

Logistic-tree Normal Mixture for Clustering Microbiome  
Compositions

by

Jiongran Wang

Department of Statistical Science  
Duke University

Date: \_\_\_\_\_

Approved:

\_\_\_\_\_  
Li Ma, Supervisor

\_\_\_\_\_  
Surya Tapas Tokdar

\_\_\_\_\_  
Mike West

A thesis submitted in partial fulfillment of the  
requirements for the degree of Master of Science  
in the Department of Statistical Science  
in the Graduate School of  
Duke University

2023

ABSTRACT

Logistic-tree Normal Mixture for Clustering Microbiome  
Compositions

by

Jiongran Wang

Department of Statistical Science  
Duke University

Date: \_\_\_\_\_

Approved:

\_\_\_\_\_  
Li Ma, Supervisor

\_\_\_\_\_  
Surya Tapas Tokdar

\_\_\_\_\_  
Mike West

An abstract of a thesis submitted in partial fulfillment of the  
requirements for the degree of Master of Science  
in the Department of Statistical Science  
in the Graduate School of  
Duke University

2023

Copyright © 2023 by Jiongran Wang  
All rights reserved

# Abstract

Human microbiome has become an interesting research topic in recent years and a common task in the analysis of these data is to cluster microbiome compositions into subtypes. However, this seemingly standard task is very challenging in the microbiome composition context due to several key features of such data: discrete nature, sparsity and variable size. Common distance-based algorithms can not produce reliable results as they do not account for these features. In addition, existing model-based approaches are not flexible enough to capture the complex within-cluster variation from cross-cluster variation. An useful Bayesian generative model Dirichlet-tree multinomial mixtures (DTMM) has been proposed to overcome these challenges. DTMM indeed achieves reliable results, but it is still not flexible enough in characterizing covariance structure among taxa and lacks the scalability to higher dimensions. In this work, we propose another generative model, called the "Logistic-tree normal mixture" (LTNM), that addresses this need. The LTN kernel incorporates the tree-based decomposition as the Dirichlet-tree does, but it also models the branching probability using a multivariate logistic-normal distribution. Hence it has a rich covariance structure along with computational efficiency through Pólya-Gamma data augmentation technique. We perform extensive simulation studies to compare the performance of LTNM with other competitors. At last, we report a case study on the fecal data from the American Gut Project (AGP) to identify enterotypes (clusters) among participants with inflammatory bowel disease (IBD).

# Contents

<b>Abstract</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Methodology</b>	<b>4</b>
2.1 LTN and LTNM . . . . .	4
2.2 Discriminative taxa selection . . . . .	11
2.3 Prior specification and hyperparameter choice . . . . .	13
2.4 Inference strategy . . . . .	14
<b>3 Simulation</b>	<b>17</b>
3.1 Simulation setup . . . . .	17
3.2 Result analysis . . . . .	19
<b>4 Case study</b>	<b>23</b>
<b>5 Conclusion</b>	<b>27</b>
<b>A Posterior sampling for <math>\log \alpha^*</math></b>	<b>28</b>
<b>B Block Gibbs sampler for <math>\Omega</math> and <math>\tau</math></b>	<b>30</b>
<b>C Full conditionals of other parameters</b>	<b>32</b>
<b>D Posterior sampling algorithm for the LTNM</b>	<b>36</b>
<b>Bibliography</b>	<b>38</b>

## List of Tables

3.1	Average $R^2$ for each scenario . . . . .	19
3.2	RMSE of the Jaccard index under different scenarios . . . . .	21
4.1	Top 10 important OTUs in determining $C_{LS}$ . . . . .	25

# List of Figures

2.1	A typical phylogenetic tree over four OTUs . . . . .	5
2.2	Graphical representation of the LTNM with automatic taxa selection	12
3.1	Distribution of Jaccard Index for all models under all scenarios; models from left to right are: DMM, DTMM, K-ms, LTNM . . . . .	21
3.2	Discriminative taxa selection of one round simulation in LN-Diagonal kernel with weak signal . . . . .	22
4.1	Estimated pairwise co-clustering probabilities with different $\lambda_1$ . . . . .	24
4.2	Discriminative taxa selection of IBD group dataset . . . . .	25
4.3	Two-dimensional NMDS plots for the IBD group dataset . . . . .	26

# Chapter 1

## Introduction

The human microbiome collects the genetic information from all microbes inhabiting human body. The development of next-generation sequencing technology provides an efficient way of profiling the microbiome, through either shotgun metagenomic sequencing or amplicon sequencing on target genes (e.g. the 16S rRNA gene). Many preprocessing pipelines such as DADA2 [CMR<sup>+</sup>16] and MetaPhLAN [BMBM<sup>+</sup>21] have been developed to deal with the sequencing reads and report the results in terms of operational taxonomic unit (OTU) or amplicon sequence variant (ASV) abundance. Both OTUs and ASVs serve as the unit for downstream compositional analysis: each sample is a vector of counts on a list of units (OTUs or ASVs), representing the composition of the underlying community. The methodology developed in this work can be applied to both OTUs and ASVs, and for simplicity, we use the OTU to refer to the unit.

Heterogeneity is an evident property of microbiome samples. As a result, capturing and understanding the heterogeneity among samples is important for further microbiome analysis. Moreover, clustering the compositional data into subtypes and understanding the relation of these clusters with the host environment have become a central task. However, complex within- and cross- cluster variations hinder the implements of many popular clustering models. For example,  $k$ -means is a classical distance-based clustering method and has been extensively applied to many fields. But it fails to be applied to microbiome datasets even though the Bray-Curtis dissimilarity and the Unifrac distances, which are carefully tailored for microbiome

compositions, are chosen [LK05]. This is caused by the fact that  $k$ -means relies on the distance measures between samples and a single measure fails to account for the microbiome data given its discrete nature, sparsity and variable size.

In addition, [HHQ12] has proposed a model-based clustering method, called the Dirichlet multinomial mixture (DMM), which adopts a Dirichlet-multinomial schema and generates the sample-specific multinomial parameters from a finite mixture of Dirichlet components. However, a single concentration parameter in Dirichlet distribution is not enough to capture the complex variation among samples within each cluster. Hence, the clustering results generated by DMM is not reliable. To overcome these difficulties, [MM22] come up with a Bayesian generative model for clustering microbiome datasets, called the Dirichlet-tree multinomial mixture (DTMM). Instead of adopting a Dirichlet mixture components, DTMM chooses a Dirichlet-tree distribution [Den91] as the mixture components. Specifically, instead of a single concentration parameter to capture the variation, DTMM allows multiple concentration parameters, one for each internal node in the phylogenetic tree. As a result, DTMM allows more flexible cross-sample variation in the microbiome datasets. In addition, utilizing the information of the phylogenetic tree makes DTMM better capture the relations among the OTUs. At last, a node selection feature has been inserted into the DTMM framework that allows more interpretable inference results. However, the drawbacks of the DTMM are still evident. On the one hand, the DTMM fails to capture the variation across the internal nodes. On the other hand, the DTMM lacks the scalability to high dimensions since there involves two-dimensional integral approximation when the DTMM is implemented.

To obtain a flexible structure among internal nodes, [WMM21] has proposed another Bayesian generative model, called the logistic-tree normal (LTN) model. The

LTN assumes the log-odds ratio for all internal nodes are normally distributed. Specifically, assume the number of internal nodes in a phylogenetic tree is  $d$ . The DTMM allows  $d$  parameters to profile the cross-sample variation. In contrast, the LTN allows a  $d \times d$  matrix, which is a much more flexible structure, to profile the relation among internal nodes. On the other hand, the LTN could restore the conjugacy by applying Pólya-Gamma data augmentation technique [PSW13], which results in the efficient computation even in high-dimensional cases. Hence, we insert this LTN kernel into the sparse finite mixture framework and name it Logistic-tree normal mixture (LTNM) model. The LTNM adopts a multivariate Gaussian distribution as the mixture component which allows a more realistic structure for taxa, and we also incorporate the node selection feature into the LTNM framework for better profiling the latent clustering process.

We carry out extensive simulation studies to evaluate the performance of the LTNM based on the clustering and node selection results, and compare the LTNM to other competitors to verify the flexibility of the LTNM. Then with the proposed model, we report a case study of the American Gut Project (AGP) [MBK15, MHD<sup>+</sup>18] data to explore enterotypes of samples which are diagnosed with inflammatory bowel disease (IBD).

This thesis will be organized as follows: first we briefly review some popular existing algorithms; then we will introduce LTNM in detail; then we will do extensive simulation study to compare LTNM and other existing methods; at last we apply LTNM to a real microbiome study, the AGP, to analyze the inference results of LTNM.

# Chapter 2

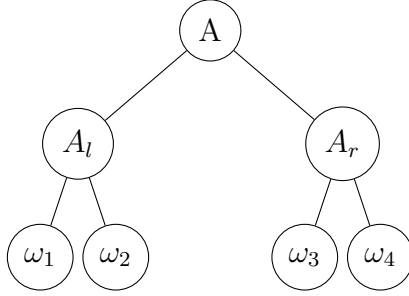
## Methodology

### 2.1 LTN and LTNM

In this section we will briefly review LTN model and introduce how we apply this kernel to cluster microbiome compositions into subtypes.

Consider a microbiome dataset with OTU counts of  $n$  samples  $\mathbf{X}_1, \dots, \mathbf{X}_n$ . Each sample is a vector containing  $M$  OTU counts denoted by  $\Omega = \{\text{OTU}_1, \dots, \text{OTU}_M\} = \{\omega_1, \dots, \omega_M\}$ . The OTU table is an  $n \times M$  matrix whose  $(i, j)$ -th element represents the count of OTU  $j$  in  $i$ -th sample. OTUs in a microbiome study are evolutionarily related. This relationship can be summarized into a rooted phylogenetic tree, where each internal node can be viewed as a “taxa” that represents the most recent common ancestor of its descendant OTUs [MM22]. Suppose  $\mathcal{T} = \mathcal{T}(\mathcal{I}, \mathcal{U}; \mathcal{E})$  is a rooted full binary phylogenetic tree over the  $M$  OTUs, where  $\mathcal{I}$ ,  $\mathcal{U}$ ,  $\mathcal{E}$  denote the set of interior nodes, leaves, and edges respectively. For a leaf node  $A \in \mathcal{U}$ , which contains only a single OTU  $\omega$  by definition, we define  $A = \{\omega\}$ . Then each interior node  $A \in \mathcal{I}$  can be defined iteratively from leaf to root through  $A = A_l \cup A_r$  where  $A_l$  and  $A_r$  represent left and right children nodes of  $A$ . Figure 2.1 shows an example of a phylogenetic tree over four OTUs.

