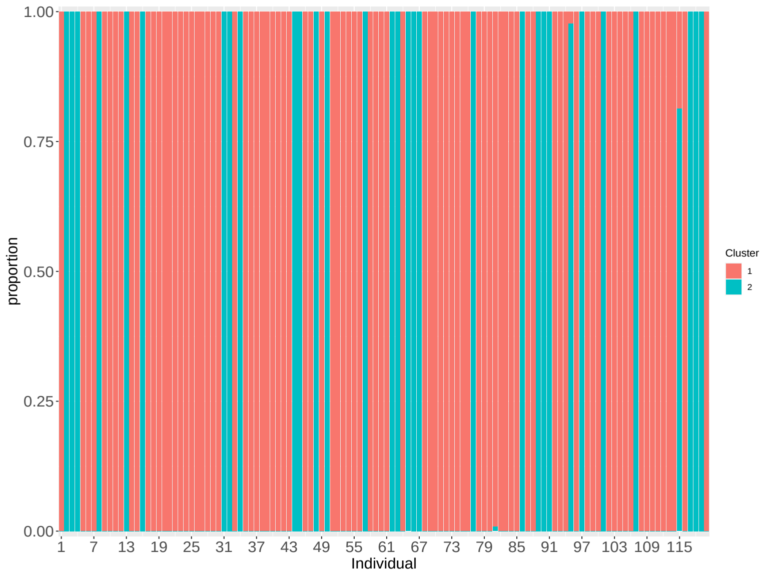

Figure A.19: Proportion of times (y axis) an individual (x axis) is allocated to each of the 2 clusters, after a burn-in of 100,000 iterations, for the Erdös-Rényi model specification.

A.4 MCMC algorithm

Below, we present details of how the MCMC algorithm is implemented to make posterior inferences from the proposed model.

```
Algorithm 1: MCMC Algorithm for Clustering Network Populations
    input : $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_N; C, K, M, w_0, \theta_0, \alpha_0, \beta_0, \gamma_0, \delta_0, \epsilon_0, \zeta_0, \psi, \chi$
    output: Posterior distributions of $A_{\mathcal{G}_1^*}, \dots, A_{\mathcal{G}_C^*}$, $p_1, \dots, p_C$, $q_1, \dots, q_C$, $\tau_1, \dots, \tau_C$,
            $z_1, \dots, z_N$, $\boldsymbol{\theta_1}, \dots, \boldsymbol{\theta_C}$, $\boldsymbol{w_1}, \dots, \boldsymbol{w_C}$, $\boldsymbol{b_1}, \dots, \boldsymbol{b_C}$
    Initialisation: randomly generate $A_{\mathcal{G}_1^*}^{(0)}, \dots, A_{\mathcal{G}_C^*}^{(0)}$, $p_1^{(0)}, \dots, p_C^{(0)}$, $q_1^{(0)}, \dots, q_C^{(0)}$,
    $\tau_1^{(0)}, \dots, \tau_C^{(0)}$, $z_1^{(0)}, \dots, z_N^{(0)}$, $\boldsymbol{\theta_1}^{(0)}, \dots, \boldsymbol{\theta_C}^{(0)}$, $\boldsymbol{w_1}^{(0)}, \dots, \boldsymbol{w_C}^{(0)}$, $\boldsymbol{b_1}^{(0)}, \dots, \boldsymbol{b_C}^{(0)}$
    for $i \leftarrow 1$ to $M$ do
        Gibbs step: Update $\tau_1, \dots, \tau_C$
        compute: $\eta_c = \sum_{j=1}^{N} \mathbb{1}_c(z_j^{(i-1)})$ for $c = 1, \dots, C$
        sample: $\tau_1^{(i)}, \dots, \tau_C^{(i)} \sim Dir(\psi + \eta_1, \dots, \psi + \eta_C)$
        for $c \leftarrow 1$ to $C$ do
            MH step with a mixture of kernels: Update $A_{\mathcal{G}_c^*}$ or $p_c$ or $q_c$
            sample: $v \sim \text{Multinomial}(\xi_1, \dots, \xi_L)$
            Depending on the value of $v$, update either $A_{\mathcal{G}_c^*}$ or $p_c$ or $q_c$ as per the
            Measurement Error model with SBM structure, where the sum in likelihood is
            over the networks $\{j : z_j^{(i-1)} = c\}$
            Gibbs step: Update $\boldsymbol{w_c}$
            compute: $h_k = \sum_{j=1}^{n} \mathbb{1}_k(b_j^{(i-1)})$
            sample: $\boldsymbol{w_c}^{(i)} \sim Dir(\chi + h_1, \dots, \chi + h_K)$
            Gibbs step: Update $\boldsymbol{\theta_c}$
            compute: $A[st] = \sum_{(u,v): b_u = s, b_v = t} A_{\mathcal{G}_c^*}^{(i)}(u, v)$ and
            $n_{st} = \sum_{(u,v): u \neq v} \mathbb{I}(b_u = s, b_v = t)$ for $s, t \in \{1, \dots, K\}$
            sample: $\theta_{c,st}^{(i)} \sim \text{Beta}(A[st] + \epsilon_0, \zeta_0 + n_{st} - A[st])$
            Gibbs step: Update $\boldsymbol{b_c}$
            for $j \leftarrow 1$ to $n$ do
                compute: $p_{kj} = w_{c,k}^{(i)} \cdot \prod_{m=1}^{n} \theta_{kb_{c,m}^{(i-1)}}^{(i)A(j,m)} (1 - \theta_{kb_{c,m}^{(i-1)}}^{(i)})^{1-A(j,m)}$ for $k = 1, \dots, K$
                sample: $b_{c,j}^{(i)} \sim \text{Multin}(p_{lj}, \dots, p_{Kj})$
            end
        end
        Gibbs step: Update $z_1, \dots, z_N$
        for $j \leftarrow 1$ to $N$ do
            compute: $p_{cj} = \tau_c^{(i)} \cdot \prod_{(u,v): u<v} \left( (1 - q_c^{(i)})^{A_{\mathcal{G}_j}(u,v)} q_c^{(i)(1 - A_{\mathcal{G}_j}(u,v))} \right)^{A_{\mathcal{G}_c^*}^{(i)}(u,v)} \cdot$
            $\left( p_c^{(i)A_{\mathcal{G}_j}(u,v)} (1 - p_c^{(i)})^{1 - A_{\mathcal{G}_j}(u,v)} \right)^{1 - A_{\mathcal{G}_c^*}^{(i)}(u,v)}$ for $c = 1, \dots, C$
            sample: $z_j^{(i)} \sim \text{Multin}(p_{1j}, \dots, p_{Cj})$
        end
    end
```

