Journal of Statistical Research *https://doi.org/10.3329/jsr.v56i2.67469*
2022, Vol. 56, No. 2, pp. 185-216 *ISSN 0256 - 422 X*

PENALIZED LOGISTIC NORMAL MULTINOMIAL FACTOR ANALYZERS FOR HIGH DIMENSIONAL COMPOSITIONAL DATA

WANGSHU TU
School of Mathematics and Statistics, Carleton University, 1125 Colonel By Dr, Ottawa, Ontario, Canada K1S 5B6
Email: wangshu.tu@carleton.ca

SANJEENA SUBEDI*
School of Mathematics and Statistics, Carleton University, 1125 Colonel By Dr, Ottawa, Ontario, Canada K1S 5B6
Email: sanjeena.dang@carleton.ca

SUMMARY

Model-based clustering utilizes a finite mixture model to identify underlying patterns or clusters across samples. A finite mixture model is a convex combination of two or more distributions, where appropriate distributions are chosen depending on the type of the data. Recently, there has been a great interest in clustering human microbiome data. Microbiome data are compositional (yielding relative abundance) and are high-dimensional. Previously, a family of logistic normal multinomial factor analyzers (LNM-FA) for model-based clustering of high-dimensional microbiome data was proposed via a factor analyzer structure. This reduced the number of parameters and computation overhead compared to a traditional mixtures of logistic normal multinomial models. Here, we propose a penalized LNM-FA (PLNM-FA) model by utilizing lasso regularization to each entry of the loading matrix. This introduces further parsimony compared to LNM-FA and also estimates the number of latent factors simultaneously. Parameter estimation is done using a variational variant of the alternating expectation conditional maximization algorithm to maximize the penalized maximum likelihood. The performance of proposed algorithm is evaluated using simulation studies and real data.

Keywords and phrases: Model-based clustering, penalized factor analyzers, microbiome data, variational approximation

AMS Classification: 62-08, 62P10

1 Introduction

Cluster analysis is widely used to group observations into homogeneous subpopulations. A model-based clustering approach utilizes a finite mixture model, which assumes the data come from a finite collection of subpopulations or components where each subpopulation can be represented by a probability distribution. For a random variable $\mathbf{W}$, a G-component finite mixture density can be written as

$$f(\boldsymbol{w}_i|\boldsymbol{\Theta}) = \sum_{g=1}^{G} \pi_g f_g(\boldsymbol{w}_i|\boldsymbol{\Omega}_g),$$

* Corresponding author

where $\pi_g > 0$ is the mixing portion such that $\sum_{g=1}^{G} \pi_g = 1$, $f_g(\boldsymbol{w}_i|\boldsymbol{\Omega}_g)$ is the density function of g^{th} component, and $\boldsymbol{\Theta} = (\pi_1, \ldots, \pi_G, \boldsymbol{\Omega}_1, \ldots, \boldsymbol{\Omega}_G)$ represents the model parameters.

Compositional count data are routinely encountered in bioinformatics. In such data, the count of each sample is constrained by the total count, and the sample space can be represented by a simplex (Pawlowsky-Glahn et al., 2007) defined as

$$\Big\{\mathbf{x} = [x_1, x_2, \ldots, x_D] \mid x_i > 0, \sum_{i=1}^{D} x_i = \kappa\Big\}.$$

Thus, it is important to take into account κ while modelling $\mathbf{x}$ or while computing distance or dissimilarity measures involving $\mathbf{x}$. An example of such compositional count data is microbiome data. Microbiome data provides information on a dynamic ecosystem of microorganisms (bacteria, archaea, fungi, and viruses) that live in and on us. These microbes play a vital role in host-immune responses and host health (Metwally et al., 2018). Microbiome data obtained through next generation sequencing technology can be represented as a count matrix in which observations (i.e., rows) correspond to the abundance of various taxa of an individual/sample. Microbiome data is treated as compositional because the observed counts are restricted by the total counts in a sample (Gloor et al., 2017), and thus only yield relative abundance. Similar to RNA-seq data, count normalization is either performed on these datasets as the first step of the analysis or normalization is incorporated in the modelling paradigm. However, such approaches are less suitable for microbiome datasets because microbiome data are skewed and are highly sparse at lower taxonomic levels (Gloor et al., 2017).

Cluster analysis has been used to gain insight from microbiome data (Wu et al., 2011; Hotterbeekx et al., 2016; Taie et al., 2018; Abdel-Aziz et al., 2021). Several model-based clustering frameworks have been proposed for microbiome data (Holmes et al., 2012; Subedi et al., 2020; Fang and Subedi, 2020; Tu and Subedi, 2021). Holmes et al. (2012) proposed a Dirichlet-multinomial mixture model (DMM) to cluster microbiome data. Subedi et al. (2020) proposed mixtures of Dirichlet-multinomial regression models to cluster microbiome data, which can also model covariates. A Dirichlet-multinomial (DM) distribution that takes into account the compositional nature of the data (La Rosa et al., 2012; Chen and Li, 2013; Wadsworth et al., 2017; Koslovsky and Vannucci, 2020). However, the covariance of the microbiome data cannot be modelled adequately using a Dirichlet-multinomial distribution because of the limited number of parameters in the Dirichlet distribution (Xia et al., 2013). An additive logistic normal multinomial (LNM) model (Aitchison, 1982) was used by Xia et al. (2013) to model microbiome data. In an LNM model, the observed counts are modelled using a hierarchical structure where the observed counts conditional on the proportions are assumed to be multinomial. An additive log-ratio transformation is then used to transform the proportions from a simplex to an open real space and a Gaussian prior is imposed on this log-ratio transformed composition. Fang and Subedi (2020) developed mixtures of LNM models (LNM-MM) to cluster microbiome data.

While the LNM model provides flexibility in modelling the covariance structure, it can be highly parameterized for high dimensional data. Within a mixture model framework, a factor analyzer structure (Spearman, 1904) has been widely used to reduce the number of parameters in the component covariance matrix (McLachlan and Peel, 2000; McNicholas and Murphy, 2008; Subedi et al., 2013). In a mixture of factor analyzers (McLachlan and Peel, 2000), a K-dimensional vector Y can be modelled as

$$\mathbf{Y}_i = \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \mathbf{U}_{ig} + \epsilon_{ig},$$

where $\boldsymbol{U}_{ig} \sim N_q(0, \mathbf{I}_q)$ is a q-dimensional vector of latent factors, $\epsilon_{ig} \sim N_K(0, \boldsymbol{D}_g)$ is a K-dimensional vector of errors, $\boldsymbol{\mu}_g$ is $K \times 1$ mean vector of g^{th} component, $\boldsymbol{\Lambda}_g$ is $K \times q$ factor loadings matrix, $\boldsymbol{D}_g$ is diagonal matrix, and $\boldsymbol{U}_{ig} \perp \epsilon_{ig}$. Thus, the covariance matrix of $\mathbf{Y}$ can now be decomposed as $\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^T + \boldsymbol{D}_g$ with number of parameters $Kq - \frac{q(q-1)}{2} + K$ compared to a general covariance matrix with parameters $\frac{K(K-1)}{2}$. When $q << K$, the number of parameters in the covariance matrix can be greatly reduced. Recently, Tu and Subedi (2021) developed a mixture of logistic normal multinomial factor analyzers (LNM-FA) for high dimensional compositional count data by utilizing a factor analyzer structure within the LNM mixture model.

It is important to introduce sparsity in the covariance matrix as it provides information on possible independence among the variables. An early approach to introducing sparsity in the covariance matrix was by Dempster (1972) who suggested simplifying the covariance structure by setting some of the elements in the inverse covariance matrix to 0. In recent years, L_1 (lasso) regularization has been widely used to obtain a sparse estimation of the covariance matrix (Rothman et al., 2010; Bien and Tibshirani, 2011) or the inverse of the covariance matrix (Meinshausen and Bühlmann, 2006; Friedman et al., 2008; Banerjee et al., 2008). Alternately, Xie et al. (2010) proposed a penalized mixture factor analyzer (PMFA) model which introduces sparsity in the covariance matrix by using a group lasso regularization and shrinking an entire row of factor loading matrix $\boldsymbol{\Lambda}$ to 0. While the approach is effective for introducing sparsity, it can be restrictive for microbiome data. As $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{D}$, if the entire row of the loading matrix $\boldsymbol{\Lambda}$ is set to 0, the taxon will be modeled as uncorrelated to all other taxa. However, for microbiome data, this may be impractical as Taxon A can be uncorrelated with Taxon B but can be correlated with Taxon C. Furthermore, Xie et al. (2010) also used an L_1 penalty to shrink some entries of $\boldsymbol{\mu}$ to 0. In the context of compositional data, however, setting the entries in $\boldsymbol{\mu}$ (i.e., in the latent space) to 0 will impact the relative abundance estimation of other variables. Here, we introduce a penalized LNM-FA (PLNM-FA) mixture model, which only utilizes a lasso penalty to entries of the loading matrix $\boldsymbol{\Lambda}$ in LNM-FA. This approach provides a flexible and sparse estimation of the covariance structure. In Section 2, we provide the mathematical details of the proposed models. In Section 3 and 4, we illustrate our approach on both simulated and real datasets. In Section 5, we conclude with a summary of the paper.