The OTU table contains count data. For each sample, suppose  $\mathbf{X} = (X_1, \dots, X_M)$  is the associated OTU count, and let  $N = \sum_{i=1}^M X_i$  represent the total number of OTU counts. In this work, we treat the total counts  $N_i$ 's as known since they are



**Figure 2.1:** A typical phylogenetic tree over four OTUs

artificial quantities which depend on the sequencing depth. When  $N$  is given, a natural choice of sampling model for  $\mathbf{X}$  would be multinomial model:

$$\mathbf{X} \mid N, \mathbf{p} \sim \text{Multinomial}(N, \mathbf{p}) \quad (2.1)$$

where  $\mathbf{p}$  lies in  $(M - 1)$ -simplex, is the underlying OTU relative abundance vector by definition. DMM [HHQ12] models  $\mathbf{p}$  through a mixture of Dirichlet distribution:

$$\mathbf{p}_i \mid \boldsymbol{\pi}, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_K \stackrel{iid}{\sim} \sum_{k=1}^K \pi_k \text{Dir}(\boldsymbol{\alpha}_k) \quad (2.2)$$

However, a single concentration parameter in Dirichlet distribution is hard to capture the cross-sample variability within clusters and the simplex constraint sometimes makes modeling difficult. Hence, the relative abundance vector is usually mapped into an Euclidean space through a so-called log-ratio transformation [Ait82].

There are three popular choices of log-ratio transformation: additive log-ratio (*alr*), centered log-ratio (*clr*) [Ait82] and isometric log-ratio (*ilr*) [SWMD17]. The *alr* and *clr* transform are defined as follows:

$$alr(\mathbf{p}) = \{\log(p_i/p_K) : i = 1, \dots, K - 1\}$$

$$clr(\mathbf{p}) = \{\log(p_i/g(\mathbf{p})) : i = 1, \dots, K\}$$

where  $g(\mathbf{p})$  is the geometric mean of  $\mathbf{p}$ . Basically, the *alr* transform treats  $p_K$  as basement and do log-ratio transformation for other components. Both of the *alr* and *clr* transform do not utilize information of the phylogenetic tree. On the other hand, the *ilr* transform is built from a sequential binary partition of the original variable space. Hence, when applying the *ilr* transform to microbiome datasets, we can introduce the phylogenetic tree as a natural and informative sequential binary partition [SWMD17]. For each interior node  $A \in \mathcal{I}$ , the associated “balances”  $\eta(A)$  is defined as follows:

$$\eta(A) = \sqrt{\frac{|A_l||A_r|}{|A_l| + |A_r|}} \log \frac{g(\mathbf{p}(A_l))}{g(\mathbf{p}(A_r))}$$

where  $|A_l|$  and  $|A_r|$  represent the number of OTUs in the left and right sub-tree of node  $A$ ,  $g(\mathbf{p}(A_l))$  and  $g(\mathbf{p}(A_r))$  represent the geometric mean of the underlying relative abundance vector associated with the left and right sub-tree of node  $A$ . Basically, the *ilr* transform maps a composition  $\mathbf{p}$  into a series of “balances” associated with the interior nodes of the tree.

After applying the log-ratio transformation to a composition  $\mathbf{p}$ , we relax the simplex constraint. A log-ratio normal (LN) model assumes that the log-ratios follow a multivariate normal distribution. This model setting has been applied to microbiome data analysis in many ways [XCFL13, B17]. However, when the number of OTUs increases, LN model will result in computational challenges due to the lack of conjugacy between the multinomial likelihood and multivariate normal.

In contrast, we can consider another classical approach: the Dirichlet-tree multinomial model (DTM) [Den91]. Given a phylogenetic tree  $\mathcal{T}$  over the OTUs, the DTM uses the fact that a  $M$ -dimensional multinomial density is equivalent to a product of

$M - 1$  binomial densities. Specifically, the sampling model in Eq. (2.1) is equivalent to for all  $A \in \mathcal{I}$

$$Y(A_l) | Y(A), \theta(A) \stackrel{ind}{\sim} \text{Binomial}(Y(A), \theta(A))$$

where  $Y(A) = \sum_{i:\omega_i \in A} X_i$  represents the total number of OTU counts associated with sub-tree of node  $A$  and  $\theta(A) = \sum_{i:\omega_i \in A_l} X_i / \sum_{i:\omega_i \in A} X_i$  represents the “branching” ratio of node  $A$ . In order to obtain conjugacy, the DTM puts a beta model for each  $\theta(A)$ . That is

$$\theta(A) \stackrel{ind}{\sim} \text{Beta}(\mu(A)\tau(A), (1 - \mu(A))\tau(A))$$

where  $\mu(A)$  is the prior mean of  $\theta(A)$  and  $\tau(A)$  controls the variability of  $\theta(A)$  around its mean. DTM is not only transforming the abundance vector  $\mathbf{p}$  in terms of  $\theta(A)$ ’s through the phylogenetic tree, but also decomposing the sampling model, which is the biggest difference between the DTM and the *ilr*-based LN approach. As a result, this decomposition of the sampling model makes DTM computationally efficient through the beta-binomial conjugacy. Given its advantages, DTM has been treated as a mixture component in DTMM [MM22]:

$$\boldsymbol{\theta} | \boldsymbol{\pi}, \{(\boldsymbol{\mu}_k^*, \boldsymbol{\tau}_k^*)\}_{k=1}^K \stackrel{iid}{\sim} \sum_{k=1}^K \pi_k \text{DT}(\boldsymbol{\mu}_k^*, \boldsymbol{\tau}_k^*) \quad (2.3)$$

where  $\mathbf{p} = \text{tr}^{-1}(\boldsymbol{\theta})$ ,  $\boldsymbol{\theta} \sim \text{DT}(\boldsymbol{\mu}, \boldsymbol{\tau})$

However, the drawback of the DTM mixture component is that it fails to capture the covariance among taxa. Actually, the independence between  $\theta(A)$ ’s is the underlying assumption of the DTM, which is unrealistic in practical sense. In contrast, the LN model allows a  $(M - 1) \times (M - 1)$  covariance matrix even if the computation may not be efficient.

Motivated by the LN and the DTM, [WMM21] proposes a hybrid model, named logistic-tree normal (LTN), that combines the decomposition of the sampling model under the DTM with the log-ratio transformation under the LN. Hence, LTN is both computationally efficient and having flexible covariance structure. Specifically, the LTN applies a logistic transform to the branching probability  $\theta(A)$  for each interior node  $A$ . That is

$$\psi(A) = \log \frac{\theta(A)}{1 - \theta(A)}$$

and the LTN posits the joint distribution of the log-odds on the interior nodes is multivariate Gaussian. Hence, the full framework of the LTN is summarized as follows:

$$Y(A_l) | Y(A), \theta(A) \stackrel{ind}{\sim} \text{Binomial}(Y(A), \theta(A)) \quad (2.4)$$

$$\psi(A) = \log \frac{\theta(A)}{1 - \theta(A)} \quad (2.5)$$

$$\boldsymbol{\psi} \sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (2.6)$$

Parameter  $\boldsymbol{\Sigma}$  ensures the flexibility of the LTN. But how to restore the conjugacy such that the LTN could be efficiently implemented? Fortunately, this question could be solved by utilizing a data-augmentation technique called Pólya-Gamma (PG) augmentation [PSW13].

Based on [PSW13], the binomial likelihood in Eq. (2.4)&(2.5) could be written as follows:

$$p(Y(A_l) | Y(A), \psi(A)) \propto \frac{(e^{\psi(A)})^{Y(A_l)}}{(1 + e^{\psi(A)})^{Y(A)}} = 2^{-Y(A)} e^{\kappa(A)\psi(A)} \int_0^\infty e^{-\omega\psi(A)^2/2} f(\omega) d\omega$$

where  $\kappa(A) = Y(A_l) - Y(A)/2$  and  $f(\omega)$  is the probability density function of PG( $Y(A)$ , 0) distribution. Then an auxiliary random variable  $\omega(A)$  which is inde-

pendent of  $Y(A_l)$  given  $Y(A)$  and  $\psi(A)$  has been introduced to the LTN framework:

$$\omega(A) \mid Y(A), \psi(A) \sim \text{PG}(Y(A), \psi(A))$$

and the conditional density is

$$p(\omega(A) \mid Y(A), \psi(A)) \propto e^{-\omega(A)\psi(A)^2/2} f(\omega(A))$$

Then the joint density of  $\omega(A)$  and  $Y(A_l)$  given  $Y(A)$  and  $\psi(A)$  is

$$p(\omega(A), Y(A_l) \mid Y(A), \psi(A)) \propto 2^{-Y(A)} e^{\kappa(A)\psi(A) - \omega(A)\psi(A)^2/2} f(\omega(A))$$

whose exponential part contains quadratic term of  $\psi(A)$ , hence is conjugate to the multivariate normal likelihood of  $\psi$ .

With the help of Pólya-Gamma augmentation, the LTN restore the conjugacy and could be efficiently implemented. This flexible framework could be extensively applied to microbiome data analysis through being embedded into more complicated hierarchical models [WMM21]. In this work, we are interested in applying the LTN kernel to cluster microbiome compositions into subtypes and we propose a Bayesian generative model, named logistic-tree normal mixture (LTNM), to solve this problem.

In general, the LTNM is a combination of the LTN and the Gaussian mixture model (GMM). A typical microbiome dataset usually contains an OTU table, a phylogenetic tree and the associated covariate information. The LTN model factorizes the sampling likelihood and represents the abundance vector  $\mathbf{p}$  in terms of  $\psi(A)$ 's. The information of  $i$ -th sample (tree),  $i = 1, \dots, n$ , is characterized by the parameter

$\psi$ . Hence, in cluster setting, we introduce a latent label  $c$  for each  $\psi$ . That is

$$\psi_i \mid c_i = k, \boldsymbol{\mu}, \boldsymbol{\Sigma} \sim \text{MVN}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad \text{for all } i \in 1, \dots, n$$

where  $k$  represents the  $k$ -th cluster. The number of clusters is an important topic in clustering analysis. Many traditional clustering methods, e.g. k-means clustering [Mac67], predetermine the number of clusters. However, this setting is too restrictive under many real applications since new clusters should always be expected to occur. In order to relax this constraint, many nonparametric approaches have been introduced, e.g. dirichlet process mixture [EW95] and mixture of finite mixture [MH18]. Inference strategies for these Bayesian nonparametric models, such as described in [Nea00] or [IJ01], can be applied.