Appendix B

Appendix for 'Bayesian Inference for Models in the Spherical Network Family (SNF) using Importance Sampling (IS)'

B.1 Mixing issue with Auxiliary Variable method for the SNF model

In this Section we demonstrate the MCMC chain mixing issue that arises with the implementation of the Auxiliary Variable method (Møller et al. [2006]) for the SNF model (Lunagómez et al. [2021]), through the use of a specific simulated data example. We further demonstrate the improvement in the chain mixing after the implementation of our proposed MCMC scheme with IS step, for the same simulated data example.

To illustrate the mixing issue, we first simulate a population of $N = 50$ networks, with $n = 5$ nodes from the SNF model with parameters $\gamma = 7$ and centroid $\mathcal{G}^m$ as seen in Figure B.1.

We apply the MCMC scheme with Auxiliary Variable method proposed in Lunagómez et al. [2021] on the simulated network data to make inferences for the SNF model parameters. Figure B.2 shows the traceplot for the dispersion parameter γ, after running the MCMC for 10,000 iterations with no burn-in.

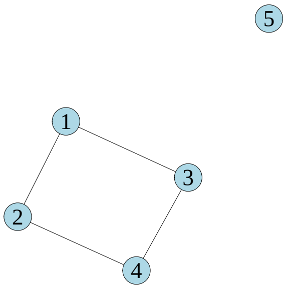

Figure B.1: 5-node centroid $\mathcal{G}^m$ generated for simulated data example.

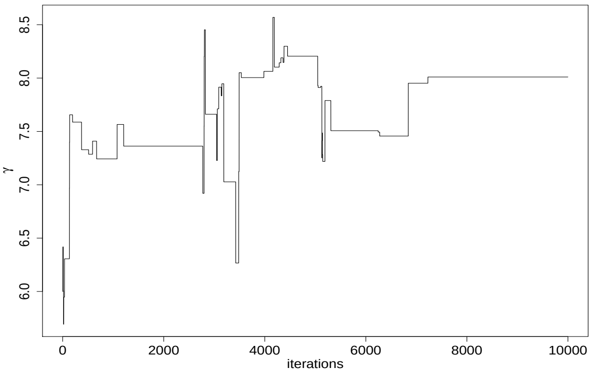

Figure B.2: Traceplot of γ for 10,000 iterations of the MCMC with Auxiliary Variable method applied on the simulated population of 5-node networks.

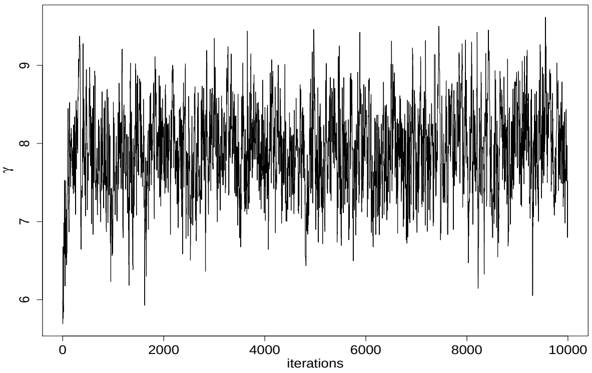

Figure B.3: Traceplot of γ for 10,000 iterations of the MCMC with IS step applied on the simulated population of 5-node networks.

To facilitate comparisons, we apply our MCMC scheme with IS step on the same simulated network data, and present the traceplot for the dispersion γ for 10,000 iterations of the MCMC and no burn-in in Figure B.3. From the figures we notice a significant improvement in the mixing of the MCMC chain for γ, under our proposed MCMC scheme with IS step.

B.2 Proof of HS distance metric

We now show that the HS dissimilarity measure proposed in Section 4.3.1 is a distance metric. The HS measure is the weighted sum of the Hamming distance and the symmetric difference of cycles between two graphs. The Hamming distance is a well-known distance metric, thus, to prove that the HS measure is also a distance metric, we need to prove that the symmetric difference between graphs' cycles is a distance metric.

Let $\mathcal{C}_n$ be the set of cycles for graphs of size n, and each $C_{\mathcal{G}_i}, C_{\mathcal{G}_j}, C_{\mathcal{G}_k} \in \mathcal{C}_n$ be the subset of cycles found in graphs $\mathcal{G}_i$, $\mathcal{G}_j$ and $\mathcal{G}_k$ respectively. Thence, the symmetric difference of the cycles of two graphs is $d_{symm} = |C_{\mathcal{G}_\cdot} \Delta C_{\mathcal{G}_\cdot}|$. The function $d_{symm} : \mathcal{C}_n \times \mathcal{C}_n \rightarrow [0, \infty)$ is a distance metric if the following conditions are satisfied:

1. $d_{symm}(C_{\mathcal{G}_i}, C_{\mathcal{G}_j}) = 0 \Leftrightarrow C_{\mathcal{G}_i} = C_{\mathcal{G}_j}$
2. $d_{symm}(C_{\mathcal{G}_i}, C_{\mathcal{G}_j}) = d_{symm}(C_{\mathcal{G}_j}, C_{\mathcal{G}_i})$
3. $d_{symm}(C_{\mathcal{G}_i}, C_{\mathcal{G}_j}) \leq d_{symm}(C_{\mathcal{G}_i}, C_{\mathcal{G}_k}) + d_{symm}(C_{\mathcal{G}_k}, C_{\mathcal{G}_j})$