2 Methodology

2.1 Additive logistic normal multinomial model

In the LNM model, conditional on the composition, the observed count vector $\mathbf{W}$ with $K+1$ taxa is modelled using a multinomial distribution such that

$$f(\mathbf{W}|\mathbf{P}) \propto p_1^{w_1} p_2^{w_2} \dots p_{K+1}^{w_{K+1}}.$$

An additive log-ratio transformation ϕ is then used to transform $\mathbf{P}$ from simplex to $\mathbf{Y}$ in an open real space

$$\mathbf{Y} = \phi(\mathbf{P}) = \left\{ \log\left(\frac{p_1}{p_{K+1}}\right), \dots, \log\left(\frac{p_K}{p_{K+1}}\right) \right\}, \tag{2.1}$$

where $\phi : (0,1)^{K+1} \rightarrow \mathbf{R}^K$ is a one-to-one function, and a Gaussian prior is imposed on this log-ratio transformed variable $\mathbf{Y}$. Using ϕ^{-1}, $\mathbf{P}$ can be written as

$$\mathbf{P} = \phi^{-1}(\mathbf{Y}) = \left\{ \frac{\exp(y_1)}{\sum_{k=1}^{K} \exp(y_k) + 1}, \cdots, \frac{\exp(y_K)}{\sum_{k=1}^{K} \exp(y_k) + 1}, \frac{1}{\sum_{k=1}^{K} \exp(y_k) + 1} \right\}. \tag{2.2}$$

Thus, the density of $\mathbf{W}|\mathbf{Y}$ can be also written as

$$f(\mathbf{W}|\mathbf{Y}) \propto \prod_{k=1}^{K} \left\{ \frac{\exp(y_k)}{\sum_{k=1}^{K} \exp(y_k) + 1} \right\}^{w_k} \left\{ \frac{1}{\sum_{k=1}^{K} \exp(y_k) + 1} \right\}^{w_{K+1}}.$$

Furthermore, $\mathbf{Y}$ is assumed to be a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The marginal density of $\mathbf{W}$ becomes

$$\begin{aligned} f(\boldsymbol{w} \,|\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \int_{\mathbf{R}^K} f(\boldsymbol{w}|\boldsymbol{y})\, f(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}_g)\, d\boldsymbol{y} \\ &\propto \int_{\mathbf{R}^K} \prod_{k=1}^{K+1} \left\{ \phi^{-1}(\boldsymbol{y})_k \right\}^{w_k} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{y} - \boldsymbol{\mu}_g)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \right\} d\boldsymbol{y}. \end{aligned}$$

One main challenge when using the LNM model is that the posterior distributions of the transformed variable do not have a closed form solution and thus, parameter estimation typically involves a Markov chain Monte Carlo (MCMC) approach (Xia et al., 2013; Äijö et al., 2018), which comes with heavy computational cost. Recently, Fang and Subedi (2020) developed a computationally efficient framework for parameter estimation for an additive logistic normal multinomial (LNM-MM) mixture model by utilizing variational Gaussian approximations (VGA; Wainwright et al., 2008). In VGA, a complex posterior distribution is approximated using computationally convenient Gaussian densities by minimizing the Kullback-Leibler (KL) divergence between the true and approximating Gaussian densities.

2.2 Mixture of penalized logistic normal multinomial factor analyzers

A G-component finite mixture of logistic normal multinomial models can be written as:

$$f(\boldsymbol{w} \mid \boldsymbol{\Theta}) = \sum_{g=1}^{G} \pi_g f_g(\boldsymbol{w}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g),$$

where $f_g(\boldsymbol{w}_i|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ is a logistic normal multinomial model with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, $\pi_g > 0$ is the mixing proportion such that $\sum_{g=1}^{G} \pi_g = 1$, and $\boldsymbol{\Theta} = (\pi_1, \ldots, \pi_G, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_G)$ denotes all the model parameters. In the context of clustering, the unobserved group membership of each observation is treated as missing data. We define a cluster membership indicator variable $\boldsymbol{Z}_i$ such that

$$Z_{ig} = \begin{cases} 1 & \text{observation } i \in g^{th} \text{group}, \\ 0 & \text{otherwise.} \end{cases}$$

Tu and Subedi (2021) developed a mixture of logistic normal multinomial factor analyzers by utilizing the factor analysis structure for the log-ratio transformed variable $\mathbf{Y}_i$ such that

$$\mathbf{Y}_i = \boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \mathbf{U}_{ig} + \epsilon_{ig},$$

where $\boldsymbol{U}_{ig} \sim N_q(0, \mathbf{I}_q)$ is a q-dimensional vector of latent factors, $\epsilon_{ig} \sim N_K(0, \boldsymbol{D}_g)$ is K-dimensional vector of errors, $\boldsymbol{\mu}_g$ is $K \times 1$ mean vector of g^{th} component, $\boldsymbol{\Lambda}_g$ is $K \times q$ factor loadings matrix, $\boldsymbol{D}_g$ is a diagonal matrix, and $\boldsymbol{U}_{ig} \perp \epsilon_{ig}$.

Using the observed data $(\mathbf{w}_1, \ldots, \mathbf{w}_n)$ and missing data $(\mathbf{z}_1, \ldots, \mathbf{z}_n)$, the complete data likelihood can be written as

$$L(\boldsymbol{\Omega}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \Big\{ \pi_g f_g(\boldsymbol{w}_i | \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \mathbf{D}_g) \Big\}^{z_{ig}},$$

where $\boldsymbol{\Omega} = (\pi_1, \ldots, \pi_G, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G, \boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_G, \mathbf{D}_1, \ldots, \mathbf{D}_G)$ denote all model parameters. The complete-data log-likelihood can be written as

$$l(\boldsymbol{\Omega}) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \Big\{ \log \pi_g + \log f(\boldsymbol{w}_i | \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{D}_g) \Big\},$$

where the marginal distribution of $\mathbf{W}$ is

$$\begin{aligned} f(\boldsymbol{w}_i \,|\boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{D}_g) &= \int_{\mathbf{R}^K} f(\boldsymbol{w}_i | \boldsymbol{y}_i)\, f(\boldsymbol{y}_i | \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^T + \boldsymbol{D}_g)\, d\boldsymbol{y} \\ &\propto \int_{\mathbf{R}^K} \prod_{k=1}^{K+1} \left\{ \phi^{-1}(\boldsymbol{y}_i)_k \right\}^{w_k} |\boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^T + \boldsymbol{D}_g|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (\boldsymbol{y}_i - \boldsymbol{\mu}_g)^\top (\boldsymbol{\Lambda}_g \boldsymbol{\Lambda}_g^T + \boldsymbol{D}_g)^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_g) \right\} d\boldsymbol{y}. \end{aligned}$$

The lasso penalty (Tibshirani, 1996) is a widely used regularization technique (L_1 regularization) that shrinks some parameters to be exactly zero. Here, we introduce the lasso regularization to loading matrix $\boldsymbol{\Lambda}_g$:

$$p(\boldsymbol{\Lambda}) = \frac{s}{1-s} \sum_{g=1}^{G} \sum_{i=1}^{K} \sum_{j=1}^{q} |[\boldsymbol{\Lambda}_g]_{ij}|,$$

where $|[\boldsymbol{\Lambda}_g]_{ij}|$ is the absolute value of the i^{th} row and j^{th} column of $\boldsymbol{\Lambda}_g$, and s is for the shrinkage/tuning parameter such that $0 < s < 1$. Similar to Xie et al. (2010), the penalized complete-data log likelihood can be written as

$$l(\boldsymbol{\Omega}) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \left\{ \log \pi_g + \log f(\boldsymbol{w}_i | \boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g, \boldsymbol{D}_g) \right\} - p(\boldsymbol{\Lambda}).$$

2.3 Parameter estimation

Here, similar to Tu and Subedi (2021), we develop a variational variant of the alternating expectation conditional maximization (AECM; Meng and Van Dyk, 1997) algorithm that uses different specification of the missing data at different cycles.

First Cycle

In the first cycle, we treat $\mathbf{Z}$ and $\mathbf{Y}$ as missing variables. The complete-data penalized log-likelihood using the marginal probability mass function of $\mathbf{W}$ can be written as

$$l_1(\boldsymbol{\Omega}) = \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\left\{\log \pi_g + \log \int f(\mathbf{w}_i|\mathbf{y}_i) f_g(\boldsymbol{y}_i|\boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^T + \boldsymbol{D}_g) d\boldsymbol{y}\right\} - p(\boldsymbol{\Lambda}).$$

Replacing the marginal density of $\mathbf{W}$ by the component specific ELBO $\tilde{F}(\boldsymbol{\mu}_g, \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^T + \boldsymbol{D}_g, \mathbf{m}_{ig}, \mathbf{V}_{ig})$ (see Appendix A.1 for detail), the variational Gaussian lower bound of complete-data log-likelihood becomes

$$\begin{aligned}\tilde{l}_1 = &\sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig}\left\{\log \pi_g - \left(\mathbf{1}_{(K+1)}^T \boldsymbol{w}_i\right)\left[\log\left(\mathbf{1}_{(K)}^T \exp\left(\mathbf{m}_{ig} + \frac{\text{diag}(\mathbf{V}_{ig})}{2}\right) + 1\right)\right]\right. \\ &+ C_i + \boldsymbol{w}^{*T}_i \mathbf{m}_{ig} + \frac{1}{2}\log|\mathbf{V}_{ig}| + \frac{K}{2} - \frac{1}{2}\log|\boldsymbol{\Sigma}_g| - \frac{1}{2}\text{tr}(\boldsymbol{\Sigma}_g^{-1}\mathbf{V}_{ig}) \\ &\left. -\frac{1}{2}(\boldsymbol{m}_{ig} - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}_g^{-1}(\boldsymbol{m}_{ig} - \boldsymbol{\mu}_g)\right\} - p(\boldsymbol{\Lambda}),\end{aligned}$$

where $\mathbf{1}_{(K)}$ stands for a column vector of 1's with dimension K, C_i stands for $\log \frac{\mathbf{1}^T \boldsymbol{w}_i!}{\prod_{k=1}^{K} \boldsymbol{w}_{ik}!}$, $\text{diag}(\mathbf{V}_{ig}) = (v^2_{ig,11}, v^2_{ig,22}, \ldots, v^2_{ig,KK})$ puts the diagonal elements of the $K \times K$ matrix $\mathbf{V}_{ig}$ into a K-dimensional vector, and $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^T + \boldsymbol{D}_g$. In this cycle, for the parameter updates in the $(t+1)^{th}$ iteration, the following steps are conducted:

Step 1: Update the variational Gaussian lower bound of the complete-data log-likelihood from the first cycle $\tilde{l}_1$ by updating $\mathbf{m}_{ig}$ and $\mathbf{V}_{ig}$. For updating $\mathbf{V}_{ig}^{(t+1)}$ and $\mathbf{m}_{ig}^{(t+1)}$, we use the Newton-Raphson method. We take the derivative with respect to $\boldsymbol{v}_{ig}^{(t+1)}$ and $\mathbf{m}_{ig}^{(t+1)}$ and find the solution to the following score function:

$$\frac{\partial \tilde{l}_1}{\partial \boldsymbol{v}_{ig}} = \boldsymbol{v}_{ig}^{(t)^{-1}} - \boldsymbol{v}_{ig}^{(t)}\text{diag}(\boldsymbol{\Sigma}_g^{(t)^{-1}}) - (\mathbf{1}_{(K+1)}^T \boldsymbol{w}_i)\boldsymbol{v}_{ig}^{(t)} \frac{\exp\left(\mathbf{m}_{ig}^{(t)} + \frac{(\boldsymbol{v}_{ig}^{(t)})^2}{2}\right)}{\mathbf{1}_{(K)}^T \exp\left(\mathbf{m}_{ig}^{(t)} + \frac{(\boldsymbol{v}_{ig}^{(t)})^2}{2}\right) + 1}.$$

$$\frac{\partial \tilde{l}_1}{\partial \mathbf{m}_{ig}} = \boldsymbol{w}_i^* - \boldsymbol{\Sigma}_g^{(t)^{-1}}(\mathbf{m}_{ig}^{(t)} - \boldsymbol{\mu}_g^{(t)}) - (\mathbf{1}_{(K+1)}^T \boldsymbol{w}_i) \frac{\exp\left(\mathbf{m}_{ig}^{(t)} + \frac{(\boldsymbol{v}_{ig}^{(t)})^2}{2}\right)}{\mathbf{1}_{(K)}^T \exp\left(\mathbf{m}_{ig}^{(t)} + \frac{(\boldsymbol{v}_{ig}^{(t)})^2}{2}\right) + 1}.$$

Step 2: Update the component indicator variable Z_{ig}. Conditional on the variational parameters $\mathbf{m}_{ig}^{(t+1)}$, $\mathbf{V}_{ig}^{(t+1)}$ and on $\boldsymbol{\mu}_g^{(t)}$, $\boldsymbol{\Lambda}_g^{(t)}$, and $\mathbf{D}_g^{(t)}$, we use an approximation of $E(Z_{ig}^{(t+1)})$ using the ELBO:

$$\hat{z}_{ig}^{(t+1)} = \frac{\pi_g^{(t)} \exp\left\{\tilde{F}(\boldsymbol{\mu}_g^{(t)}, \boldsymbol{\Lambda}_g^{(t)}\boldsymbol{\Lambda}_g^{(t)^T} + \boldsymbol{D}_g^{(t)}, \boldsymbol{m}_{ig}^{(t+1)}, \boldsymbol{V}_{ig}^{(t+1)})\right\}}{\sum_{g=1}^{G} \pi_g^{(t)} \exp\left\{\tilde{F}(\boldsymbol{\mu}_g^{(t)}, \boldsymbol{\Lambda}_g^{(t)}\boldsymbol{\Lambda}_g^{(t)^T} + \boldsymbol{D}_g^{(t)}, \boldsymbol{m}_{ig}^{(t+1)}, \boldsymbol{V}_{ig}^{(t+1)})\right\}}.$$

Step 3: Given the variational parameters and $\hat{z}_{ig}^{(t+1)}$, we update the parameters π_g and $\boldsymbol{\mu}_g$ as:

$$\hat{\pi}_g^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ig}^{(t+1)}}{n}, \quad \text{and} \quad \hat{\boldsymbol{\mu}}_g^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ig}^{(t+1)} \boldsymbol{m}_{ig}^{(t+1)}}{\sum_{i=1}^{n} \hat{z}_{ig}^{(t+1)}}.$$

Second Cycle

In the second cycle, we treat $\mathbf{Z}$, $\mathbf{Y}$ and $\boldsymbol{U}$ as the missing variables and the complete penalized log-likelihood using marginal probability mass function of $\mathbf{W}$ has the following form:

$$l_2(\boldsymbol{\Omega}) = \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \left\{ \log \pi_g + \log \left[\int f(\boldsymbol{w}_i|\boldsymbol{y}_i) f_g(\boldsymbol{y}_i|\boldsymbol{\mu}_g + \boldsymbol{\Lambda}_g \boldsymbol{u}_i, \boldsymbol{D}_g) f_g(\boldsymbol{u}_i|0, \mathbf{I}_q)\, d\boldsymbol{y}\, d\boldsymbol{u} \right] \right\} - p(\boldsymbol{\Lambda}).$$

Here, we assume $q(\boldsymbol{y}, \boldsymbol{u})$ can be factorized as $q(\boldsymbol{y}, \boldsymbol{u}) = q(\boldsymbol{y})q(\boldsymbol{u})$, $\mathbf{m}_{ig}$ and $\mathbf{V}_{ig}$ are the variational parameters of $q(\boldsymbol{y}_i)$ from first cycle, and $\tilde{\mathbf{m}}_{ig}$ and $\tilde{\mathbf{V}}_{ig}$ are the variational parameters of $q(\boldsymbol{u}_i)$. The approximate variational Gaussian lower bound of complete penalized data log-likelihood using $\tilde{F}_2$ becomes:

$$\begin{aligned}
\tilde{l}_2 = & \sum_{i=1}^{n}\sum_{g=1}^{G} z_{ig} \left\{ \log \pi_g - \left(1_{(K+1)}^T \boldsymbol{w}_i\right) \left[\log \left(1_{(K)}^T \exp\left(\mathbf{m}_{ig} + \frac{\text{diag}(\mathbf{V}_{ig})}{2} \right) + 1 \right) \right] \right. \\
& + C_i + \boldsymbol{w}_i^{*T} \mathbf{m}_{ig} + \frac{1}{2} \Bigg(\log|\mathbf{V}_{ig}| + \log|\tilde{\mathbf{V}}_g| + q + K - \log|\boldsymbol{D}_g| - \tilde{\mathbf{m}}_{ig}^T \tilde{\mathbf{m}}_{ig} - \text{tr}(\tilde{\mathbf{V}}_g) \\
& - \text{tr}(\boldsymbol{D}_g^{-1}(\boldsymbol{V}_{ig} + (\boldsymbol{m}_{ig} - \boldsymbol{\mu}_g)^T(\boldsymbol{m}_{ig} - \boldsymbol{\mu}_g))) + 2(\boldsymbol{m}_{ig} - \boldsymbol{\mu}_g)^T \boldsymbol{D}_g^{-1} \boldsymbol{\Lambda}_g \tilde{\mathbf{m}}_{ig} \\
& \left. -\tilde{\mathbf{m}}_{ig}^T \boldsymbol{\Lambda}_g^T \boldsymbol{D}_g^{-1} \boldsymbol{\Lambda}_g \tilde{\mathbf{m}}_{ig} - tr(\boldsymbol{\Lambda}_g^T \boldsymbol{D}_g^{-1} \boldsymbol{\Lambda}_g \tilde{\mathbf{V}}_g) \Bigg) \right\} - p(\boldsymbol{\Lambda}).
\end{aligned}$$

Details of the derivation of the lower bound $\tilde{F}_2$ is provided in Appendix A.2. In this cycle, for the parameter updates in the $(t+1)^{th}$ iteration, the following steps are conducted:

Step 1: Update the variational Gaussian lower bound of complete-data log-likelihood of the second cycle $\tilde{l}_2$ by updating $\tilde{\mathbf{m}}_{ig}^{(t+1)}$ and $\tilde{\mathbf{V}}_g^{(t+1)}$ as

$$\begin{aligned}
\tilde{\mathbf{m}}_{ig}^{(t+1)} &= ({\boldsymbol{\Lambda}_g^{(t)}}^T {\boldsymbol{D}_g^{(t)}}^{-1} \boldsymbol{\Lambda}_g^{(t)} + \mathbf{I}_q)^{-1} {\boldsymbol{\Lambda}_g^{(t)}}^T {\boldsymbol{D}_g^{(t)}}^{-1} (\boldsymbol{m}_{ig}^{(t+1)} - \boldsymbol{\mu}_g^{(t+1)}), \text{ and} \\
\tilde{\mathbf{V}}_g^{(t+1)} &= ({\boldsymbol{\Lambda}_g^{(t)}}^T {\boldsymbol{D}_g^{(t)}}^{-1} \boldsymbol{\Lambda}_g^{(t)} + \mathbf{I}_q)^{-1}.
\end{aligned}$$