On the other hand, there is another approach to deal with the difficulty of pre-determining the number of clusters: sparse finite mixture. This idea has been introduced within the framework of Bayesian model-based clustering [MWFSG16], and serves as a bridge between the standard finite mixture and Dirichlet process mixture. The key idea is selecting a large number of components (greater than the anticipated number), and choosing a sparse symmetric Dirichlet prior which encourages the extra components to have weights close to zero. [GR01] has shown that a  $K$  component finite mixture model with symmetric Dirichlet prior, i.e.  $\text{Dir}_K(\alpha/K)$ , on the weights approximates a Dirichlet process mixture with concentration parameter  $\alpha$  as  $K$  increases. We adopt sparse finite mixture in the LTNM, and the basic framework of the LTNM is as follows:

$$Y(A_l) \mid Y(A), \theta(A) \stackrel{\text{ind}}{\sim} \text{Binomial}(Y(A), \theta(A)) \quad \text{for all } A \in \mathcal{I} \quad (2.7)$$

$$\psi(A) = \log \frac{\theta(A)}{1 - \theta(A)} \quad \text{for all } A \in \mathcal{I} \quad (2.8)$$

$$\boldsymbol{\psi}_i \stackrel{ind}{\sim} \text{MVN}(\boldsymbol{\mu}_{c_i}, \boldsymbol{\Sigma}_{c_i}) \quad \text{for all } i \in 1, \dots, n \quad (2.9)$$

$$c_i \stackrel{iid}{\sim} \text{Categorical}(\pi_1, \dots, \pi_K) \quad \text{for all } i \in 1, \dots, n \quad (2.10)$$

$$\pi_1, \dots, \pi_K \sim \text{Dir}\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right) \quad (2.11)$$

The prior specification of parameters of the LTNM, especially  $p(\boldsymbol{\Sigma})$ , needs to be treated carefully, which will be discussed in later section.

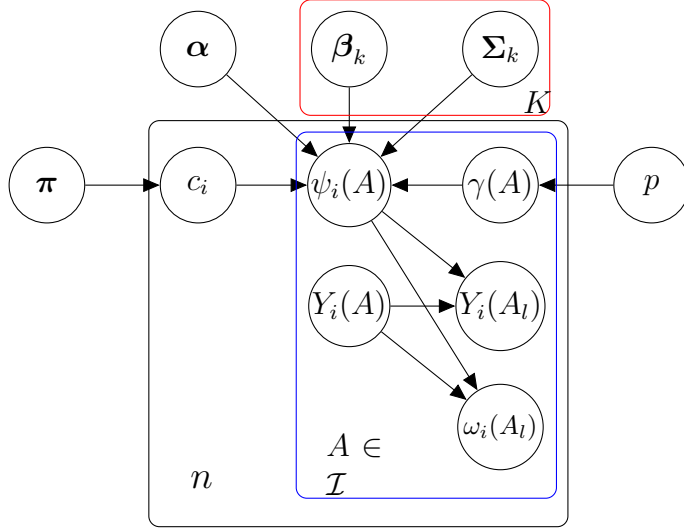
## 2.2 Discriminative taxa selection

A typical microbiome dataset usually contains a high-dimensional OTU table. As a result, for each transformed sample, the number of taxa is very large. However, in many applications, not all of these taxa play a role in determining the underlying cluster. When this is the case, identifying these signature taxa can enhance the sensitivity for separating the clusters, and improve the interpretability of the resulting inference [MM22]. In this section, we describe how to incorporate the taxa selection into the LTNM model.

For each  $A \in \mathcal{I}$ , we define  $\gamma(A) \in \{0, 1\}$  as an indicator of whether node  $A$  contributes to the clustering. If  $\gamma(A) = 1$ , node  $A$  is active in the latent clustering; otherwise, node  $A$  is inactive and we require all the clusters at node  $A$  to share the same branching probability. To meet this requirement, we can rewrite  $\boldsymbol{\mu}_{c_i}$  in Eq. 2.9 as follows:

$$\boldsymbol{\mu}_{c_i} = \boldsymbol{\alpha} \odot (\mathbf{1}_d - \boldsymbol{\gamma}) + \boldsymbol{\beta}_{c_i} \odot \boldsymbol{\gamma}$$

where  $d = M - 1$  is the number of internal nodes;  $\odot$  represents element-wise multipli-



**Figure 2.2:** Graphical representation of the LTNM with automatic taxa selection;  $\gamma = \{\gamma(A) : A \in \mathcal{I}\}$  is the collection of indicators of all internal nodes;  $\alpha$  and  $\beta_{c_i}$  represent population and cluster specific mean of the branching probabilities of all internal nodes respectively, where  $c_i \in \{1, \dots, K\}$ . Hence, for  $A \in \mathcal{I}$ , if  $\gamma(A) = 0$ , the mean of the branching probability on node  $A$  is  $\alpha(A)$ , which results in the indistinguishability of node  $A$  for all clusters. On the other hand, if  $\gamma(A) = 1$ , node  $A$  will be active in clustering since it has the cluster specific mean of the branching probability.

We insert  $\gamma$  into the LTNM in the following way:

$$\psi_i | \gamma \stackrel{ind}{\sim} \text{MVN}(\alpha \odot (\mathbf{1}_d - \gamma) + \beta_{c_i} \odot \gamma, \Sigma_{c_i}) \quad \text{for all } i \in 1, \dots, n \quad (2.12)$$

$$\gamma(A) \stackrel{ind}{\sim} \text{Binom}(p) \quad \text{for } A \in \mathcal{I} \quad (2.13)$$

with appropriate prior for  $\alpha$ ,  $\beta_{c_i}$  and  $p$  respectively. Figure 2.2 is a graphical representation of the LTNM.

## 2.3 Prior specification and hyperparameter choice

The prior distributions of the parameters in the LTNM are deliberately specified. Specifically, the Inverse-Wishart distribution is usually a natural choice for  $p(\Sigma)$ . However, this distribution performs poorly in high-dimensional cases, which are common in microbiome datasets. In the LTNM, the inverse covariance is expected to not only capture the correlation among taxa, but also remain sparsity in the high-dimensional cases. Hence, we adopt the graphical lasso prior [Wan12] on the inverse covariance, which can be represented in the following form:

$$p(\mathbf{\Omega} \mid \lambda) = C^{-1} \prod_{i < j} \{\text{DE}(\omega_{ij} \mid \lambda)\} \prod_{i=1}^d \{\text{EXP}(\omega_{ii} \mid \frac{\lambda}{2})\} \mathbf{1}_{\mathbf{\Omega} \in M^+} \quad (2.14)$$

where  $\mathbf{\Omega} = \Sigma^{-1}$  is the precision matrix,  $\text{DE}(x \mid \lambda)$  represents the double exponential density function of the form  $p(x) = \lambda/2 \exp(-\lambda|x|)$ ,  $\text{EXP}(x \mid \lambda)$  represents the exponential density function of the form  $p(x) = \lambda \exp(-\lambda x) \mathbf{1}_{x > 0}$ ,  $M^+$  is the space of positive definite matrices, and  $C$  is the normalizing constant not involving  $\lambda$  or  $\mathbf{\Omega}$ .  $\lambda$  is the shrinkage parameter, and a Gamma prior can be put on it, i.e.  $\lambda \sim \text{Gamma}(r, s)$ , which leads to a Gamma conditional posterior  $\lambda \sim \text{Gamma}(r + d(d+1)/2, s + \|\mathbf{\Omega}\|_1/2)$ . Here  $\|\mathbf{\Omega}\|_1$  is the  $L_1$  norm of  $\mathbf{\Omega}$ . On the other hand, we can allow different  $\lambda_{ij}$ 's for different  $\omega_{ij}$ 's rather than using one single  $\lambda$  to penalize the whole matrix. In the LTNM, we choose two different  $\lambda$ 's ( $\lambda_1$  and  $\lambda_2$ ) for the diagonal and off-diagonal terms of  $\mathbf{\Omega}$  respectively, and we set a relatively small value for  $\lambda_1$  and a relatively large value for  $\lambda_2$  which result in more penalty on the off-diagonal terms and less penalty on the diagonal terms. In addition, we set the same  $\lambda_1$  and  $\lambda_2$  for all cluster specific precision matrices  $\mathbf{\Omega}_k$  for  $k = 1, \dots, K$ .

The prior distributions of  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$  are as follows:

$$\boldsymbol{\alpha} \sim \text{MVN}(\mathbf{0}, 5 * \mathbf{I})$$

$$\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_k \stackrel{iid}{\sim} \text{MVN}(\mathbf{0}, 5 * \mathbf{I})$$

Under this setting, we assume the population mean variable and the cluster specific mean variables have the same prior distributions. For parameter  $p$ , we use a Beta prior to keep the conjugacy, i.e.  $p \sim \text{Beta}(a, b)$ , where one may choose  $a = 1$  and  $b = 1$  to get a non-informative prior, or  $a = 1$  and  $b = 0.5d$  as suggested by [RG18].

As mentioned before, a sparse finite mixture model has been embedded into the LTNM. Here we set the number of clusters  $K = 50$ , and a very small  $\alpha/K$  value will induce sparsity. Instead of making  $\alpha/K = l$  for some small constant  $l$ , we define  $\alpha^* = \alpha/K$  and assign a Gamma prior to it, i.e.  $\alpha^* \sim \text{Gamma}(l_1, l_2)$ . Then we adopt a random-walk Metropolis-Hastings for  $\log \alpha^*$  (since  $\alpha^* > 0$ ) for posterior sampling.

## 2.4 Inference strategy

Putting all pieces together, all components in the LTNM except  $\boldsymbol{\Omega}_k$  and  $\alpha^*$  are conditionally conjugate. Metropolis-Hastings can be applied to sample  $\log \alpha^*$  (see Appendix A). To effectively sample  $\boldsymbol{\Omega}$ , we adopt a data augmentation scheme proposed in [Wan12]. Specifically, given the fact that the double exponential distribution can be represented as a scale mixture of normals [AM74, Wes87], we can introduce a latent scale parameter  $\tau$  to Eq. (2.14):

$$p(\boldsymbol{\omega}_k \mid \boldsymbol{\tau}_k, \lambda) = C_{\boldsymbol{\tau}_k}^{-1} \prod_{i < j} \left\{ \frac{1}{\sqrt{2\pi\tau_{kij}}} \exp\left(-\frac{\omega_{kij}^2}{2\tau_{kij}}\right) \right\} \prod_{i=1}^d \left\{ \frac{\lambda}{2} \exp\left(-\frac{\lambda}{2}\omega_{kii}\right) \right\} \mathbf{1}_{\boldsymbol{\Omega}_k \in M^+}$$

$$p(\boldsymbol{\tau}_k | \lambda) \propto C_{\boldsymbol{\tau}} \prod_{i < j} \frac{\lambda^2}{2} \exp\left(-\frac{\lambda^2}{2} \tau_{kij}\right)$$

where  $\boldsymbol{\omega}_k = \{\omega_{kij} : 1 \leq i \leq j \leq d\}$  is a vector of the upper off-diagonal and diagonal terms of  $\boldsymbol{\Omega}_k$ ,  $\boldsymbol{\tau}_k = \{\tau_{kij} : 1 \leq i < j \leq d\}$  is the latent scale parameters and  $C_{\boldsymbol{\tau}_k}^{-1}$  is the normalizing constant involving  $\boldsymbol{\tau}_k$ . Then the data-augmented target distribution can be expressed as follows:

$$p(\boldsymbol{\Omega}_k, \boldsymbol{\tau}_k | \boldsymbol{\mu}_k, \boldsymbol{\psi}, \lambda) \propto |\boldsymbol{\Omega}_k|^{\frac{n}{2}} \exp\left\{-\text{tr}\left(\frac{1}{2} \mathbf{S}_k \boldsymbol{\Omega}_k\right)\right\} \prod_{i < j} \left\{\tau_{kij}^{-\frac{1}{2}} \exp\left(-\frac{\omega_{kij}^2}{2\tau_{kij}}\right) \exp\left(-\frac{\lambda^2}{2} \tau_{kij}\right)\right\} \prod_{i=1}^d \left\{\exp\left(-\frac{\lambda}{2} \omega_{kii}\right)\right\} 1_{\boldsymbol{\Omega}_k \in M^+} \quad (2.15)$$

where  $\mathbf{S}_k = \sum_{i:c_i=k} (\boldsymbol{\psi}_i - \boldsymbol{\mu}_k)(\boldsymbol{\psi}_i - \boldsymbol{\mu}_k)^T$  and  $\boldsymbol{\mu}_k$  is defined in Eq. (2.2). [Wan12] proposed an efficient block Gibbs sampler for the data-augmented target distribution after appropriate reparametrization (see Appendix B for details). Except  $\boldsymbol{\Omega}_k$  and  $\alpha^*$ , the full conditionals of all other terms could be analytically derived (see Appendix C for details). Then the whole sampling algorithm for the LTNM is summarized in Algorithm 1 (see Appendix D).