Conditions 1 and 2 are clearly satisfied. Thus, we need to prove that the triangle inequality holds for the symmetric difference of cycles. The symmetric difference has the following property,

$$C_{\mathcal{G}_i} \Delta C_{\mathcal{G}_j} = (C_{\mathcal{G}_i} \Delta C_{\mathcal{G}_k}) \Delta (C_{\mathcal{G}_k} \Delta C_{\mathcal{G}_j}).$$

It follows that

$$\begin{gathered} C_{\mathcal{G}_i} \Delta C_{\mathcal{G}_j} \subseteq (C_{\mathcal{G}_i} \Delta C_{\mathcal{G}_k}) \cup (C_{\mathcal{G}_k} \Delta C_{\mathcal{G}_j}) \Rightarrow \\ |C_{\mathcal{G}_i} \Delta C_{\mathcal{G}_j}| \leq |C_{\mathcal{G}_i} \Delta C_{\mathcal{G}_k}| + |C_{\mathcal{G}_k} \Delta C_{\mathcal{G}_j}|. \end{gathered}$$

Thus condition 3 is satisfied for the symmetric difference of cycles between graphs.

B.3 Challenges with cycle detection

We now present the centrality-betweenness distance metric considered as covariate for the xgboost algorithm, to predict the HS distance between graphs in each iteration of the MCMC.

As discussed in Donnat and Holmes [2018], the centrality-betweenness dissimilarity measure for two graphs $\mathcal{G}$ and $\tilde{\mathcal{G}}$, quantifies changes between graphs with respect to the betweenness of the graphs' nodes, i.e. the centrality of the nodes in each graph. Specifically, the centrality-betweenness of node i denoted by c_i, measures the number of shortest paths going through node i. Thus, the centrality-betweenness distance between graphs $\mathcal{G}$ and $\tilde{\mathcal{G}}$ is

$$d(\mathcal{G}, \tilde{\mathcal{G}}) = \sqrt{\sum_{i=1}^{n} (c_i^{\mathcal{G}} - c_i^{\tilde{\mathcal{G}}})^2}.$$

B.4 Simulation study: Performance of MCMC for small network sizes

B.4.1 Simulation study for HS distance

In Figures B.4, B.5, B.6, B.7 we present the autocorrelation plots for γ under the simulation regimes $\gamma = \{0.01, 0.6, 1.1, 1.6\}$ and $N = 50$, for both IS sample sizes $K = \{2000, 4000\}$ and the two distinct IS schemes (adaptive and non-adaptive scheme), as discussed in Section 4.4.2.

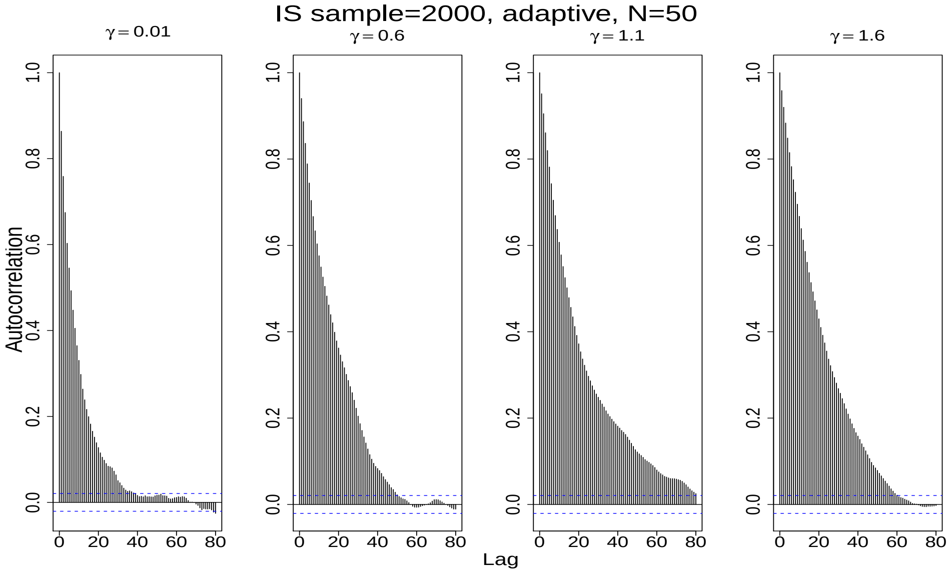

Figure B.4: Autocorrelation plots for γ under the simulation regimes $\gamma = \{0.01, 0.6, 1.1, 1.6\}$ and $N = 50$, for IS sample size $K = 2,000$, for the adaptive MCMC scheme.

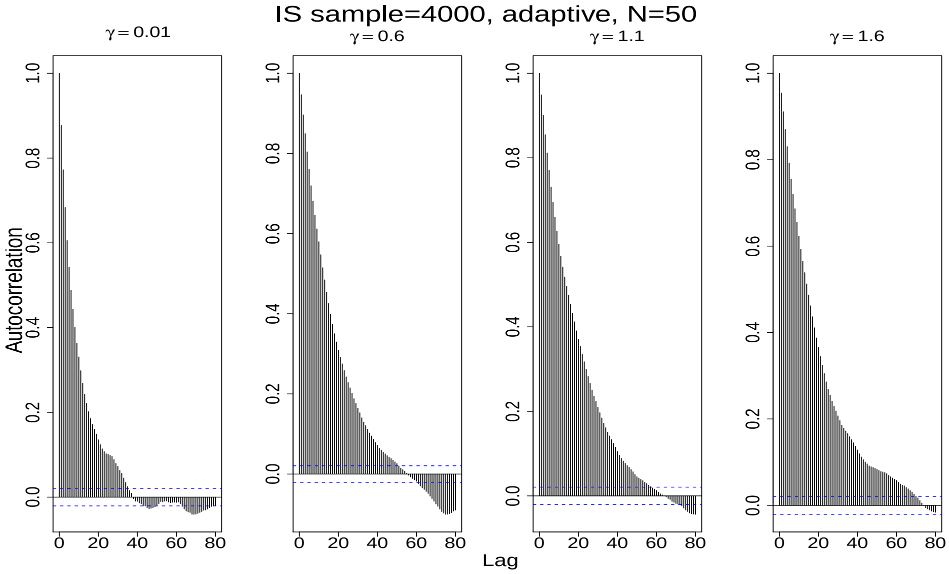

Figure B.5: Autocorrelation plots for γ under the simulation regimes $\gamma = \{0.01, 0.6, 1.1, 1.6\}$ and $N = 50$, for IS sample size $K = 4,000$, for the adaptive MCMC scheme.