Step 2: Update the group indicator variable $\mathbf{Z}$. Similar to the first cycle, we compute an approximation of $E(Z_{ig})$ using the ELBO from the second cycle:

$$\hat{z}_{ig}^{(t+1)} = \frac{\pi_g^{(t+1)} \exp\left\{ \tilde{F}_2(\boldsymbol{\mu}_g^{(t+1)}, \boldsymbol{\Lambda}_g^{(t)}, \boldsymbol{D}_g^{(t)}, \boldsymbol{m}_{ig}^{(t+1)}, \boldsymbol{V}_{ig}^{(t+1)}, \tilde{\mathbf{m}}_{ig}^{(t+1)}, \tilde{\mathbf{V}}_g^{(t+1)}) \right\}}{\sum_{g=1}^{G} \pi_g^{(t+1)} \exp\left\{ \tilde{F}_2(\boldsymbol{\mu}_g^{(t+1)}, \boldsymbol{\Lambda}_g^{(t)}, \boldsymbol{D}_g^{(t)}, \boldsymbol{m}_{ig}^{(t+1)}, \boldsymbol{V}_{ig}^{(t+1)}, \tilde{\mathbf{m}}_{ig}^{(t+1)}, \tilde{\mathbf{V}}_g^{(t+1)}) \right\}}.$$

Step 3: Update $\boldsymbol{D}_g^{(t+1)}$as

$$\hat{\boldsymbol{D}}_g^{(t+1)} = \text{diag}\Big\{\hat{\boldsymbol{\Sigma}}_g^{(t+1)} - 2\boldsymbol{\Lambda}_g^{(t)}(\boldsymbol{\Lambda}_g^{(t)^T}\boldsymbol{D}_g^{(t)^{-1}}\boldsymbol{\Lambda}_g^{(t)} + \mathbf{I}_q)^{-1}\boldsymbol{\Lambda}_g^{(t)^T}\boldsymbol{D}_g^{(t)^{-1}}\hat{\boldsymbol{S}}_g^{(t+1)} + \boldsymbol{\Lambda}_g^{(t)}\boldsymbol{\theta}_g^{(t+1)}\boldsymbol{\Lambda}_g^{(t)^T}\Big\}.$$

Step 4: When updating $\boldsymbol{\Lambda}_g$, we will update each entry of $\boldsymbol{\Lambda}_g$: $[\boldsymbol{\Lambda}_g]_{ij}$ with $i = 1\dots K$ and $j = 1\dots q$. Let $[\boldsymbol{\Lambda}_g]_{i(-j)}$ represents the entire i^{th} row of $\boldsymbol{\Lambda}_g$ but without the j^{th} entry. Since at $[\boldsymbol{\Lambda}_g]_{ij}$ is not differentiable at 0, we break it down into 2 cases:

- When $[\boldsymbol{\Lambda}_g]_{ij} > 0$, then the solution has to be greater than 0, which is

$$[\boldsymbol{\Lambda}_g]_{ij} = \frac{[\boldsymbol{S}_g\boldsymbol{\beta}_g^T]_{ij} - \frac{\frac{s}{1-s}[\boldsymbol{D}_g]_{ii}}{n_g} - [\boldsymbol{\Lambda}_g]_{i(-j)}[\boldsymbol{\theta}_g]_{(-j)j}}{[\boldsymbol{\theta}_g]_{jj}} > 0.$$

- When $[\boldsymbol{\Lambda}_g]_{ij} < 0$, then the solution has to be less than 0, which is

$$[\boldsymbol{\Lambda}_g]_{ij} = \frac{[\boldsymbol{S}_g\boldsymbol{\beta}_g^T]_{ij} + \frac{\frac{s}{1-s}[\boldsymbol{D}_g]_{ii}}{n_g} - [\boldsymbol{\Lambda}_g]_{i(-j)}[\boldsymbol{\theta}_g]_{(-j)j}}{[\boldsymbol{\theta}_g]_{jj}} < 0.$$

We select the $[\boldsymbol{\Lambda}_g]_{ij}$ that satisfies the above condition. When neither conditions are satisfied, we set $[\boldsymbol{\Lambda}_g]_{ij} = 0$. Here,

$$\begin{aligned}
\hat{\boldsymbol{S}}_g^{(t+1)} &= \frac{\sum_{i=1}^n z_{ig}^{(t+1)}(\boldsymbol{m}_{ig}^{(t+1)} - \boldsymbol{\mu}_g^{(t+1)})^T(\boldsymbol{m}_{ig}^{(t+1)} - \boldsymbol{\mu}_g^{(t+1)})}{\sum_{i=1}^n \hat{z}_{ig}^{(t+1)}},\\
\hat{\boldsymbol{\Sigma}}_g^{(t+1)} &= \frac{\sum_{i=1}^n z_{ig}^{(t+1)}\left[\boldsymbol{V}_{ig}^{(t+1)} + (\boldsymbol{m}_{ig}^{(t+1)} - \boldsymbol{\mu}_g^{(t+1)})(\boldsymbol{m}_{ig}^{(t+1)} - \boldsymbol{\mu}_g^{(t+1)})^\top\right]}{\sum_{i=1}^n \hat{z}_{ig}^{(t+1)}},\\
\boldsymbol{\theta}_g^{(t+1)} &= (\boldsymbol{\Lambda}_g^{(t)^T}\boldsymbol{D}_g^{(t)^{-1}}\boldsymbol{\Lambda}_g^{(t)} + \mathbf{I}_q)^{-1} + \boldsymbol{\beta}_g^{(t+1)}\boldsymbol{S}_g^{(t+1)}\boldsymbol{\beta}_g^{(t+1)^T},\\
\boldsymbol{\beta}_g^{(t+1)} &= (\boldsymbol{\Lambda}_g^{(t)^T}\boldsymbol{D}_g^{(t)^{-1}}\boldsymbol{\Lambda}_g^{(t)} + \mathbf{I}_q)^{-1}\boldsymbol{\Lambda}_g^{(t)^T}\boldsymbol{D}_g^{(t)^{-1}}, \quad \text{and} \quad \sum_{i=1}^n z_{ig} = n_g.
\end{aligned}$$

Overall, the variational AECM algorithm consists of the following steps:

1. Set the number of clusters: G and q, then initialize $\boldsymbol{\Lambda}_g$, $\boldsymbol{D}_g$, and z_{ig}.

2. First cycle:

Step 1: Approximate $\log f(\boldsymbol{w})$ by estimating $\mathbf{V}_{ig}$ and $\mathbf{m}_{ig}$.

Step 2: E-step: update z_{ig}.

Step 3: CM-step: update π_g and $\boldsymbol{\mu}_g$.

3. Second cycle:

Step 1: Approximate $\log f(\boldsymbol{w})$ by estimating $\tilde{\mathbf{V}}_{ig}$ and $\tilde{\mathbf{m}}_{ig}$.

Step 2: E-step: update z_{ig} again.

Step 3: CM-step: update $\boldsymbol{S}_g, \boldsymbol{\Sigma}_g, \boldsymbol{D}_g$, and $\boldsymbol{\Lambda}_g$.

4. Compute the penalized mixture density $\sum_{i=1}^{n} \log \sum_{g=1}^{G} \pi_g f(\boldsymbol{w}_i|\boldsymbol{\Omega}) - p(\boldsymbol{\Lambda})$ for current estimates. If converged, stop. Otherwise, go to step 2.

For high dimensional data, the factor loading matrix $\boldsymbol{\Lambda}_g$ tends to be most parameterized. Thus, to introduce further parsimony, we allow for imposing constraints on $\boldsymbol{\Lambda}_g$ to be equal or different across groups (see Table 1).

Table 1: Constrained and unconstrained covariance structures derived from PLNM-FA model.

Model	$\boldsymbol{\Lambda}_g$	Total number of parameters
Unconstrained	$\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}_g$	$\sum_{g=1}^{G} r_g + KG + G - 1 + KG$
Constrained	$\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$	$r + KG + G - 1 + KG$

In Table 1, the two models refer to whether or not constraints were imposed on the loading matrix, and $r_g = o_g^* - q_g^*(q_g^* - 1)/2$, where o_g^* denotes the number of nonzero entries in $\boldsymbol{\Lambda}_g$, and q_g^* is the number of nonzero columns in $\boldsymbol{\Lambda}_g$. For high dimensional data with small sample size, imposing such a constraint can reduce the number of parameters needed for the covariance matrices substantially. Details of the parameter estimation for the constrained PLNM-FA models are provided in the Appendix A.3.

2.4 Initialization and model selection

Let z_{ig}^*, π_g^*, $\boldsymbol{\mu}_g^*$, $\boldsymbol{D}_g^*$, $\boldsymbol{\Lambda}_g^*$, $\mathbf{m}_{ig}^*$ and $\mathbf{V}_{ig}^*$ be the initial values for Z_{ig}, π_g, $\boldsymbol{\mu}_g$, $\boldsymbol{D}_g$, $\boldsymbol{\Lambda}_g$, $\mathbf{m}_{ig}$ and $\mathbf{V}_{ig}$ respectively. Following Tu and Subedi (2021), we initialize our component indicator variable Z_{ig}, model parameters, and variational parameters as:

1. z_{ig}^* is initialized using the cluster membership obtained by fitting parsimonious Gaussian mixture models (PGMM; McNicholas and Murphy, 2008) to the transformed variable $\mathbf{Y}$ obtained using Equation (2.1). For computational purposes, any 0 in the $\mathbf{W}$ are replaced by 0.001 for initialization. The implementation of PGMM is available in R package "pgmm"(McNicholas et al., 2019).