Given the clustering and taxa selection goal, we focus on  $\mathbf{c}$  and  $\boldsymbol{\gamma}$  parameters. After discarding the first  $B$  samples as burn-in, we obtain the posterior samples of  $\mathbf{c}$  and  $\boldsymbol{\gamma}$  denoted as  $[\{\boldsymbol{\gamma}^{(B+1)}, \mathbf{c}^{(B+1)}\}, \dots, \{\boldsymbol{\gamma}^{(T)}, \mathbf{c}^{(T)}\}]$ . There are many ways of yielding a representative cluster. Here we report the least-squared model-based clustering [Dah06], which incorporates information from all posterior samples and output one of the observed clustering results. Specifically, for each clustering  $\mathbf{c}$  in  $\{\mathbf{c}^{(B+1)}, \dots, \mathbf{c}^{(T)}\}$ , a  $n \times n$  association matrix  $\boldsymbol{\delta}(\mathbf{c})$  is defined whose  $(i, j)$  element is 1 if  $\mathbf{c}_i = \mathbf{c}_j$  and 0 otherwise. The pairwise clustering probability matrix  $\Pi$  is

estimated by taking element-wise average of these association matrices. The least-square clustering  $\mathbf{C}_{LS}$  is one of the observed clustering  $\mathbf{c}^{(t)}$  which minimizes the sum of squared deviations of its associated matrix  $\boldsymbol{\delta}(\mathbf{c}^{(t)})$  from the estimated pairwise clustering probability matrix  $\hat{\Pi}$ , and it is defined as follows

$$\mathbf{C}_{LS} = \arg \min_{\mathbf{c}^{(t)}: B < t \leq T} \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} (\delta_{ij}(\mathbf{c}^{(t)}) - \hat{\Pi}_{ij})^2$$

For taxa selection, we report a representative  $\gamma$  with the highest posterior probabilities. Specifically, for each  $A \in \mathcal{I}$ , calculate the posterior probabilities,  $\hat{p}_1 = p(\gamma(A) = 1)$  and  $\hat{p}_0 = p(\gamma(A) = 0)$ , through the posterior samples after burn-in period. Then if  $\hat{p}_1 > \hat{p}_0$ , we set  $\gamma(A) = 1$  and 0 otherwise. Repeating this procedure for all internal nodes  $A$  yields a representative  $\gamma$ .

# Chapter 3

## Simulation

### 3.1 Simulation setup

We carry out several simulation studies to compare the proposed model against existing methodologies for clustering microbiome count data: namely the Dirichlet-tree multinomial mixtures (DTMM) [MM22], the Dirichlet multinomial mixtures (DMM) [HHQ12], and the k-means algorithm (K-ms) [Llo82].

We simulate datasets with  $N = 90$  samples,  $K = 3$  clusters, and  $M = 50$  OTUs, where the OTU counts are generated from mixture of LN kernels. Specifically, the OTU counts  $\mathbf{X}_1, \dots, \mathbf{X}_{90}$  are generated as follows:

$$\begin{aligned}\mathbf{X}_i | \mathbf{p}_i &\stackrel{iid}{\sim} \text{Multinomial}(n_i, \mathbf{p}_i) \\ n_i &\stackrel{iid}{\sim} \text{Neg-Binom}(15000, 20) \\ \mathbf{p}_i &= \text{alr}^{-1}(\boldsymbol{\eta}_i) \\ \boldsymbol{\eta}_i &\stackrel{iid}{\sim} \sum_{k=1}^K \pi_k \cdot \text{MVN}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\end{aligned}$$

We treat a randomly-generated fully-rooted binary tree as the "phylogenetic tree", and consider three different types of covariance matrix  $\boldsymbol{\Sigma}$ . Under each scenario, we assume  $(\pi_1, \pi_2, \pi_3) = (\frac{4}{9}, \frac{3}{9}, \frac{2}{9})$  and  $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3$ . Three types of covariance matrix are described as follows:

- **Diagonal:**  $\boldsymbol{\Sigma}$  is a  $49 \times 49$  symmetric matrix and it has to be positive definite.

In this case, we set  $\sigma_1^2 = 1$  and  $\sigma_2^2 = 0.05$ . Let the first 31 diagonal entries equal to  $\sigma_1^2$  and the rest 18 diagonal entries equal to  $\sigma_2^2$ . All off-diagonal terms equal to 0.

- **Block:** In this case, we divide the whole matrix into two blocks whose dimensions are  $31 \times 31$  and  $18 \times 18$  respectively. The diagonal entries equal to 1 in the first block and equal to 0.05 in the second block. Then randomly choose 75 of upper diagonal elements in the first block and set them equal to 0.05. Randomly choose 27 of upper diagonal elements in the second block and set them equal to 0.04. All other entries are set to 0.
- **General:** Given the Block case, this setting makes some changes by adding covariance between blocks. Specifically, randomly choose 114 of upper elements between two blocks defined in Block case and set them equal to 0.02. All other entries are still set to 0.

Under each covariance matrix setting, we define  $\boldsymbol{\mu}_k = (c_1, \dots, c_{31}, d_{1k}, \dots, d_{18k})$ . In each case, we assume  $c_1 = \dots = c_{31} = c$  and  $d_{ik} \stackrel{ind}{\sim} \text{Unif}(2, 4)$  for all  $i \in \{1, \dots, 18\}$  and  $k \in \{1, 2, 3\}$ . By choosing different  $c$ , the simulated datasets can have different strength of the signal [And01], which measures the within-cluster variability against the total variation. The signal  $R^2$  is defined as follows:

$$R^2 = \frac{\sum_{k=1}^3 \sum_{i=1}^{N-1} \sum_{j=i+1}^N d_{BC}(\mathbf{X}_i, \mathbf{X}_j)^2 \epsilon_{ij}^k / N_k}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N d_{BC}(\mathbf{X}_i, \mathbf{X}_j)^2 / N}$$

where  $d_{BC}(\cdot, \cdot)$  is the Bray-Curtis dissimilarity,  $\epsilon_{ij}^k = 1$  if sample  $i$  and  $j$  are both in the cluster  $k$  and 0 otherwise,  $N_k$  is the number of samples in cluster  $k$ . In this simulation study, we choose  $c = 6$  and  $c = 1$  to generate datasets with strong and weak signals respectively. For each scenario, we conduct 100 rounds of simulations and summarize

the average  $R^2$  in Table 3.1. When  $R^2$  is large, the total variation is almost only contributed by the within-cluster variation, which results in the separability of the whole datasets is poor. We would like to evaluate the performance of models on both separable and non-separable datasets. In addition, the data-generating mechanism is based on the LN kernel, which indicates the tree-based log-odds  $\psi_i(A)$ 's are not normally distributed. Hence, we can also test the robustness of the proposed model in this simulation study.

**Table 3.1:** Average  $R^2$  for each scenario

	Signal	LN-Diagonal	LN-Block	LN-General
$c = 1$	Strong	0.40	0.54	0.54
$c = 6$	Weak	0.97	0.96	0.97

In each simulation round, we run the Gibbs sampler for the LTNM for 2000 iterations and discard the first 1000 iterations. The choices of priors and hyperparameters for the LTNM are based on the discussions of Section 2.3. In addition, we set  $\lambda_1 = 1$ ,  $\lambda_2 = 200$ , and the initial values of  $\mathbf{c}$  is the output of PAM [KR90] with  $k = 20$ . We set  $k = 3$ , which is the correct number of clusters, for K-ms when implementing this method.

## 3.2 Result analysis

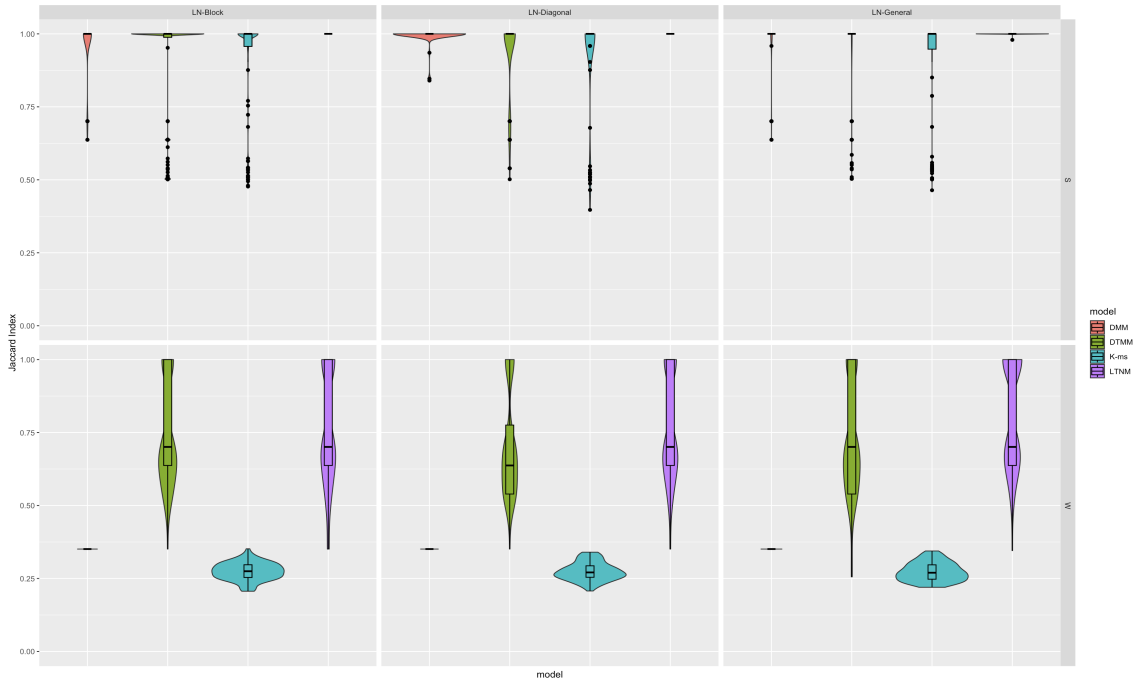
The metric we choose to compare the clustering results of different models is the Jaccard index [Jac12]. For a clustering result  $\mathbf{c}$  and the true cluster  $\mathbf{c}_t$ , the Jaccard index between  $\mathbf{c}$  and  $\mathbf{c}_t$ ,  $J(\mathbf{c}, \mathbf{c}_t)$ , is defined as  $|\mathbf{c} \cap \mathbf{c}_t|/|\mathbf{c} \cup \mathbf{c}_t|$ , where the numerator denotes the number of pairs of samples that belong to the same cluster under both  $\mathbf{c}$  and  $\mathbf{c}_t$  and the denominator denotes the number of pairs of samples that belong to the same cluster under at least one of  $\mathbf{c}$  and  $\mathbf{c}_t$ . Note that