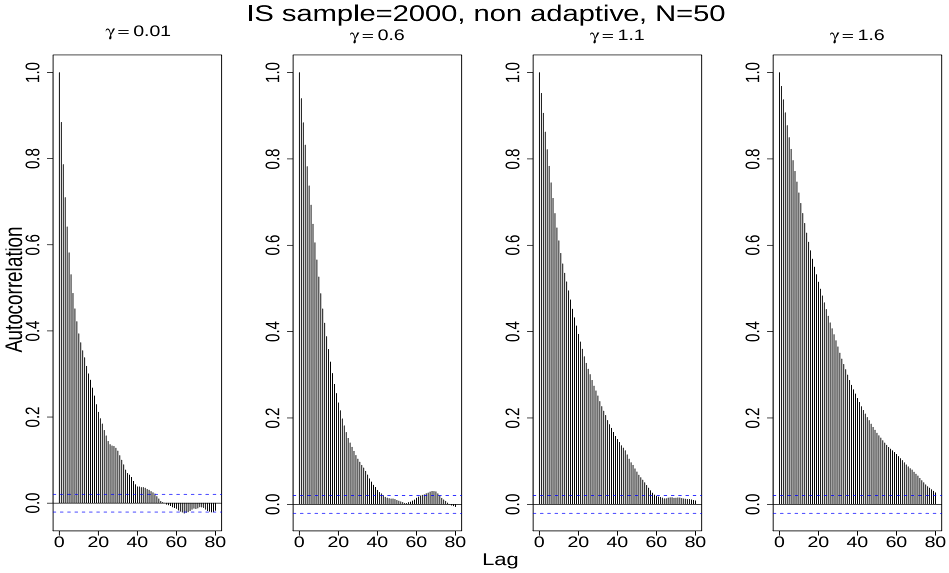

Figure B.6: Autocorrelation plots for γ under the simulation regimes $\gamma = \{0.01, 0.6, 1.1, 1.6\}$ and $N = 50$, for IS sample size $K = 2,000$, for the non-adaptive MCMC scheme.

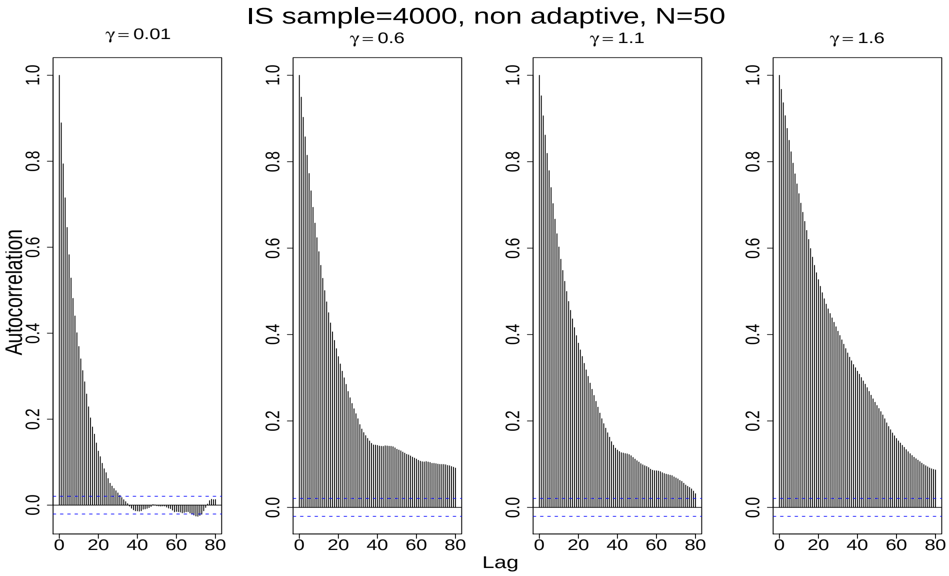

Figure B.7: Autocorrelation plots for γ under the simulation regimes $\gamma = \{0.01, 0.6, 1.1, 1.6\}$ and $N = 50$, for IS sample size $K = 4,000$, for the non-adaptive MCMC scheme.

B.4.2 Simulation study for Jaccard distance

In Figure B.8, we present the simulation results under the specification of a Uniform distribution over the interval $[0.01, 18]$ for the dispersion parameter γ, and for $N = 50$ indicatively, as discussed in Section 4.2.2.