2. Using this initial partition, $\boldsymbol{\mu}_g^*$ is initialized as the sample mean of the g^{th} cluster and π_g^* is initialized as the proportion of observations in the g^{th} cluster in this initial partition.

3. Similar to McNicholas and Murphy (2008), we estimate the sample covariance matrix $\boldsymbol{S}_g^*$ for each group and then used eigendecomposition of $\boldsymbol{S}_g^*$ to obtain $\boldsymbol{D}_g^*$ and $\boldsymbol{\Lambda}_g^*$. Suppose $\boldsymbol{\lambda}_g$ is a vector of the first q largest eigenvalues of $\mathbf{S}_g^*$ and the columns of $\mathbf{L}_g$ are the corresponding eigenvectors, then

$$\boldsymbol{\Lambda}_g^* = \mathbf{L}_g \boldsymbol{\lambda}_g^{\frac{1}{2}}, \quad \text{and} \quad \boldsymbol{D}_g^* = \text{diag}\{\boldsymbol{S}_g^* - \boldsymbol{\Lambda}_g^* \boldsymbol{\Lambda}_g^{*^T}\}.$$

4. As Newton-Raphson method is used to update the variational parameters, we need $\mathbf{m}^*$ and $\mathbf{V}^*$. For $\mathbf{m}^*$, we apply an additive log ratio transformation on the observed taxa compositions $\hat{\mathbf{p}}$ and set $\mathbf{m}^* = \phi(\hat{\mathbf{p}})$ using Equation (2.1). For $\mathbf{V}^*$, we use a diagonal matrix with all diagonal entries set to 0.1.

Note the variational parameters $\tilde{\mathbf{V}}_{ig}$ and $\tilde{\mathbf{m}}_{ig}$ are initialized using $\mathbf{m}^*$, $\boldsymbol{D}_g^*$, and $\boldsymbol{\Lambda}_g^*$ using their respective updating equation from step 1 of the second cycle.

In clustering, the number of components are generally unknown. In our context, the number of latent factors q as well as the best fitting model between constrained and unconstrained PLNM-FA are also unknown. Hence, we run both constrained and unconstrained models for a range of G and q and the best-fitting model is chosen using a model-selection criteria *a posteriori*. Here, we use the Bayesian Information Criterion (BIC; Schwarz, 1978) for model selection, which is considered to be consistent and efficient in practice under certain regularity conditions (Keribin, 2000; Fraley and Raftery, 1998). Mathematically,

$$\text{BIC} = -2l + \psi \log n,$$

where l is the log-likelihood, ψ is the number of free parameters, and n is the number of observations. The agreement between the true and observed classification can be assessed using the adjusted Rand index (ARI; Hubert and Arabie, 1985). ARI has a value of 1 under perfect agreement and expected value of 0 under random classification. As lasso regularization shrinks some of the entries in $\boldsymbol{\Lambda}_g$ to 0, one can always over specify the number of latent factors q, and the entries in the additional columns of $\boldsymbol{\Lambda}_g$ will all shrink to 0 if they are redundant, thus indicating a lower q is preferred. According to Lawley and Maxwell (1962), when K is very large, q should satisfy $(K-q)^2 > K+q$ which can be simplified as $q < K + \frac{1}{2} - \sqrt{2K + \frac{1}{4}}$. The shrinkage/tuning parameter selection is also done using BIC.

The optimal tuning parameter s is determined using a multi-stage grid search. We first apply LNM-FA to determine G and model (i.e. constrained or unconstrained). Although LNM-FA can also select the the number of latent factors q, it imposes a constraint that the number of latent factors are same for all clusters. The G and model type selected by LNM-FA is then used by PLNM-FA to select the tuning parameter s and number of latent factors q. Thus, we fit the PLNM-FA with a larger value for q (say initial q) and we impose L_1 penalization on the elements of the factor loading matrix. The effective number of latent factors for each cluster equals the difference of initial q and the number of columns of the factor loading matrix with all 0's. Thus, it allows us to select the optimal value for q for each cluster.

Regarding the choice of shrinkage tuning parameter s, a fine grid search over the entire range of s (i.e., between 0 and 1) for all datasets would be ideal but that can be computationally intensive. Since all the 100 datasets in one simulation setting are simulated using the same set of parameters, in each simulation setting, we first picked 1 out of 100 dataset, and ran a grid search over the full range of s between 0 and 1 but with only 10 values (i.e. 0, 0.11, 0.22 $\cdots$ 0.999). The s with the smallest BIC is rounded to the closest 0.05 and we do another grid search within ± 0.1 of the selected value. The s with the smallest BIC is again rounded to the closest 0.05. 15 equally spaced points within ± 0.05 of this newly selected value were then used for grid search for all 100 datasets in that simulation setting. The optimal s was chosen within this interval using BIC (model with the smallest BIC) for all 100 datasets. This approach is computationally efficient as it avoids doing fine grid search in regions that are far away from optimal s. This approach worked well in all simulation studies. For real dataset, a similar approach was used where a similar grid search was done for each dataset separately. Note that similar to the LNM-FA models, $\boldsymbol{\Lambda}_g$ is not identifiable because any orthonormal matrix $\boldsymbol{A}$ could satisfy $\boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g^T = \boldsymbol{\Lambda}_g\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{\Lambda}_g^T = \boldsymbol{\Lambda}_g^*\boldsymbol{\Lambda}_g^{*T}$.

3 Simulation Study

In this section, we use simulation studies to demonstrate the clustering performance and parameter recovery of the proposed PLNM-FA models. We first generated $\mathbf{Y}$ from a multivariate normal distribution, then transformed the data into composition $\boldsymbol{P}$ using an additive log ratio transformation. Count data are then generated using a multinomial distribution with composition $\boldsymbol{P}$ and the total count for each observation were generated from a uniform distribution $U\,[5000, 10000]$. Four sets of simulation studies were conducted, each consisting of 100 different datasets. The best fitting model and the pair of (q, s) was chosen using the BIC. For all simulation studies, we compared the performance of the proposed model with two other competing models: LNM-FA and DMM.

3.1 Simulation study 1

Here, we generated 100 eight-dimensional datasets, each of size $n = 500$ from the constrained model with $G = 3$, and $q = 3$. Figure 1 shows a visualization of the cluster structure in the latent space for one of the hundred datasets and Figure 2 shows the visualization of the relative abundance for observed count data from the same simulated data.

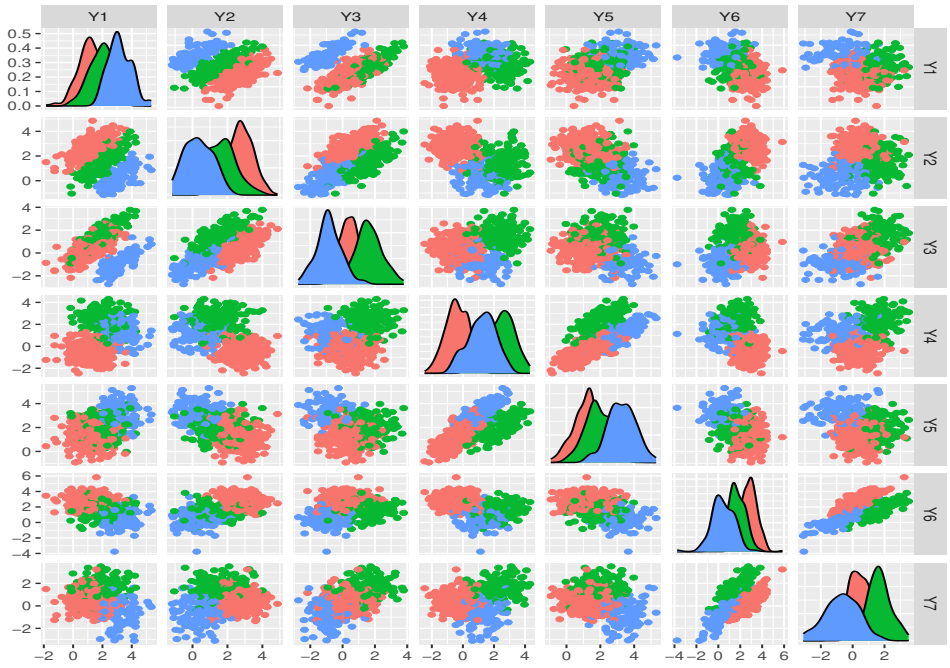


Figure 1: Pairwise scatter plot of latent variable $\mathbf{Y}$ from an example dataset from simulation study 1. The observations are colored using their true class label. For this dataset, an ARI of 1 was obtained by PLNM-FA.