$0 \leq J(\mathbf{c}, \mathbf{c}_t) \leq 1$  by definition. In each scenario, we compute the root mean square error  $RMSE = \sqrt{\sum_{i=1}^{100} (J(\mathbf{c}_{LS}, \mathbf{c}_t) - 1)^2 / 100}$  for each method where  $\mathbf{c}_{LS}$  is the representative cluster in each simulation round. The RMSE of all methods under all scenarios is summarize in Table 3.2. As the table shows, when the signal is strong, i.e. the data is separable, all models perform better than the case where the signal is weak. Especially the RMSE of the LTNM is 0 under all three kernels, which means the LTNM can exactly determine the cluster label for each sample when the signal is strong. Another notable thing is the RMSE of all competitors is lower under LN-Diagonal compared with other two kernels. This is caused by the fact that LN-Diagonal kernel has the simplest covariance structure. When the signal is weak, both the LTNM and DTMM outperform the other two methods. Without utilizing the phylogenetic tree information, the DMM and the K-ms output unreliable clustering results. In contrast, the LTNM and DTMM can still identify the correct clustering in most simulation rounds, which can be seen in Figure 3.1.

This plot shows the distribution of the Jaccard indexes for each method under all scenarios. We realize although the performance of all models is good when the signal is strong, the variations of all competitors are much larger than the LTNM's. On the other hand, when the signal is weak, the DMM identifies all samples as one single cluster and the K-ms assigns samples to incorrect clusters on all simulation rounds. As a result, their clustering results are not reliable. From this figure we notice the clustering results of the LTNM and the DTMM are similar to each other on average. However, the variation of the LTNM is less than DTMM's and the clustering results of the LTNM are better than DTMM's on most rounds under LN-Diagonal and LN-General kernels. We believe this is caused by the more flexible covariance structure of the LTNM.

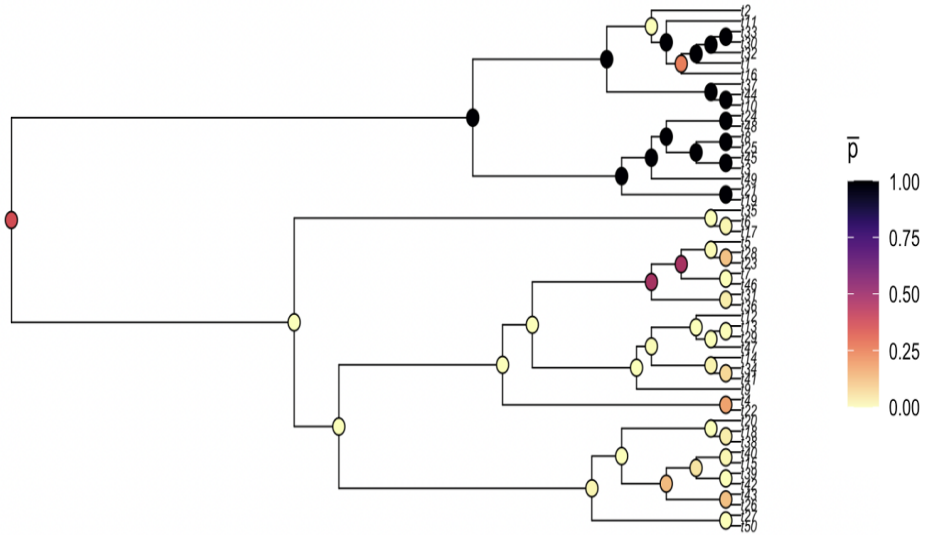
**Table 3.2:** RMSE of the Jaccard index under different scenarios

Kernel	Signal	LTNM	DTMM	DMM	K-ms
LN-Diagonal	W	<b>0.30</b>	0.34	0.65	0.73
	S	<b>0.00</b>	0.16	0.03	0.19
LN-Block	W	<b>0.31</b>	0.32	0.65	0.73
	S	<b>0.00</b>	0.21	0.10	0.21
LN-General	W	<b>0.26</b>	0.34	0.65	0.73
	S	<b>0.00</b>	0.18	0.10	0.22

**Figure 3.1:** Distribution of Jaccard Index for all models under all scenarios; models from left to right are: DMM, DTMM, K-ms, LTNM

Next we choose one simulation round to demonstrate the result of discriminative taxa selection. Here we choose round 3 under LN-Diagonal kernel with weak signal. Figure 3.2 shows the taxa selection result. The tree structure in this figure is the phylogenetic tree we use to generate datasets and implement the LTNM and the

DTMM. The degree of dark in node  $A \in \mathcal{I}$  represents the posterior probability of  $\gamma(A) = 1$ . As a result, a darker node denotes the probability is closer to 1. A black node means  $P(\gamma(A) = 1) = 1$ . Based on our simulation setting, all internal nodes in right sub-tree of the root should play a role in the latent clustering process. From this figure we can conclude the LTNM exactly identifies 16 of 18 active nodes and the other two nodes with intermediate and low posterior active probabilities respectively.



**Figure 3.2:** Discriminative taxa selection of one round simulation in LN-Diagonal kernel with weak signal

# Chapter 4

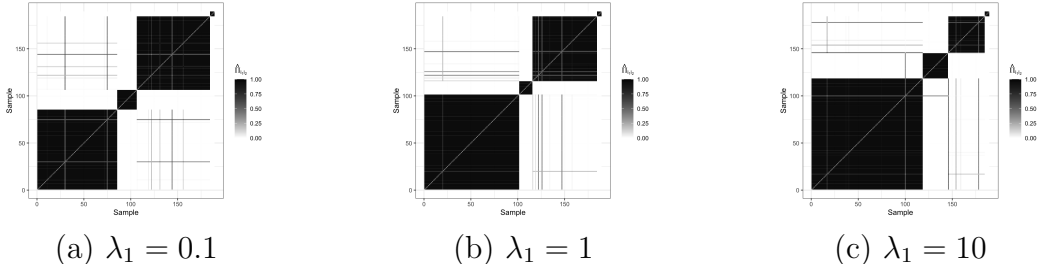
## Case study

The American Gut project [MBK15, MHD<sup>+</sup>18] has been launched as a collaboration between the Earth Microbiome Project [TSM<sup>+</sup>17] and the Human Food Project to build an open platform for citizen science microbiome research. It generates an open-source reference microbiome dataset through collecting mouth, skin and feces samples over a large human participants.

We apply LTNM to the July 2016 version of the fecal data from the AGP to construct enterotype for a specific group of samples where participants are diagnosed with inflammatory bowel disease (IBD). This specific version of the AGP datasets contains an OTU table of 27774 OTUs. Following the setup in [MM22], we focus on the top 75 OTUs which keep 2/3 of the total counts in each sample on average. In addition, we only keep the samples whose counts on the top 75 OTUs are over 500. As a result, there are 189 samples in the IBD group.

$\lambda_1$  is the hyperparameter putting penalty on the diagonal terms of  $\mathbf{\Omega}$ . The choice of  $\lambda_1$  in graphical lasso distribution is important since the clustering results of LTNM may be sensitive to it. In general, a large  $\lambda_1$  will result in a large variance terms and produce overly large clusters. On the other hand, a small  $\lambda_1$  will bring opposite effects. In order to test the sensitivity of LTNM with respect to  $\lambda_1$ , we try  $\lambda_1 = 0.1, 1, 10$  and  $\lambda_2 = 200$ . In each scenario, we fit LTNM with the same choices of priors and hyperparameters discussed in Section 2.3. We run the Gibbs sampler for 5000 iterations and discard the first 2500 iterations. Then we examine the clustering results.  $\mathcal{C}_{LS}$

is the output representative clustering for the IBD group dataset. It contains four clusters and is shown in Figure 4.1. Based on this figure, we realize the clustering process is quite stable, i.e. most samples have only been identified into one fixed cluster. On the other hand, as  $\lambda_1$  increases, an overly large cluster is more likely to be generated.



**Figure 4.1:** Estimated pairwise co-clustering probabilities with different  $\lambda_1$

We set  $\lambda_1 = 0.1$  to analyze the taxa selection results shown in Figure 4.2. Most internal nodes (41 out of 74) are relevant with clustering. In addition, some of the active nodes are close to the leaves but some are more "global" (have more descendant OTUS). This combination results in the latent clustering process is determined by a subset of internal nodes in a complicated manner.

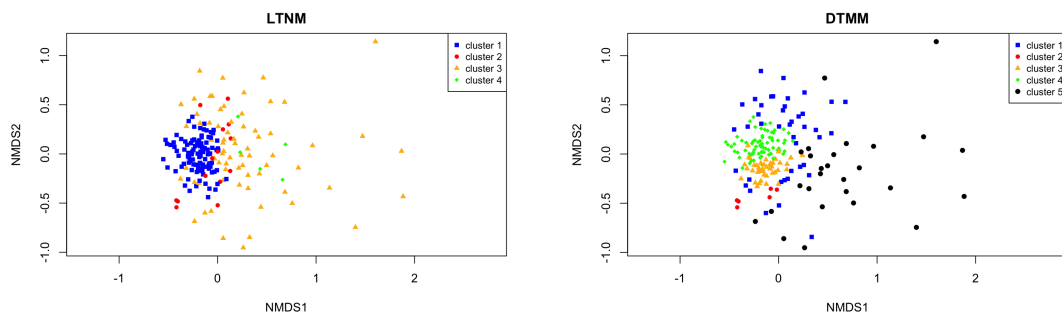
We are also interested in the "importance" of each OTU in the clustering process. Based on [MM22], we define the following metric to measure the importance:

$$\nu_j = \frac{\sum_{c \in \mathcal{C}_{LS}} n_c (\bar{x}_{cj} - \bar{x}_j)^2}{\sum_{c \in \mathcal{C}_{LS}} \sum_{c_i=c} (x_{ij} - \bar{x}_{cj})^2} \text{ for } j \in 1, \dots, M$$

where  $M$  is the number of OTUs,  $\bar{x}_j$  is the overall mean of  $x_{ij}$ , and  $\bar{x}_{cj}$  is the mean of  $x_{ij}$  for samples with  $c_i = c$ . Table 4.1 summarizes the information of top 10 OTUs in the clustering process. Bacteroides, Prevotella and Ruminococcus are three genera characterizing the three enterotypes in human gut microbial communities [AMR<sup>+</sup>11].



group by 2-D NMDS plot, which is a way to condense information from multidimensional data into a 2D representation. The closer two points are, the more similar the corresponding samples are. Figure 4.3 summarizes the clustering results of the LTNM and DTMM. Compared with LTNM, DTMM can identify small clusters within large clusters. This is expected since the Dirichlet process mixture can identify the small clusters that do not reflect the true data-generating process [MH13]. On the other hand, LTNM tends to generate large clusters.