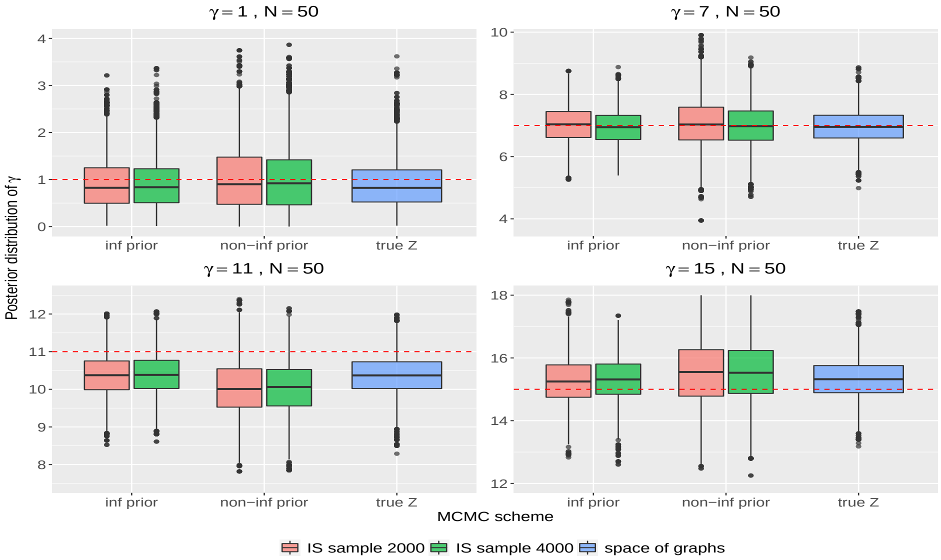

Figure B.8: Posterior distribution of γ, for simulation regimes $\gamma = \{1, 7, 11, 15\}$, $N = 50$ and IS sample $K = \{2000, 4000\}$, for MCMC algorithm with informative and non-informative prior for γ, and MCMC with exact calculation of Z.

B.5 Simulation study: Performance of MCMC for moderate network sizes

B.5.1 Simulation study for Jaccard distance

In Figure B.9, we present the simulation results of our MCMC for a greater size of $K = 18,000$ networks, for the simulation regime where $\gamma = 90$ and $N = 20$, discussed in Section 4.4.3.

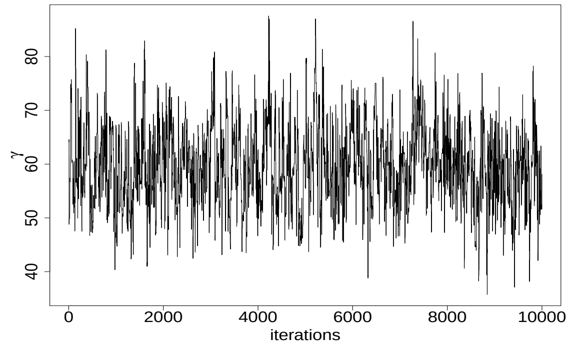

Figure B.9: Traceplot of γ for the simulation regime $\gamma = 90$ and $N = 20$ and IS sample size $K = 18,000$.

B.5.2 Simulation study for HS distance

In Figures B.10 and B.11, we present the traceplots for γ for IS density specification the mixture of CER models with $J = 2$ mixture components with probabilities $\beta = (1/2, 1/2)$ corresponding to two dispersion parameters $\tilde{\alpha} = (0.02, 0.07)$, discussed in Section 4.4.3. We run the MCMC for 5,000 iterations, implementing the xgboost algorithm.

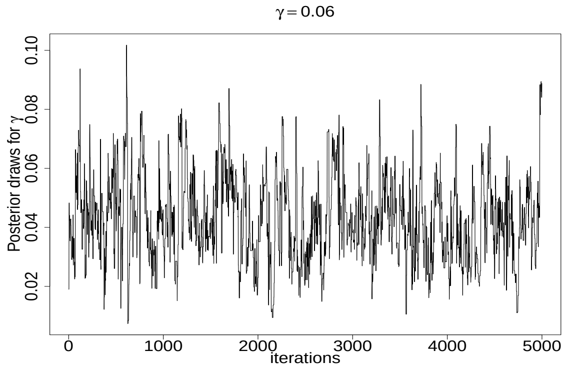

Figure B.10: Traceplots for γ, for simulation regime $\gamma = 0.06$, using a mixture of CER as the IS density.

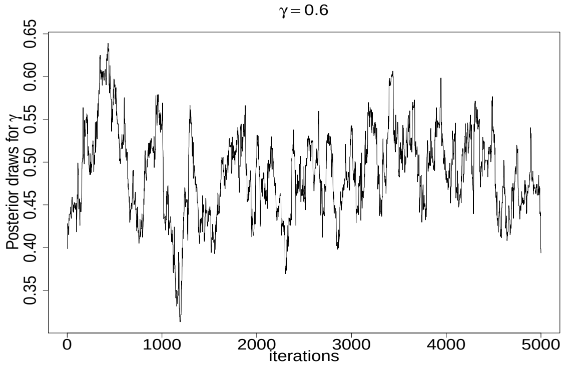

Figure B.11: Traceplots for γ, for simulation regime $\gamma = 0.6$, using a mixture of CER as the IS density.

The posterior mode of the representative for $\gamma = 0.06$ concentrates a posterior mass 0.039, and its Hamming distance from the true centroid is 41 edges. The posterior mode of the representative for $\gamma = 0.6$ concentrates a posterior mass 0.0204, and its Hamming distance from the true centroid is 35 edges.

Bibliography

C. C. Aggarwal, Y. Zhao, and S. Y. Philip. Outlier detection in graph streams. In *2011 IEEE 27th International Conference on Data Engineering*, pages 399–409. IEEE, 2011.

E. M. Airoldi and A. W. Blocker. Estimating latent processes on a network from indirect measurements. *Journal of the American Statistical Association*, 108(501):149–164, 2013.

P. Alquier, N. Friel, R. Everitt, and A. Boland. Noisy monte carlo: Convergence of markov chains with approximate transition kernels. *Statistics and Computing*, 26 (1-2):29–47, 2016.

V. Arora and M. Ventresca. Action-based modeling of complex networks. *Scientific reports*, 7(1):1–10, 2017.

V. Arora, D. Guo, K. D. Dunbar, and M. Ventresca. Examining the variability in network populations and its role in generative models. *Network Science*, 8(S1):S43–S64, 2020.

J. Arroyo, A. Athreya, J. Cape, G. Chen, C. E. Priebe, and J. T. Vogelstein. Inference for multiple heterogeneous networks with a common invariant subspace. *arXiv preprint arXiv:1906.10026*, 2019.

M. Avella-Medina, F. Parise, M. T. Schaub, and S. Segarra. Centrality measures for graphons: Accounting for uncertainty in networks. *IEEE Transactions on Network Science and Engineering*, 7(1):520–537, 2018.