We first ran the LNM-FA for both constrained and unconstrained models with $G = 1, \ldots, 4$ and $q = 1, \ldots, 4$, and then applied PLNM-FA for selecting tuning parameter s in range $0 < s < 1$ with $q = 4$. In

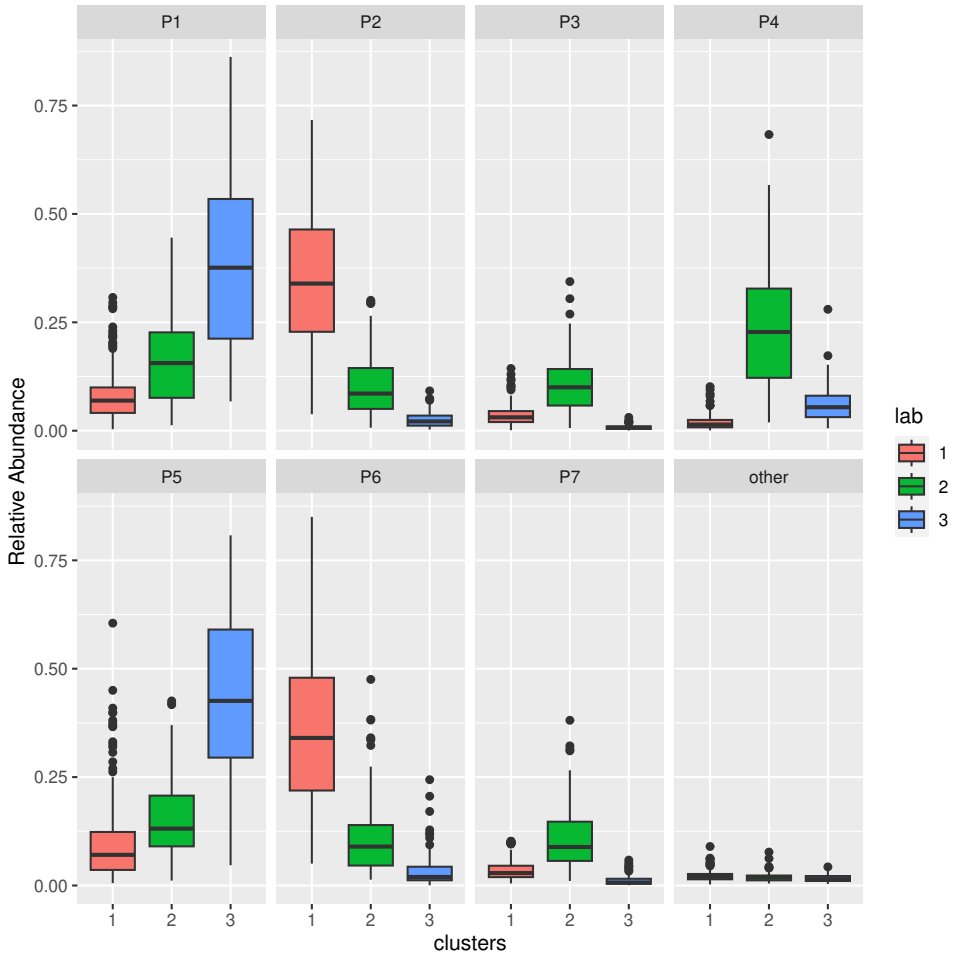


Figure 2: Boxplots of the relative abundances of the observed counts in each cluster from an example dataset from simulation study 1.

all 100 datasets, BIC selected a constrained model with $G = 3$ and the tuning parameter s had a mean = 0.97 (standard deviation [sd] 0.01). The average ARI of the final model selected by PLNM-FA was 1.00 (sd = 0.00). The ARI for both the LNM-FA and DMM were also 1 with sd of 0.00. The true values of the parameters π_g, $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}$ (i.e., common $\boldsymbol{\Sigma}$ across all groups) are provided in Table 2. The true values of $\boldsymbol{\Lambda}$ and $\boldsymbol{D}$ for $\boldsymbol{\Sigma}$ used to generate the datasets are provided in Appendix A.4. Note that the estimated values of $\boldsymbol{\Sigma}$ are biased as the estimate for $\boldsymbol{\Lambda}$ is biased due to the lasso regularization. The proportion of times the elements of $\boldsymbol{\Sigma}$ were estimated as non-zero over 100 estimations is also provided in Table 2. When the true values of the entries in $\boldsymbol{\Sigma}$ were non-zero, our approach always identified these entries as non-zero. The proportion of times when the truly null values of the entries in $\boldsymbol{\Sigma}$ were estimated as non-zero were close to 20%. However, the average of the estimated values was close to 0.

Table 2: Generating parameters along with the averages and standard errors of the estimated values of the parameters from the 100 datasets of simulation study 1. Note that for $\boldsymbol{\Sigma}$, we are providing an average across all components and all 100 datasets as a common $\boldsymbol{\Sigma}$ value was used to generate the data for all three components.

	True parameters	Average of estimated parameters (standard errors)
	Component 1($n_1 = 250$)	
$\boldsymbol{\mu}_1$	[1.2, 2.8, 0.4, -0.4, 1.2, 2.8, 0.4]	[1.19, 2.79, 0.40, -0.41, 1.18, 2.81, 0.40] (0.05, 0.05, 0.05, 0.06, 0.06, 0.05, 0.06)
π_1	0.5	0.5 (0.02)
	Component 2($n_2 = 150$)	
$\boldsymbol{\mu}_2$	[2.0, 1.6, 1.6, 2.4, 2.0, 1.6, 1.6]	[2.01, 1.60, 1.60, 2.39 , 1.99, 1.60 , 1.60] (0.07, 0.08, 0.07, 0.08, 0.08,0.06, 0.06)
π_2	0.3	0.3 (0.02)
	Component 3($n_3 = 100$)	
$\boldsymbol{\mu}_3$	[3.2, 0.4, -0.8, 1.2, 3.2, 0.4, -0.8]	[3.20 ,0.41, -0.79, 1.20, 3.19, 0.39, -0.81] (0.10, 0.09, 0.09, 0.08,0.08, 0.08, 0.08)
π_3	0.2	0.2 (0.02)
	The common covariance matrix for all three components and the average of estimated covariance matrix.	
$\boldsymbol{\Sigma}$	$\begin{bmatrix} 0.7 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0.6 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0.62 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.75 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.68 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.72 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.7 \end{bmatrix}$	$\begin{bmatrix} 0.46 & 0.38 & 0.37 & -0.00 & -0.00 & -0.00 & 0.00 \\ 0.38 & 0.58 & 0.37 & -0.00 & 0.00 & -0.00 & 0.00 \\ 0.37 & 0.37 & 0.47 & -0.00 & -0.00 & 0.00 & 0.00 \\ -0.00 & -0.00 & -0.00 & 0.65 & 0.39 & -0.00 & -0.00 \\ -0.00 & 0.00 & -0.00 & 0.39 & 0.59 & -0.00 & -0.00 \\ -0.00 & -0.00 & 0.00 & -0.00 & -0.00 & 0.60 & 0.40 \\ 0.00 & 0.00 & 0.00 & -0.00 & -0.00 & 0.40 & 0.50 \end{bmatrix}$
Proportion of times the elements of $\hat{\boldsymbol{\Sigma}}$ were non-zero		$\begin{pmatrix} 1.00 & 1.00 & 1.00 & 0.20 & 0.27 & 0.24 & 0.26 \\ 1.00 & 1.00 & 1.00 & 0.19 & 0.22 & 0.19 & 0.20 \\ 1.00 & 1.00 & 1.00 & 0.16 & 0.21 & 0.18 & 0.18 \\ 0.20 & 0.19 & 0.16 & 1.00 & 1.00 & 0.16 & 0.18 \\ 0.27 & 0.22 & 0.21 & 1.00 & 1.00 & 0.16 & 0.18 \\ 0.24 & 0.19 & 0.18 & 0.16 & 0.16 & 1.00 & 1.00 \\ 0.26 & 0.20 & 0.18 & 0.18 & 0.18 & 1.00 & 1.00 \end{pmatrix}$

3.2 Simulation study 2

We generated 100 eight-dimensional datasets, each of size $n = 500$, $G = 3$, and $q = 3$. Here, the parameters $\boldsymbol{\mu}_g$ and π_g were the same as simulation study 1 but $\boldsymbol{\Sigma}_g$ was different for different components. We first ran LNM-FA for both constrained and unconstrained models with $G = 1, \ldots, 4$ and $q = 1, \ldots, 4$, and then applied PLNM-FA with $q = 4$. In all 100 datasets, the BIC selected an unconstrained model with $G = 3$ and the tuning parameter s had a mean of 0.95 and sd 0.01. The average ARI of the selected PLNM-FA model is 1.00 (sd: 0.00). The LNM-FA and DMM both had an average ARI of 0.99 (sd: 0.002). The true values and estimations of the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}_g$ are provided in Table 3. Similar to simulation study 1, when the true values of the entries in $\boldsymbol{\Sigma}_g$ were non-zero, our approach always identified these entries as non-zero and the proportion of times when the true 0 values of the entries in $\boldsymbol{\Sigma}_g$ were estimated as non-zero are small.

Table 3: Generating parameters along with the averages and standard errors of the estimated values of the parameters from the 100 datasets from Simulation 2.