**Figure 4.3:** Two-dimensional NMDS plots for the IBD group dataset

# Chapter 5

## Conclusion

We have introduced a Bayesian generative model for clustering the microbiome dataset. Adopting a multivariate Gaussian distribution for log-odds ratios, LTNM is flexible to capture the relations among taxa. In addition, we provide a way of choosing subsets of internal nodes that are active in clustering process. Moreover, there exists an easy and efficient sampling algorithm for the LTNM.

In real microbiome settings, when the number of OTUs is large, the dimension of the activation indicator parameter  $\gamma$  will also be large. As a result, a conventional Gibbs sampler usually causes mixing problem when updating a high-dimensional binary vector. A classical way of solving it is to integrate  $\psi$  out and make data directly influence  $\gamma$ . However, under the LTNM framework, the integral is not easily computed. Another method worth trying is stochastic search algorithm, which may lower the efficiency but improve the mixing. On the other hand, right now  $\gamma$  is only introduced to the mean vector of multivariate Gaussian distribution. Another direction worth exploring is modeling  $\Sigma$  by incorporating  $\gamma$  and adopting a reasonable prior for it. In addition, as we discussed above, the choices of  $\lambda_1$  and  $\lambda_2$  are important to the clustering results. A reasonable strategy of tuning hyperparameter is to use out-of-sample predictive performance to identify suitable choices of  $\lambda_1$  and  $\lambda_2$ .

# Appendix A

## Posterior sampling for $\log \alpha^*$

In the LTNM, we adopt a sparse symmetric Dirichlet prior on the weight distribution of an overfitting finite mixture distribution to deal with the difficulty in setting the number of clusters beforehand. Specifically, we have

$$\pi_1, \dots, \pi_K \sim \text{Dir}\left(\frac{\alpha}{K}, \dots, \frac{\alpha}{K}\right)$$

Now we let  $\alpha^* = \frac{\alpha}{K}$  and assign a Gamma prior to it, i.e.  $\alpha^* \sim \text{Gamma}(l_1, l_2)$ . Then we have

$$\begin{aligned} f(\alpha^* \mid \boldsymbol{\pi}, l_1, l_2) &\propto f(\alpha^* \mid l_1, l_2) \times f(\boldsymbol{\pi} \mid \alpha^*) \\ &= C(\alpha^*)^{l_1-1} e^{-l_2 \alpha^*} \frac{\Gamma(K \alpha^*)}{[\Gamma(\alpha^*)]^K} \prod_{i=1}^K \pi_i^{\alpha^*-1} \end{aligned} \quad (\text{A.1})$$

where  $C$  is a constant not involving  $\alpha^*$ . A random-walk Metropolis-Hastings can not be directly implemented since  $\alpha^* > 0$ . Hence, we let  $\eta = \log \alpha^*$  and choose a random-walk Metropolis-Hastings for posterior sampling. Based on the change of variable, we have

$$\begin{aligned} f(\eta \mid \boldsymbol{\pi}, l_1, l_2) &= f(\alpha^* \mid \boldsymbol{\pi}, l_1, l_2) \times \frac{d\alpha^*}{d\eta} \\ &= C[\exp(\eta)]^{l_1-1} e^{-l_2 \exp(\eta)} \frac{\Gamma(K \exp(\eta))}{[\Gamma(\exp(\eta))]^K} \prod_{i=1}^K \pi_i^{\exp(\eta)-1} \times \exp(\eta) \end{aligned} \quad (\text{A.2})$$

The proposals are formed as:

$$\mathbf{y}^{(t)} = \eta^{(t-1)} + \epsilon^{(t)} \quad (\text{A.3})$$

where  $\epsilon^{(t)} \sim \text{N}(0, \sigma^2)$  which is symmetric around 0. Therefore,

$$q(\mathbf{y}^{(t)} | \eta^{(t-1)}) = q(\epsilon^{(t)}) = q(-\epsilon^{(t)}) = q(\eta^{(t-1)} | \mathbf{y}^{(t)})$$

As a result, the acceptance probability is

$$p_t = \min\left\{\frac{f(\mathbf{y}^{(t)} | \boldsymbol{\pi}, l_1, l_2)}{f(\eta^{(t-1)} | \boldsymbol{\pi}, l_1, l_2)}, 1\right\} \quad (\text{A.4})$$

Draw  $u_t \sim \text{Unif}(0, 1)$  and if  $u_t < p_t$ , set  $\eta^{(t)} = \mathbf{y}^{(t)}$ . Otherwise  $\eta^{(t)} = \eta^{(t-1)}$ . At last, set  $(\alpha^*)^{(t)} = \exp(\eta^{(t)})$ .

# Appendix B

## Block Gibbs sampler for $\Omega$ and $\tau$

In this section, we briefly review the block Gibbs sampler proposed by [Wan12] for the data-augmented target distribution defined in Eq. (2.15). Note: we get rid of the cluster label for the following equations.

Without loss of generality, we can focus on updating the last row and column of  $\Omega$ . Define  $\Upsilon = (\tau_{ij})$  be a  $d \times d$  symmetric matrix with zeros in the diagonal entries and  $\tau$  in the upper diagonal entries. We can partition  $\Omega$ ,  $\mathbf{S}$  and  $\Upsilon$  as follows:

$$\Omega = \begin{pmatrix} \Omega_{11} & \omega_{12} \\ \omega'_{12} & \omega_{22} \end{pmatrix}, \mathbf{S} = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{s}_{12} \\ \mathbf{s}'_{12} & s_{22} \end{pmatrix}, \Upsilon = \begin{pmatrix} \Upsilon_{11} & \tau_{12} \\ \tau'_{12} & 0 \end{pmatrix}$$

From Eq. (2.15), the conditional distribution of the last column in  $\Omega$  is

$$\begin{aligned} p(\omega_{12}, \omega_{22} \mid \Omega_{11}, \mathbf{S}, \Upsilon, \lambda) &\propto \{|\Omega_{11}| - (\omega_{22} - \omega'_{12}\Omega_{11}^{-1}\omega_{12})\}^{\frac{n}{2}} \\ &\quad \times \exp\left\{-\frac{1}{2}(2\mathbf{s}'_{12}\omega_{12} + s_{22}\omega_{22})\right\} \\ &\quad \times \exp\left\{-\frac{1}{2}(\omega'_{12}\mathbf{D}_{\tau}^{-1}\omega_{12})\right\} \times \exp\left\{-\frac{\lambda}{2}\omega_{22}\right\} \\ &\propto (\omega_{22} - \omega'_{12}\Omega_{11}^{-1}\omega_{12})^{\frac{n}{2}} \\ &\quad \times \exp\left\{-\frac{1}{2}[\omega'_{12}\mathbf{D}_{\tau}^{-1}\omega_{12} + 2\mathbf{s}'_{12}\omega_{12} + (s_{22} + \lambda)\omega_{22}]\right\} \end{aligned}$$

where  $\mathbf{D}_{\tau} = \mathbf{diag}(\tau_{12})$ . Then we can make a change of variable

$$(\omega_{12}, \omega_{22}) \longrightarrow (\mathbf{D} = \omega_{12}, s = \omega_{22} - \omega'_{12}\Omega_{11}^{-1}\omega_{12})$$

The Jacobian is a constant not involving  $(\mathbf{D}, s)$ , hence

$$p(\mathbf{D}, s \mid \boldsymbol{\Omega}_{11}, \mathbf{S}, \boldsymbol{\Upsilon}, \lambda) \propto s^{\frac{n}{2}} \exp\left\{-\frac{s_{22} + \lambda}{2}s\right\} \\ \times \exp\left\{-\frac{1}{2}[\mathbf{D}'(\mathbf{D}_{\tau}^{-1} + (s_{22} + \lambda)\boldsymbol{\Omega}_{11}^{-1})\mathbf{D} + 2\mathbf{s}'_{12}\mathbf{D}]\right\}$$

Then we can conclude

$$(s, \mathbf{D}) \mid \boldsymbol{\Omega}_{11}, \mathbf{S}, \boldsymbol{\Upsilon}, \lambda \sim \text{Gamma}\left(\frac{n}{2} + 1, \frac{s_{22} + \lambda}{2}\right)\text{MVN}(-\mathbf{C}\mathbf{s}_{21}, \mathbf{C})$$

where  $\mathbf{C} = \{\mathbf{D}_{\tau}^{-1} + (s_{22} + \lambda)\boldsymbol{\Omega}_{11}^{-1}\}^{-1}$ . After drawing  $\mathbf{D}$  and  $s$ , we can transfer them back to  $\omega_{12}$  and  $\omega_{22}$ .

# Appendix C

## Full conditionals of other parameters

In this section, we provide details on the full conditionals of other parameters of the LTNM.