P. Balachandran, E. D. Kolaczyk, and W. D. Viles. On the propagation of low-rate measurement error to subgraph counts in large networks. *The Journal of Machine Learning Research*, 18(1):2025–2057, 2017.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286 (5439):509–512, 1999.

G. Biau. Analysis of a random forests model. *The Journal of Machine Learning Research*, 13:1063–1095, 2012.

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

C. Borgs, J. T. Chayes, L. Lovász, V. T. Sós, and K. Vesztergombi. Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, 219(6):1801–1851, 2008.

C. R. Buchanan, M. E. Bastin, S. J. Ritchie, D. C. Liewald, J. W. Madole, E. M. Tucker-Drob, I. J. Deary, and S. R. Cox. The effect of network thresholding and weighting on structural brain networks in the uk biobank. *NeuroImage*, 211:116443, 2020.

W. Budiaji. *kmed: Distance-Based K-Medoids*, 2019. URL `https://CRAN.R-project.org/package=kmed`. R package version 0.3.0.

C. M. Carvalho, N. G. Polson, and J. G. Scott. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80. PMLR, 2009.

J. Chang, E. D. Kolaczyk, and Q. Yao. Estimation of subgraph densities in noisy networks. *Journal of the American Statistical Association*, pages 1–14, 2020.

S. Chatterjee et al. Matrix estimation by universal singular value thresholding. *Annals of Statistics*, 43(1):177–214, 2015.

P. Chauhan and M. Sood. Big data: Present and future. *Computer*, 54(04):59–65, 2021.

M.-H. Chen and Q.-M. Shao. On monte carlo methods for estimating ratios of normalizing constants. *The Annals of Statistics*, 25(4):1563–1594, 1997.

S. Chen, Y. Xing, J. Kang, P. Kochunov, and L. E. Hong. Bayesian modeling of dependence in brain connectivity data. *Biostatistics*, 21(2):269–286, 2020.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, and Y. Li. *xgboost: Extreme Gradient Boosting*, 2021. URL `https://CRAN.R-project.org/package=xgboost`. R package version 1.3.2.1.

F. Chung and L. Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002.

M. K. Chung. Statistical challenges of big brain network data. *Statistics & probability letters*, 136:78–82, 2018.

R. C. Craddock, G. A. James, P. E. Holtzheimer III, X. P. Hu, and H. S. Mayberg. A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human brain mapping*, 33(8):1914–1928, 2012.

G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `https://igraph.org`.

J. Diquigiovanni and B. Scarpa. Analysis of association football playing styles: An innovative method to cluster networks. *Statistical Modelling*, 19(1):28–54, 2019.

C. Donnat and S. Holmes. Tracking network dynamics: A survey of distances and similarity metrics. *arXiv preprint arXiv:1801.07351*, 2018.

D. Durante, D. B. Dunson, and J. T. Vogelstein. Nonparametric bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112(520): 1516–1530, 2017.

J. Eldridge, M. Belkin, and Y. Wang. Graphons, mergeons, and so on! In *Advances in Neural Information Processing Systems*, pages 2307–2315, 2016.

P. Erdös and A. Rényi. On random graphs, i. page 6:290–297, 1959.

J. A. Espinosa, S. Kaisler, F. Armour, and W. Money. Big data redux: New issues and challenges moving forward. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

R. G. Everitt, A. M. Johansen, E. Rowing, and M. Evdemon-Hogan. Bayesian model comparison with un-normalised likelihoods. *Statistics and Computing*, 27(2):403–422, 2017a.

R. G. Everitt, D. Prangle, P. Maybank, and M. Bell. Marginal sequential monte carlo for doubly intractable models. *arXiv preprint arXiv:1710.04382*, 2017b.

T. Fan, L. Lü, and D. Shi. Towards the cycle structures in complex network: A new perspective. *arXiv preprint arXiv:1903.01397*, 2019.

S. Fields and O.-k. Song. A novel genetic system to detect protein–protein interactions. *Nature*, 340(6230):245–246, 1989.

J. Friedman, T. Hastie, R. Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

J. O. Garcia, A. Ashourvan, S. Muldoon, J. M. Vettel, and D. S. Bassett. Applications of community detection techniques to brain graphs: Algorithmic considerations and implications for neural function. *Proceedings of the IEEE*, 106(5):846–867, 2018.

A. Gelman and X. Meng. Path sampling for computing normalizing constants: identities and theory. *University of Chicago Department of Statistics Technical Report*, (377), 1994.

E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

C. E. Ginestet, J. Li, P. Balachandran, S. Rosenberg, E. D. Kolaczyk, et al. Hypothesis testing for network data in functional neuroimaging. *The Annals of Applied Statistics*, 11(2):725–750, 2017.

A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. 2010.

I. Gollini and T. B. Murphy. Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25(1):246–265, 2016.

S. M. Goodreau, J. A. Kitts, and M. Morris. Birds of a feather, or friend of a friend? using exponential random graph models to investigate adolescent social networks. *Demography*, 46(1):103–125, 2009.

P. K. Gopalan and D. M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.

D. Han, J. Li, H. Wang, X. Su, J. Hou, Y. Gu, C. Qian, Y. Lin, X. Liu, M. Huang, et al. Circular rna circmto1 acts as the sponge of microrna-9 to suppress hepatocellular carcinoma progression. *Hepatology*, 66(4):1151–1164, 2017.

D. J. Hand. Statistical analysis of network data: Methods and models by eric d. kolaczyk. 2010.

M. S. Handcock and K. J. Gile. Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1):5, 2010.

N. A. Heard, D. J. Weston, K. Platanioti, and D. J. Hand. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, pages 645–662, 2010.

D. F. Heitjan and D. B. Rubin. Inference from coarse data via multiple imputation with application to age heaping. *Journal of the American Statistical Association*, 85(410): 304–314, 1990.