	True parameters	Average of estimated parameters (standard errors)
	Component 1($n_1 = 250$)	
π_1	0.5	0.5 (0.02)
$\boldsymbol{\mu}_1$	[1.2, 2.8, 0.4, -0.4, 1.2, 2.8, 0.4]	[1.19, 2.79, 0.40, -0.40, 1.19, 2.80, 0.40] (0.05, 0.04, 0.05, 0.06, 0.05, 0.05, 0.05)
$\boldsymbol{\Sigma}_1$	$\begin{bmatrix} 0.68 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.62 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.49 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.63 & 0.4 & 0.4 \\ 0 & 0 & 0 & 0 & 0.4 & 0.54 & 0.4 \\ 0 & 0 & 0 & 0 & 0.4 & 0.4 & 0.51 \end{bmatrix}$	$\begin{bmatrix} 0.54 & 0.26 & -0.00 & -0.00 & -0.00 & -0.00 & -0.00 \\ 0.26 & 0.39 & -0.00 & -0.00 & 0.00 & -0.00 & 0.00 \\ -0.00 & -0.00 & 0.50 & 0.27 & -0.00 & 0.00 & -0.00 \\ -0.00 & -0.00 & 0.27 & 0.39 & -0.00 & 0.00 & -0.00 \\ -0.00 & 0.00 & -0.00 & -0.00 & 0.48 & 0.26 & 0.25 \\ -0.00 & -0.00 & 0.00 & 0.00 & 0.26 & 0.41 & 0.27 \\ -0.00 & 0.00 & -0.00 & -0.00 & 0.25 & 0.27 & 0.38 \end{bmatrix}$
	Proportion of times the elements of $\hat{\boldsymbol{\Sigma}}_1$ is non-zero	$\begin{pmatrix} 1.00 & 1.00 & 0.04 & 0.14 & 0.10 & 0.20 & 0.20 \\ 1.00 & 1.00 & 0.11 & 0.19 & 0.11 & 0.20 & 0.22 \\ 0.04 & 0.11 & 1.00 & 1.00 & 0.10 & 0.16 & 0.19 \\ 0.14 & 0.19 & 1.00 & 1.00 & 0.12 & 0.23 & 0.22 \\ 0.10 & 0.11 & 0.10 & 0.12 & 1.00 & 1.00 & 1.00 \\ 0.20 & 0.20 & 0.16 & 0.23 & 1.00 & 1.00 & 1.00 \\ 0.20 & 0.22 & 0.19 & 0.22 & 1.00 & 1.00 & 1.00 \end{pmatrix}$
	Component 2($n_2 = 150$)	
π_2	0.3	0.3 (0.02)
$\boldsymbol{\mu}_2$	[2, 1.6, 1.6, 2.4, 2, 1.6, 1.6]	[2.00, 1.60, 1.60, 2.39, 1.99, 1.59, 1.60] (0.07, 0.08, 0.07, 0.08, 0.08, 0.05, 0.06)

$\boldsymbol{\Sigma}_2$	$\begin{bmatrix} 0.58 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0.7 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.74 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.68 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.58 \end{bmatrix}$	$\begin{bmatrix} 0.36 & 0.26 & 0.28 & -0.00 & -0.00 & -0.00 & -0.00 \\ 0.26 & 0.45 & 0.26 & -0.00 & -0.00 & -0.00 & -0.00 \\ 0.28 & 0.26 & 0.37 & -0.00 & -0.00 & -0.00 & -0.00 \\ -0.00 & -0.00 & -0.00 & 0.54 & 0.23 & 0.00 & -0.00 \\ -0.00 & -0.00 & -0.00 & 0.23 & 0.49 & 0.00 & -0.00 \\ -0.00 & -0.00 & -0.00 & 0.00 & 0.00 & 0.49 & 0.29 \\ -0.00 & -0.00 & -0.00 & -0.00 & -0.00 & 0.29 & 0.41 \end{bmatrix}$
	Proportion of times the elements of $\hat{\boldsymbol{\Sigma}}_2$ is non-zero	$\begin{pmatrix} 1.00 & 1.00 & 1.00 & 0.12 & 0.11 & 0.12 & 0.17 \\ 1.00 & 1.00 & 1.00 & 0.03 & 0.02 & 0.04 & 0.08 \\ 1.00 & 1.00 & 1.00 & 0.09 & 0.08 & 0.05 & 0.08 \\ 0.12 & 0.03 & 0.09 & 1.00 & 0.95 & 0.00 & 0.03 \\ 0.11 & 0.02 & 0.08 & 0.95 & 1.00 & 0.02 & 0.05 \\ 0.12 & 0.04 & 0.05 & 0.00 & 0.02 & 1.00 & 1.00 \\ 0.17 & 0.08 & 0.08 & 0.03 & 0.05 & 1.00 & 1.00 \end{pmatrix}$
Component 3($n_3 = 100$)		
π_3	0.2	0.2 (0.02)
$\boldsymbol{\mu}_3$	[3.2, 0.4, -0.8, 1.2, 3.2, 0.4, -0.8]	[3.20,0.40, -0.80, 1.20, 3.18, 0.39, -0.82] (0.10, 0.10, 0.09, 0.09,0.08, 0.09, 0.08)
$\boldsymbol{\Sigma}_3$	$\begin{bmatrix} 0.8 & 0.6 & 0.6 & 0.6 & 0 & 0 & 0 \\ 0.6 & 0.85 & 0.6 & 0.6 & 0 & 0 & 0 \\ 0.6 & 0.5 & 0.65 & 0.6 & 0 & 0 & 0 \\ 0.6 & 0.6 & 0.6 & 0.76 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.6 & 0.6 \\ 0 & 0 & 0 & 0 & 0.6 & 0.69 & 0.6 \\ 0 & 0 & 0 & 0 & 0.6 & 0.6 & 0.82 \end{bmatrix}$	$\begin{bmatrix} 0.43 & 0.22 & 0.23 & 0.24 & 0.00 & -0.00 & 0.00 \\ 0.22 & 0.48 & 0.22 & 0.22 & 0.00 & -0.00 & 0.00 \\ 0.23 & 0.22 & 0.33 & 0.23 & 0.00 & 0.00 & 0.00 \\ 0.24 & 0.22 & 0.23 & 0.40 & -0.00 & -0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -0.00 & 0.38 & 0.28 & 0.25 \\ -0.00 & -0.00 & 0.00 & -0.00 & 0.28 & 0.38 & 0.23 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.25 & 0.23 & 0.47 \end{bmatrix}$
	Proportion of times the elements of $\hat{\boldsymbol{\Sigma}}_3$ is non-zero	$\begin{pmatrix} 1.00 & 1.00 & 1.00 & 1.00 & 0.10 & 0.11 & 0.07 \\ 1.00 & 1.00 & 1.00 & 1.00 & 0.09 & 0.09 & 0.04 \\ 1.00 & 1.00 & 1.00 & 1.00 & 0.13 & 0.14 & 0.09 \\ 1.00 & 1.00 & 1.00 & 1.00 & 0.10 & 0.10 & 0.05 \\ 0.10 & 0.09 & 0.13 & 0.10 & 1.00 & 1.00 & 1.00 \\ 0.11 & 0.09 & 0.14 & 0.10 & 1.00 & 1.00 & 1.00 \\ 0.07 & 0.04 & 0.09 & 0.05 & 1.00 & 1.00 & 1.00 \end{pmatrix}$

3.3 Simulation study 3

Here, we study a scenario where different components have different number of latent factors and compare the performance with competing models. We generated 100 eight-dimensional datasets, each of size $n = 1000, G = 3$ and $q_1 = 5, q_2 = q_3 = 1$. Entries in $\boldsymbol{\mu}_g$ and factor loading matrix $\boldsymbol{\Lambda}_g$ were generated from a Gaussian distribution and elements of $\mathbf{D}_g$ were selected from a uniform distribution (details are provided in the Appendix A.4). Furthermore, a randomly selected 30, 4 and 4 entries in $\boldsymbol{\Lambda}_1$, $\boldsymbol{\Lambda}_2$, and $\boldsymbol{\Lambda}_3$, respectively, were set to be 0.

We first ran LNM-FA for both constrained and unconstrained models with $G = 1, \ldots, 4$ and $q = 1, \ldots, 5$, and then applied PLNM-FA with $q = 5$. In all 100 datasets, the BIC selected an unconstrained three-component model and the tuning parameter s had a mean 0.95 (sd = 0.01). The average ARI of the selected PLNM-FA model was 0.72 (sd = 0.11). On the other hand, in all 100 datasets, the LNM-FA selected a $G = 3, q = 1$ with an average ARI of 0.54 (sd = 0.10) and the DMM selected a four-component model with an average ARI with 0.00 (sd = 0). The estimated $\boldsymbol{\Sigma}_g$ along with the true $\boldsymbol{\Sigma}_g$ are provided in the Appendix A.5. Although not all true zeros in the covariance matrices were shrunk completely to zero, the average of the estimated values was close to 0 when the true values in the covariance matrix were 0. Furthermore, introducing sparsity in the covariance matrices via PLNM-FA showed an increase in the clustering performance compared to LNM-FA.

3.4 Simulation study 4

Here, we aim to demonstrate the performance of the proposed model for datasets with higher dimensions. We generated one hundred 51-dimensional datasets, each of size $n = 300$, $G = 3$, and $q_1 = q_2 = q_3 = 1$. Similar to simulation study 3, the entries in $\boldsymbol{\mu}_g$ and factor loading matrix $\boldsymbol{\Lambda}_g$ were generated from a Gaussian distribution and elements of $\mathbf{D}_g$ were selected from a uniform distribution (details are provided in the Appendix A.4). We then randomly selected 25 entries in each $\boldsymbol{\Lambda}_g$ to be 0.

We first ran LNM-FA for both the constrained and unconstrained models with $G = 1, \ldots, 4$ and $q = 1, \ldots, 4$, and then applied PLNM-FA with $q = 3$. The correct model (unconstrained three component model with $q = 1$) was selected for 70 out of 100 data sets. For 26 out of the 100 datasets, the BIC selected a two-component model (unconstrained with $q = 1$) and for 4 out of the 100 datasets, a four-component model (unconstrained with $q = 1$) was selected. The selected tuning parameter s had a mean 0.99 and sd 0.004. The average ARI of the final model selected by PLNM-FA was 0.99 (sd of 0.10). In this case, the LNM-FA had an average ARI of 0.97 (sd = 0.09), however, DMM only selected a three-component model twice resulting in an overall ARI with 0.16 (sd = 0.06). As the covariance matrix here is quite large, in Table 4, we provide the average proportion of times a true 0 in a covariance matrix was estimated to be non-zero for each component. While the numbers are higher compared to other simulation studies, in the high-dimensional setting, it is challenging to estimate the true covariance matrix due to small number of observations, which means it is harder to penalize those 0 entries as well.