**The full conditional of  $\alpha$ :** Assume  $\Omega_k = \Sigma_k^{-1}$ , the full conditional of  $\alpha$  follows that

$$\begin{aligned}
 p(\alpha \mid \mathbf{c}, \boldsymbol{\psi}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\Omega}) &\propto p(\alpha) \prod_{i=1}^n p(\boldsymbol{\psi}_i \mid \boldsymbol{\beta}_{c_i}, \alpha, \boldsymbol{\Omega}_{c_i}, \boldsymbol{\gamma}) \\
 &\propto e^{-\frac{1}{2} \alpha^T (5 * \mathbf{I})^{-1} \alpha} \\
 &\quad \times e^{-\frac{1}{2} \sum_{i=1}^n (\boldsymbol{\psi}_i - \alpha \odot (\mathbf{1}_d - \boldsymbol{\gamma}) - \boldsymbol{\beta}_{c_i} \odot \boldsymbol{\gamma})^T \boldsymbol{\Omega}_{c_i} (\boldsymbol{\psi}_i - \alpha \odot (\mathbf{1}_d - \boldsymbol{\gamma}) - \boldsymbol{\beta}_{c_i} \odot \boldsymbol{\gamma})} \\
 &\sim \text{MVN}(\mathbf{m}_\alpha, \boldsymbol{\Lambda}_\alpha)
 \end{aligned}$$

where

$$\boldsymbol{\Lambda}_\alpha = ((5 * \mathbf{I})^{-1} + (\mathbf{1}_d - \boldsymbol{\gamma})(\mathbf{1}_d - \boldsymbol{\gamma})^T \odot \sum_{i=1}^n \boldsymbol{\Omega}_{c_i})^{-1} \quad (\text{C.1})$$

$$\mathbf{m}_\alpha = \boldsymbol{\Lambda}_\alpha \left\{ \left( \sum_{i=1}^n (\boldsymbol{\psi}_i^T - (\boldsymbol{\beta}_{c_i} \odot \boldsymbol{\gamma})^T) \boldsymbol{\Omega}_{c_i} \odot (\mathbf{1}_d - \boldsymbol{\gamma})^T \right) \right\}^T \quad (\text{C.2})$$

**The full conditional of  $\beta$ :** For  $k = 1, \dots, K$ ,

$$\begin{aligned}
p(\beta_k \mid \beta_{-k}, \mathbf{c}, \psi, \Omega, \alpha, \gamma) &\propto p(\beta_k) \prod_{i:c_i=k} p(\psi_i \mid \beta_k, \alpha, \Omega_k, \gamma) \\
&\propto e^{-\frac{1}{2} \beta_k^T (5 * \mathbf{I})^{-1} \beta_k} \\
&\quad \times e^{-\frac{1}{2} \sum_{i:c_i=k} (\psi_i - \alpha \odot (\mathbf{1}_d - \gamma) - \beta_k \odot \gamma)^T \Omega_k (\psi_i - \alpha \odot (\mathbf{1}_d - \gamma) - \beta_k \odot \gamma)} \\
&\sim \text{MVN}(\mathbf{m}_k, \Lambda_k)
\end{aligned}$$

where

$$\Lambda_k = ((5 * \mathbf{I})^{-1} + \gamma \gamma^T \odot |c_k| \Omega_k)^{-1} \quad (\text{C.3})$$

$$\mathbf{m}_k = \Lambda_k \left\{ \left( \sum_{i:c_i=k} \psi_i^T - |c_k| \alpha^T \odot (\mathbf{1}_d - \gamma)^T \right) \Omega_k \odot \gamma^T \right\}^T \quad (\text{C.4})$$

and  $|c_k|$  represents the number of samples in cluster  $k$ .

**The full conditional of  $\gamma$ :** For  $A \in \mathcal{I}$ ,  $p(\gamma(A) \mid \gamma(-A), \mathbf{c}, \alpha, \beta, \Omega, p, \psi)$  follows a Bernoulli distribution with success rate  $(O(A) + 1)^{-1}$

$$\begin{aligned}
O(A) &= \frac{p(\gamma(A) = 0 \mid \gamma(-A), \mathbf{c}, \alpha, \beta, \Omega, p, \psi)}{p(\gamma(A) = 1 \mid \gamma(-A), \mathbf{c}, \alpha, \beta, \Omega, p, \psi)} \\
&= \frac{p(\gamma(A) = 0 \mid p) p(\gamma(-A) \mid p) \prod_{i=1}^n p(\psi_i \mid \alpha, \beta_{c_i}, \Omega_{c_i}, \gamma(A) = 0, \gamma(-A))}{p(\gamma(A) = 1 \mid p) p(\gamma(-A) \mid p) \prod_{i=1}^n p(\psi_i \mid \alpha, \beta_{c_i}, \Omega_{c_i}, \gamma(A) = 1, \gamma(-A))} \\
&= \frac{(1-p) \times e^{-\frac{1}{2} \sum_{i=1}^n (\psi_i - \alpha \odot (\mathbf{1}_d - \gamma_0) - \beta_{c_i} \odot \gamma_0)^T \Omega_{c_i} (\psi_i - \alpha \odot (\mathbf{1}_d - \gamma_0) - \beta_{c_i} \odot \gamma_0)}}{p \times e^{-\frac{1}{2} \sum_{i=1}^n (\psi_i - \alpha \odot (\mathbf{1}_d - \gamma_1) - \beta_{c_i} \odot \gamma_1)^T \Omega_{c_i} (\psi_i - \alpha \odot (\mathbf{1}_d - \gamma_1) - \beta_{c_i} \odot \gamma_1)}}
\end{aligned} \quad (\text{C.5})$$

where  $\gamma_0 = (\gamma(A) = 0, \gamma(-A))$  and  $\gamma_1 = (\gamma(A) = 1, \gamma(-A))$ .

**The full conditional of  $\psi$ :** For  $i = 1, \dots, n$ ,

$$\begin{aligned}
p(\boldsymbol{\psi}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Omega}_i, \omega_i, Y_{iA}, Y_{iA_l}) &\propto p(\boldsymbol{\psi}_i \mid c_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Omega}_k) p(\omega_i, Y_{iA_l} \mid Y_{iA}, \boldsymbol{\psi}_i) \\
&\propto e^{-\frac{1}{2}(\boldsymbol{\psi}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Omega}_k (\boldsymbol{\psi}_i - \boldsymbol{\mu}_k)} \times e^{\boldsymbol{\kappa}_i^T \boldsymbol{\psi}_i - \frac{1}{2} \boldsymbol{\psi}_i^T \text{diag}(\omega_i) \boldsymbol{\psi}_i} \\
&\propto e^{-\frac{1}{2} \boldsymbol{\psi}_i^T (\boldsymbol{\Omega}_k + \text{diag}(\omega_i)) \boldsymbol{\psi}_i + \boldsymbol{\psi}_i^T (\boldsymbol{\Omega}_k \boldsymbol{\mu}_k + \boldsymbol{\kappa}_i)} \\
&\sim \text{MVN}(\boldsymbol{\Pi}_k (\boldsymbol{\Omega}_k \boldsymbol{\mu}_k + \boldsymbol{\kappa}_i), \boldsymbol{\Pi}_k)
\end{aligned}$$

where  $\boldsymbol{\Pi}_k = (\boldsymbol{\Omega}_k + \text{diag}(\omega_i))^{-1}$  and  $\boldsymbol{\mu}_k$  is defined in Eq. (2.2) and  $\boldsymbol{\kappa}_i = Y_{iA_l} - Y_{iA}/2$ . Here  $Y_{iA}$  is a vector of counts in all parent nodes of  $i$ -th sample tree and  $Y_{iA_l}$  is a vector of counts in all associated left-child nodes.

**The full conditional of  $c$ :** For  $i = 1, \dots, n$  and  $k = 1, \dots, K$ , we have

$$p(c_i = k \mid \mathbf{c}_{-i}, \boldsymbol{\pi}, \boldsymbol{\psi}_i, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\Omega}) \propto p(c_i = k \mid \boldsymbol{\pi}) p(\boldsymbol{\psi}_i \mid \boldsymbol{\alpha}, \boldsymbol{\beta}_k, \boldsymbol{\gamma}, \boldsymbol{\Omega}_k)$$

Then normalizing these  $K$  quantities to obtain the corresponding probabilities. We can update  $c_i$  through drawing a cluster label from a Categorical distribution with the calculated probabilities.

**The full conditional of  $p$ :** By the beta-binomial conjugacy,

$$p \mid a, b, \boldsymbol{\gamma} \sim \text{Beta}(a + \sum_{A \in \mathcal{I}} \gamma(A), b + d - \sum_{A \in \mathcal{I}} \gamma(A))$$

**The full conditional of  $\boldsymbol{\pi}$ :** By the Dirichlet-Categorical conjugacy,

$$p(\boldsymbol{\pi} \mid \boldsymbol{\alpha}^*, \mathbf{c}) \sim \text{Dir}(\zeta_1, \dots, \zeta_K) \text{ where } \zeta_j = (\boldsymbol{\alpha}^*) + \sum_{i=1}^n \mathbf{1}_{\{c_i=j\}}$$

**The full conditional of  $\omega$ :**  $\omega$  is Pólya-Gamma random variable introduced to restore the conjugacy. By definition, we have

$$\omega(A) \mid Y(A), \psi(A) \sim \text{PG}(Y(A), \psi(A))$$

# Appendix D

## Posterior sampling algorithm for the LTNM

Posterior sampling algorithm of the LTNM is summarized in Algorithm 1. To clarify the notations in the sampler,  $\boldsymbol{\omega}$  is a  $n \times d$  matrix containing Pólya-Gamma random variables and  $\boldsymbol{\Omega}_k$  represents a  $d \times d$  precision matrix for cluster  $k$ . For any cluster-specific square matrix  $\mathbf{A}_k$ ,  $A_{k(i,i)}$  is the  $i$ -th row and  $i$ -th column element in  $\mathbf{A}_k$ ,  $\mathbf{A}_{k(i,-i)}$  is the row vector formed by removing  $A_{k(i,i)}$  from the  $i$ -th row of  $\mathbf{A}_k$ ,  $\mathbf{A}_{k(-i,i)}$  is the column vector formed by removing  $A_{k(i,i)}$  from the  $i$ -th column of  $\mathbf{A}_k$ ,  $\mathbf{A}_{k(-i,-i)}$  is the submatrix of  $\mathbf{A}_k$  by removing the  $i$ -th row and  $i$ -th column. Let  $\boldsymbol{\Upsilon}_k^{(t)}$  be a  $d \times d$  symmetric matrix with zeros in the diagonal entries and  $\boldsymbol{\tau}_k^{(t)}$  in the upper diagonal entries representing the latent scale parameter matrix for cluster  $k$ , and define  $\mathbf{C}_{k(i)}^{(t)} = ((S_{k(i,i)}^{(t)} + \lambda_1)(\boldsymbol{\Omega}_{k(-i,-i)}^{(t)})^{-1} + \text{diag}(\boldsymbol{\Upsilon}_{k(i,-i)}^{(t)}))^{-1}$ .