D. F. Heitjan and D. B. Rubin. Ignorability and coarse data. *The annals of statistics*, pages 2244–2253, 1991.

P. D. Hoff. *Random effects models for network data.* na, 2003.

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

T. P. Hughes, J. T. Kerry, M. Álvarez-Noriega, J. G. Álvarez-Romero, K. D. Anderson, A. H. Baird, R. C. Babcock, M. Beger, D. R. Bellwood, R. Berkelmans, et al. Global warming and recurrent mass bleaching of corals. *Nature*, 543(7645):373–377, 2017.

INRA and J.-B. Leger. *blockmodels: Latent and Stochastic Block Model Estimation by a 'V-EM' Algorithm*, 2015. URL `https://CRAN.R-project.org/package=blockmodels`. R package version 1.1.1.

E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925.

X. Jiang, D. Gold, and E. D. Kolaczyk. Network-based auto-probit modeling for protein function prediction. *Biometrics*, 67(3):958–966, 2011.

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

S. A. Keith, A. H. Baird, J.-P. A. Hobbs, E. S. Woolsey, A. S. Hoey, N. Fadli, and N. J. Sanders. Synchronous behavioural shifts in reef fishes linked to mass coral bleaching. *Nature Climate Change*, 8(11):986–991, 2018.

J. K. Kim and M. Hong. Imputation for statistical inference with coarse data. *Canadian Journal of Statistics*, 40(3):604–618, 2012.

E. Kolaczyk, L. Lin, S. Rosenberg, and J. Walters. Averages of unlabeled networks: Geometric characterization and asymptotic behavior. *arXiv preprint arXiv:1709.02793*, 2017.

J. H. Koskinen, G. L. Robins, P. Wang, and P. E. Pattison. Bayesian analysis for partially observed network data, missing ties, attributes and actors. *Social Networks*, 35(4):514–527, 2013.

M. Koutrouli, E. Karatzas, D. Paez-Espino, and G. A. Pavlopoulos. A guide to conquer the biological network era using graph theory. *Frontiers in bioengineering and biotechnology*, 8:34, 2020.

C. M. Le and T. Li. Linear regression and its inference on noisy network-linked data. *arXiv preprint arXiv:2007.00803*, 2020.

C. M. Le, K. Levin, E. Levina, et al. Estimating a network from multiple noisy realizations. *Electronic Journal of Statistics*, 12(2):4697–4740, 2018.

W. Lee, T. H. McCormick, J. Neil, C. Sodja, and Y. Cui. Anomaly detection in large scale networks with latent space models. *arXiv preprint arXiv:1911.05522*, 2019.

M. Levandowsky and D. Winter. Distance between sets. *Nature*, 234(5323):34–35, 1971.

W. Li, D. L. Sussman, and E. D. Kolaczyk. Causal inference under network interference with noise. *arXiv preprint arXiv:2105.04518*, 2021.

L. Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

S. Lunagómez, S. C. Olhede, and P. J. Wolfe. Modeling network populations via graph distances. *Journal of the American Statistical Association*, 116(536):2023–2040, 2021.

M.-E. Lynall, D. S. Bassett, R. Kerwin, P. J. McKenna, M. Kitzbichler, U. Muller, and E. Bullmore. Functional connectivity and brain networks in schizophrenia. *Journal of Neuroscience*, 30(28):9477–9487, 2010.

C. L. Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957.

D. J. Marchette and E. L. Hohman. Utilizing covariates in partially observed networks. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 166–172. IEEE, 2015.

P.-A. G. Maugis, S. C. Olhede, and P. J. Wolfe. Topology reveals universal features for network comparison. *arXiv preprint arXiv:1705.05677*, 2017.

T. McClanahan, E. Weil, J. Cortés, A. Baird, and M. Ateweberhan. Consequences of coral bleaching for sessile reef organisms. In *Coral bleaching*, pages 121–138. Springer, 2009.

F. J. Medina-Aguayo, A. Lee, and G. O. Roberts. Stability of noisy metropolis–hastings. *Statistics and Computing*, 26(6):1187–1211, 2016.

X.-L. Meng and W. H. Wong. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, pages 831–860, 1996.

F. L. Metz, J. Rocchi, and P. Urbani. Statistical mechanics of the spherical hierarchical model with random fields. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(9):P09018, 2014.

M. H. Mohd. Diversity in interaction strength promotes rich dynamical behaviours in a three-species ecological system. *Applied Mathematics and Computation*, 353:243–253, 2019.

J. Møller, A. N. Pettitt, R. Reeves, and K. K. Berthelsen. An efficient markov chain monte carlo method for distributions with intractable normalising constants. *Biometrika*, 2006.

M. Mørup, M. N. Schmidt, and L. K. Hansen. Infinite multiple membership relational modeling for complex networks. In *2011 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2011.

S. S. Mukherjee, P. Sarkar, and L. Lin. On clustering network-valued data. *Advances in neural information processing systems*, 2017.

I. Murray, Z. Ghahramani, and D. MacKay. Mcmc for doubly-intractable distributions. *arXiv preprint arXiv:1206.6848*, 2012.

N. Muyinda, J. M. Baetens, B. De Baets, and S. Rao. Using intransitive triads to determine final species richness of competition networks. *Physica A: Statistical mechanics and its Applications*, 540:123249, 2020a.

N. Muyinda, B. De Baets, and S. Rao. Non-king elimination, intransitive triad interactions, and species coexistence in ecological competition networks. *Theoretical Ecology*, 13(3):385–397, 2020b.

R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

B. G. Nelson, D. S. Bassett, J. Camchong, E. T. Bullmore, and K. O. Lim. Comparison of large-scale human brain functional and anatomical networks in schizophrenia. *NeuroImage: Clinical*, 15:439–448, 2017.

M. Newman. Network reconstruction and error estimation with noisy network data. *arXiv preprint arXiv:1803.02427*, 2018a.

M. E. Newman. Estimating network structure from unreliable measurements. *Physical Review E*, 98(6):062321, 2018b.

T. L. J. Ng and T. B. Murphy. Generalized random dot product graph. *Statistics & Probability Letters*, 148:143–149, 2019.

C. L. M. Nickel. *Random dot product graphs a model for social networks.* PhD thesis, Johns Hopkins University, 2008.

A. M. Nielsen and D. Witten. The multiple random dot product graph model. *arXiv preprint arXiv:1811.12172*, 2018.

C. Orsini, M. M. Dankulov, P. Colomer-de Simón, A. Jamakovic, P. Mahadevan, A. Vahdat, K. E. Bassler, Z. Toroczkai, M. Boguñá, G. Caldarelli, et al. Quantifying randomness in real networks. *Nature communications*, 6(1):1–10, 2015.

T. P. Peixoto. Reconstructing networks with unknown and heterogeneous errors. *Physical Review X*, 8(4):041011, 2018.

J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar, et al. Functional network organization of the human brain. *Neuron*, 72(4):665–678, 2011.

R. Prajapati and I. A. Emerson. Construction and analysis of brain networks from different neuroimaging techniques. *International Journal of Neuroscience*, pages 1–22, 2020.

G. Prasad, S. H. Joshi, T. M. Nir, A. W. Toga, P. M. Thompson, A. D. N. I. (ADNI, et al. Brain connectivity and novel network measures for alzheimer's disease classification. *Neurobiology of aging*, 36:S121–S131, 2015.

C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park. Scan statistics on enron graphs. *Computational & Mathematical Organization Theory*, 11(3):229–247, 2005.

C. E. Priebe, D. L. Sussman, M. Tang, and J. T. Vogelstein. Statistical inference on errorfully observed graphs. *Journal of Computational and Graphical Statistics*, 24(4): 930–953, 2015.

S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova. Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3):223–247, 2015.

J. D. A. Relión, D. Kessler, E. Levina, and S. F. Taylor. Network classification with applications to brain connectomics. *The annals of applied statistics*, 13(3):1648, 2019.

S. Richardson and P. J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997.

J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville. Decoding brain states from fmri connectivity graphs. *Neuroimage*, 56(2):616–626, 2011.

C. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p*) models for social networks. *Social networks*, 29(2):173–191, 2007.

H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC press, 2005.

M. Salter-Townshend, A. White, I. Gollini, and T. B. Murphy. Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining*, 5(4):243–264, 2012.

A. Sapountzi and K. E. Psannis. Social networking data analysis tools & challenges. *Future Generation Computer Systems*, 86:893–913, 2018.

M. N. Schmidt and M. Morup. Nonparametric bayesian modeling of complex networks: An introduction. *IEEE Signal Processing Magazine*, 30(3):110–128, 2013.

P. A. Shaw, M. A. Mikusz, P. T. Nurmi, and N. A. J. Davies. Tacita-a privacy preserving public display personalisation service. *UbiComp 2018*, 2018.

M. Signorelli and E. C. Wit. Model-based clustering for populations of networks. *Statistical Modelling*, 20(1):9–29, 2020.

A. E. Sizemore, C. Giusti, A. Kahn, J. M. Vettel, R. F. Betzel, and D. S. Bassett. Cliques and cavities in the human connectome. *Journal of computational neuroscience*, 44(1): 115–145, 2018.

R. R. Sokal. A statistical method for evaluating systematic relationships. *Univ. Kansas, Sci. Bull.*, 38:1409–1438, 1958.

N. Sokhn, R. Baltensperger, L.-F. Bersier, J. Hennebert, and U. Ultes-Nitsche. Identification of chordless cycles in ecological networks. In *International Conference on Complex Sciences*, pages 316–324. Springer, 2012.

Y.-Y. Song and L. Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

E. Sprinzak, S. Sattath, and H. Margalit. How reliable are experimental protein–protein interaction data? *Journal of molecular biology*, 327(5):919–923, 2003.

X. Tang and C. C. Yang. Dynamic community detection with temporal dirichlet process. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 603–608. IEEE, 2011.

S.-H. Teng. Scalable algorithms for data and network analysis. *Foundations and Trends® in Theoretical Computer Science*, 12(1–2):1–274, 2016.

J. L. Teugels. Some representations of the multivariate bernoulli and binomial distributions. *Journal of multivariate analysis*, 32(2):256–268, 1990.

V. Vitelli, Ø. Sørensen, M. Crispino, A. Frigessi, and E. Arjas. Probabilistic preference learning with the mallows rank model. *The Journal of Machine Learning Research*, 2017.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

L. Wang and R. Jones. Big data analytics in cyber security: network traffic and attacks. *Journal of Computer Information Systems*, pages 1–8, 2020.

S. Wang, J. Arroyo, J. T. Vogelstein, and C. E. Priebe. Joint embedding of graphs. *arXiv preprint arXiv:1703.03862*, 2017.

J. G. White, E. Southgate, J. N. Thomson, S. Brenner, et al. The structure of the nervous system of the nematode caenorhabditis elegans. *Philos Trans R Soc Lond B Biol Sci*, 314(1165):1–340, 1986.

X. Yan and X. Su. *Linear regression analysis: theory and computing.* World Scientific, 2009.

S. Yang. Networks: An introduction by mej newman: Oxford, uk: Oxford university press. 720 pp. 2013.

J.-G. Young, G. T. Cantwell, and M. Newman. Bayesian inference of network structure from unreliable data. *Journal of Complex Networks*, 8(6):cnaa046, 2020.

S. J. Young and E. R. Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer, 2007.

J. Zhang, W. Cheng, Z. Wang, Z. Zhang, W. Lu, G. Lu, and J. Feng. Pattern classification of large-scale functional brain networks: identification of informative neuroimaging markers for epilepsy. *PloS one*, 7(5):e36733, 2012.

Y. Zhao, Y.-J. Wu, E. Levina, and J. Zhu. Link prediction for partially observed networks. *Journal of Computational and Graphical Statistics*, 26(3):725–733, 2017.

X. Zhu, M. Gerstein, and M. Snyder. Getting connected: analysis and principles of biological networks. *Genes & development*, 21(9):1010–1024, 2007.

X.-N. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. C. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos, et al. An open science resource for establishing reliability and reproducibility in functional connectomics. *Scientific data*, 1(1):1–13, 2014.