---

**Algorithm 1** Posterior sampling algorithm for the LTNM
 

---

Initialize  $\omega^{(0)}, \psi^{(0)}, \Omega^{(0)}, \tau^{(0)}, \mathbf{c}^{(0)}, \alpha^{(0)}, \beta^{(0)}, \gamma^{(0)}, \pi^{(0)}, p^{(0)}, (\alpha^*)^{(0)}$   
**for**  $t = 1, \dots, T$  **do**  
 1. update  $\omega$  :  
**for**  $i = 1, \dots, n$  **do**  
**for**  $A \in \mathcal{I}$  **do**  
 Draw  $\omega_i^{(t)}(A) \sim \text{PG}(Y_i(A), \psi_i^{(t-1)}(A))$   
**end for**  
**end for**  
 2. update  $\psi$  :  
**for**  $i = 1, \dots, n$  **do**  
 Draw  $\psi_i^{(t)} \sim \text{MVN}((\Omega_{c_i}^{(t-1)} + \text{diag}(\omega_i^{(t)}))^{-1}(\Omega_{c_i}^{(t-1)} \mu_{c_i}^{(t-1)} + \kappa_i), (\Omega_{c_i}^{(t-1)} + \text{diag}(\omega_i^{(t)}))^{-1})$   
 where  $c_i = c_i^{(t-1)}, \mu_{c_i}^{(t-1)} = \alpha^{(t-1)} \odot (\mathbf{I}_d - \gamma^{(t-1)}) + \beta_{c_i}^{(t-1)} \odot \gamma^{(t-1)}$   
**end for**  
 3. update  $\alpha$  :  
 Draw  $\alpha^{(t)} \sim \text{MVN}(\mathbf{m}_\alpha, \Lambda_\alpha)$ , where  $\mathbf{m}_\alpha$  is defined in Eq. (C.2) and  $\Lambda_\alpha$  is defined in Eq. (C.1)  
 4. update  $\beta$  :  
**for**  $k = 1, \dots, K$  **do**  
 Draw  $\beta_k^{(t)} \sim \text{MVN}(\mathbf{m}_k, \Lambda_k)$ , where  $\mathbf{m}_k$  is defined in Eq. (C.4) and  $\Lambda_k$  is defined in Eq. (C.3)  
**end for**  
 5. update  $\Omega$  and  $\tau$  :  
**for**  $k = 1, \dots, K$  **do**  
**for**  $i = 1, \dots, d$  **do**  
 Draw  $s \sim \text{Gamma}(n_k/2 + 1, (S_{k(i,i)}^{(t-1)} + \lambda_1)/2)$   
 Draw  $\mathbf{D} \sim \text{MVN}(-\mathbf{C}_{k(i)}^{(t-1)} \mathbf{S}_{k(i,-i)}^{(t-1)}, \mathbf{C}_{k(i)}^{(t-1)})$   
 Update  $\Omega_{k(i,-i)}^{(t)} = \mathbf{D}, \Omega_{k(-i,i)}^{(t)} = \mathbf{D}^T, \Omega_{k(i,i)}^{(t)} = s + \mathbf{D}^T (\Omega_{k(-i,-i)}^{(t)})^{-1} \mathbf{D}$   
**end for**  
**for**  $1 \leq i < j \leq d$  **do**  
 Draw  $u_{ij} \sim \text{Inv-Gaussian}(\sqrt{\lambda_2^2 / (\Omega_{k(ij)}^{(t)})^2}, \lambda_2^2)$   
 Update  $\tau_{k(ij)}^{(t)} = \tau_{k(ji)}^{(t)} = 1/u_{ij}$   
**end for**  
**end for**  
 6. update  $\gamma$  :  
**for**  $A \in \mathcal{I}$  **do**  
 Draw  $\gamma^{(t)}(A) \sim \text{Bernoulli}((O(A) + 1)^{-1})$ , where  $O(A)$  is defined in Eq. (C.5)  
**end for**  
 7. update  $p$  :  
 Draw  $p^{(t)} \sim \text{Beta}(a + \sum_{A \in \mathcal{I}} \gamma^{(t)}(A), b + d - \sum_{A \in \mathcal{I}} \gamma^{(t)}(A))$   
 8. update  $\mathbf{c}$  :  
**for**  $i = 1, \dots, n$  **do**  
 Draw  $c_i^{(t-1)} \sim \text{Categorical}(\eta_1^{(t-1)}, \dots, \eta_K^{(t-1)})$ , where  $\eta_j^{(t-1)} = \frac{\pi_j^{(t-1)} p(\psi_i^{(t)} | \alpha^{(t)}, \beta_j^{(t)}, \gamma^{(t)}, \Omega_j^{(t)})}{\sum_{j'=1}^K \pi_{j'}^{(t-1)} p(\psi_i^{(t)} | \alpha^{(t)}, \beta_{j'}^{(t)}, \gamma^{(t)}, \Omega_{j'}^{(t)})}$   
**end for**  
 9. update  $\pi$  :  
 Draw  $\pi_1^{(t)}, \dots, \pi_K^{(t)} \sim \text{Dir}(\zeta_1^{(t-1)}, \dots, \zeta_K^{(t-1)})$ , where  $\zeta_j^{(t-1)} = (\alpha^*)^{(t-1)} + \sum_{i=1}^n \mathbf{1}_{\{c_i^{(t)}=j\}}$   
 10. update  $\alpha^*$  :  
 Update  $\alpha^*$  as described in A  
**end for**

---

## Bibliography

- [Ait82] J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):139–177, 1982.
- [AM74] D F Andrews and C L Mallows. Scale mixtures of normal distributions. *J. R. Stat. Soc.*, 36(1):99–102, September 1974.
- [AMR<sup>+</sup>11] Manimozhiyan Arumugam, MetaHIT Consortium (additional members), Jeroen Raes, Eric Pelletier, Denis Le Paslier, Takuji Yamada, Daniel R Mende, Gabriel R Fernandes, Julien Tap, Thomas Bruls, Jean-Michel Batto, Marcelo Bertalan, Natalia Borrueal, Francesc Casellas, Leyden Fernandez, Laurent Gautier, Torben Hansen, Masahira Hattori, Tetsuya Hayashi, Michiel Kleerebezem, Ken Kurokawa, Marion Leclerc, Florence Levenez, Chaysavanh Manichanh, H Bjørn Nielsen, Trine Nielsen, Nicolas Pons, Julie Poulain, Junjie Qin, Thomas Sicheritz-Ponten, Sebastian Tims, David Torrents, Edgardo Ugarte, Erwin G Zoetendal, Jun Wang, Francisco Guarner, Oluf Pedersen, Willem M de Vos, Søren Brunak, Joel Doré, Jean Weissenbach, S Dusko Ehrlich, and Peer Bork. Enterotypes of the human gut microbiome. *Nature*, 473(7346):174–180, May 2011.
- [And01] Marti J Anderson. A new method for non-parametric multivariate analysis of variance. *Austral Ecol.*, 26(1):32–46, February 2001.
- [BMBM<sup>+</sup>21] Francesco Beghini, Lauren J McIver, Aitor Blanco-Míguez, Leonard Dubois, Francesco Asnicar, Sagun Maharjan, Ana Mailyan, Paolo Manghi, Matthias Scholz, Andrew Maltez Thomas, Mireia Valles-Colomer, George Weingart, Yancong Zhang, Moreno Zolfo, Curtis Huttenhower, Eric A Franzosa, and Nicola Segata. Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with biobakery 3. *Elife*, 10, May 2021.
- [CMR<sup>+</sup>16] Benjamin J Callahan, Paul J McMurdie, Michael J Rosen, Andrew W Han, Amy Jo A Johnson, and Susan P Holmes. DADA2: High-resolution sample inference from illumina amplicon data. *Nat. Methods*, 13(7):581–583, July 2016.
- [Dah06] David B Dahl. Model-based clustering for expression data via a dirichlet process mixture model. In Kim-Anh Do and Peter Muller, editors, *Bayesian Inference for Gene Expression and Proteomics*, pages 201–218. Cambridge University Press, Cambridge, July 2006.

- [Den91] Samuel Y. Dennis. On the hyper-dirichlet type 1 and hyper-liouville distributions. *Communications in Statistics-theory and Methods*, 20:4069–4081, 1991.
- [EW95] Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *J. Am. Stat. Assoc.*, 90(430):577, June 1995.
- [GR01] Peter J Green and Sylvia Richardson. Modelling heterogeneity with and without the dirichlet process. *Scand. Stat. Theory Appl.*, 28(2):355–375, June 2001.
- [HHQ12] Ian Holmes, Keith Harris, and Christopher Quince. Dirichlet multinomial mixtures: generative models for microbial metagenomics. *PLoS One*, 7(2):e30126, February 2012.
- [IJ01] Hemant Ishwaran and Lancelot F James. Gibbs sampling methods for stick-breaking priors. *J. Am. Stat. Assoc.*, 96(453):161–173, March 2001.
- [Jac12] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytol.*, 11(2):37–50, February 1912.
- [KR90] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data*. Probability & Mathematical Statistics S. John Wiley & Sons, Nashville, TN, 99 edition, April 1990.
- [LK05] Catherine Lozupone and Rob Knight. UniFrac: a new phylogenetic method for comparing microbial communities. *Appl. Environ. Microbiol.*, 71(12):8228–8235, December 2005.
- [Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [Mac67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [MBK15] Daniel McDonald, Amanda Birmingham, and Rob Knight. Context and the human microbiome. *Microbiome*, 3(1):52, November 2015.
- [MH13] Jeffrey W Miller and Matthew T Harrison. A simple example of dirichlet process mixture inconsistency for the number of components. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

- [MH18] Jeffrey W Miller and Matthew T Harrison. Mixture models with a prior on the number of components. *J. Am. Stat. Assoc.*, 113(521):340–356, January 2018.
- [MHD<sup>+</sup>18] Daniel McDonald, Embriette Hyde, Justine W Debelius, James T Morton, Antonio Gonzalez, Gail Ackermann, Alexander A Aksenov, Bahar Behsaz, Caitriona Brennan, Yingfeng Chen, and et al. American gut: An open platform for citizen science microbiome research. *mSystems*, 3(3), June 2018.
- [MM22] Jialiang Mao and L I Ma. Dirichlet-tree multinomial mixtures for clustering microbiome compositions. *Ann. Appl. Stat.*, 16(3):1476–1499, September 2022.
- [MWFSG16] Gertraud Malsiner-Walli, Sylvia Frühwirth-Schnatter, and Bettina Grün. Model-based clustering based on sparse finite gaussian mixtures. *Stat. Comput.*, 26(1-2):303–324, 2016.
- [Nea00] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *J. Comput. Graph. Stat.*, 9(2):249, June 2000.
- [PSW13] Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *J. Am. Stat. Assoc.*, 108(504):1339–1349, December 2013.
- [RG18] Veronika Ročková and Edward I George. The spike-and-slab LASSO. *J. Am. Stat. Assoc.*, 113(521):431–444, January 2018.
- [SWMD17] Justin D Silverman, Alex D Washburne, Sayan Mukherjee, and Lawrence A David. A phylogenetic transform enhances analysis of compositional microbiota data. *eLife*, 6:e21887, feb 2017.
- [TSM<sup>+</sup>17] Luke R. Thompson, Jon G. Sanders, Daniel McDonald, Amnon Amir, Joshua Ladau, Kenneth J. Locey, Robert J. Prill, Anupriya Tripathi, Sean M. Gibbons, and et al. A communal catalogue reveals earth’s multiscale microbial diversity. *Nature*, 551(7681):457–463, Nov 2017.
- [Wan12] Hao Wang. Bayesian graphical lasso models and efficient posterior computation. *Bayesian Anal.*, 7(4):867–886, December 2012.
- [Wes87] Mike West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 09 1987.
- [WMM21] Zhuoqun Wang, Jialiang Mao, and Li Ma. Microbiome compositional analysis with logistic-tree normal models, 2021.

- [XCFL13] Fan Xia, Jun Chen, Wing Kam Fung, and Hongzhe Li. A logistic normal multinomial regression model for microbiome compositional data analysis. *Biometrics*, 69(4):1053–1063, 2013.
- [B17] Tarmo Äijö, Christian L Müller, and Richard Bonneau. Temporal probabilistic modeling of bacterial compositions derived from 16S rRNA sequencing. *Bioinformatics*, 34(3):372–380, 09 2017.