4 Real data analysis

We applied our method to two publicly available microbiome datasets. To fit the most flexible model, here we only focused on the unconstrained model as it allows for different covariance structure among the

Table 4: Performance of PLNM-FA in simulation 3

	Component	Average proportion of times a true zero entry was estimated to be non-zero	ARI
Simulation study 4	1st	0.49 (0.24)	0.99 (0.10)
	2nd	0.43 (0.22)	
	3rd	0.25 (0.15)	

components.

FerrettiP Dataset: We applied our algorithm to the gut microbiome dataset `FerrettiP` (Ferretti et al., 2018) available in the R package `curatedMetagenomicData` (Pasolli et al., 2017). The study consisted of microbiome samples from 25 mother-infant pairs across multiple body sites from birth up to 4 months postpartum. As microbiome samples from different body sites can be different, here we focus our analysis on gut microbiome samples. We used only one time point (i.e., Day 1) for the newborns and conducted our analysis at the genus level. The resulting dataset comprises of 42 individuals (23 adults and 19 newborns) and 262 genera.

PehrssonE Dataset: We also applied our algorithm to the dataset `PehrssonE` (Pehrsson et al., 2016) available in the R package `curatedMetagenomicData` (Pasolli et al., 2017). Antibiotic-resistant infections costs lives of hundreds of thousands of individuals annually in the world (Pehrsson et al., 2016). Pehrsson et al. (2016) studied the bacterial community structure and resistance exchange networks using faecal samples from two low-income resource-limited Latin American habitats: 77 from peri-urban shanty-town in Lima, Peru (PER) and 114 from rural village in El Salvador (SLV). Here, we also worked at the genus level, resulting in a dataset comprising 191 individuals and 140 genera.

While information on 262 genera for `FerrettiP` and 140 genera for `PehrssonE` datasets are available, only a small proportion of these genera are different among the two groups (i.e., adults vs. infants for `FerrettiP` dataset, and PER vs. SLV for `PehrssonE` dataset). For both datasets, our first step is to identify group differentiating variables from the noise variables as having a large number of noise variables may negatively impact the clustering performance. As we are using real datasets for illustrating clustering performance when relevant group differentiating taxa are present, first we conducted differential abundance analysis using the R package ALDEx2 (Fernandes et al., 2013, 2014; Gloor et al., 2016) to identify genera that are different among the two groups. When this information is not available, one may run the cluster analysis using the top x most abundant taxa (where x is arbitrary number). However, note that the top x most abundant taxa may not be group differentiating for the condition of interest. We used the Welch's t-test option in ALDEx2 on the log-transformed counts for each genera and selected those genera with adjusted p-value less than 0.1 (after Benjamini-Hochberg correction). The numbers of differentially abundant genera for `FerrettiP`, and `Pehrsson` datasets are 8 and 21, respectively. To preserve the relative abundance, the remaining genera are aggregated in a category "Others", which is then used as the reference level for the additive log-ratio transformation.

We first ran LNM-FA for $G = 1, \ldots, 4$ for both datasets to select the optimal number of components. Given the two datasets' different dimensions, we set $q = 1, \ldots, 4$ for `FerrettiP` dataset and $q = 1, \ldots, 6$

for `Pehrsson` dataset. After determining G, we then fit the PLNM-FA to both datasets and used BIC to select the best fitting model. The BIC selected a $G = 2$ model with $s = 0.6357$ for `PehrssonP` dataset and $G = 2$ with $s = 0.2284$ model for `FerrettiP` dataset. We also ran LNM-FA, DMM, and LNM-MM on both datasets for $G = 1 \ldots 4$ and comparison of the clustering performance of these approaches is provided in Table 5.

The entries in $\mathbf{\Lambda}_g$ are the weights of the taxa contribution in forming the latent variables. Figure 3 shows the estimated $\mathbf{\Lambda}_g$ for `Feretti` dataset.

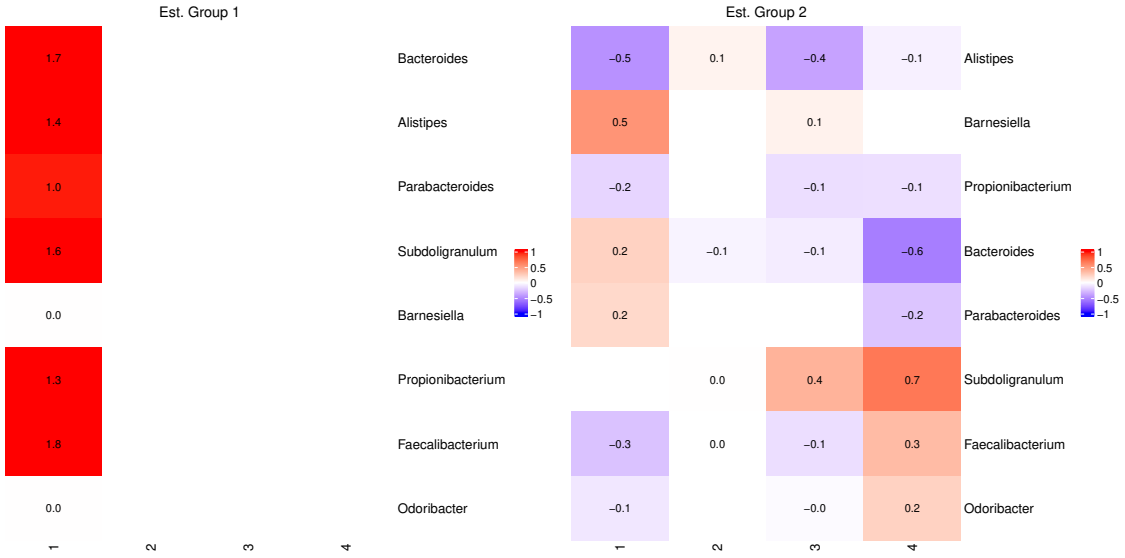


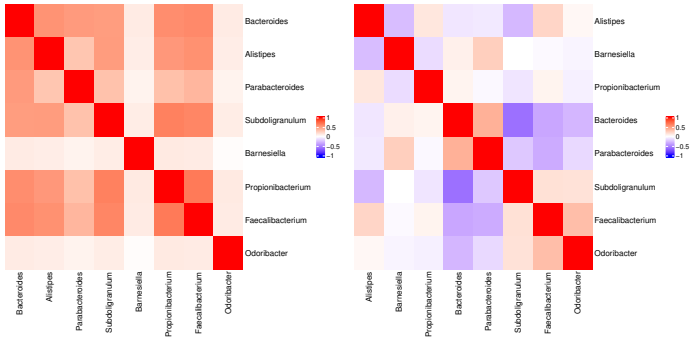
Figure 3: Heatmap of enties in Λ for `FerrettiP` dataset. The number stands for the value in $\mathbf{\Lambda}_g$, and red or blue colour stands for value being greater than 0 or less than 0, while a blank cell without any number means that estimated value was exactly 0.

When no taxa contributes to a latent variable, then all entries in that column are shrunk to 0, thus revealing that latent variable is not needed. Therefore, when the number of latent variables are over-specified, our approach also allows for the estimation of the number of latent factors q. For `Feretti` dataset, we fitted PLNM-FA with $q = 4$ for both components. However, the estimated loadings for $q = 2, 3,$ and, 4 are 0 for all taxa for the first component indicating that one latent factor would be sufficient for the first component while all four latent factors are needed to capture the covariance structure of the second component. The heatmaps of the observed and estimated correlations between the log-ratio transformed taxa from both clusters of `Feretti` dataset are shown in Figure 4. The correlation heatmap is calculated from $\mathbf{\Sigma}$. As seen in Figure 4, PLNM-FA recovers the underlying cluster correlation structure fairly well and when the observed correlation is close to 0, our approach estimates it to be 0.

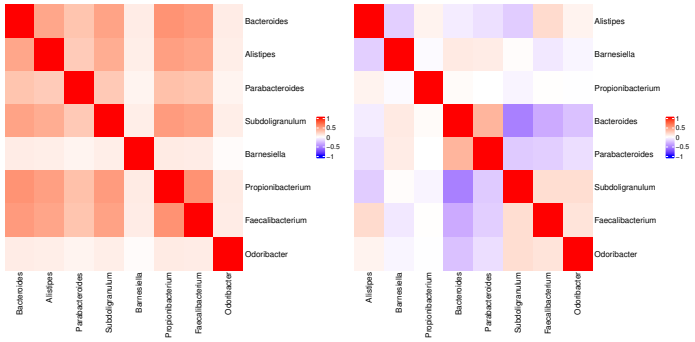
Similarly, although $q = 6$ was specified for all components for PLNM-FA for the `Pehrsson` dataset, several columns of estimated $\mathbf{\Lambda}$ were all 0's thus recommending that a lower q is sufficient for all components (i.e., $q = 2$, $q = 3$, and $q = 3$ for components 1, 2, and 3 were selected respectively). See Figure 5 for the visualization of the estimated $\mathbf{\Lambda}_g$ for `Pehrsson` dataset.

Table 5: Summary of the clustering performances on both real datasets using best fitting unconstrained model by PLNM-FA, LNM-FA, DMM and LNM-MM

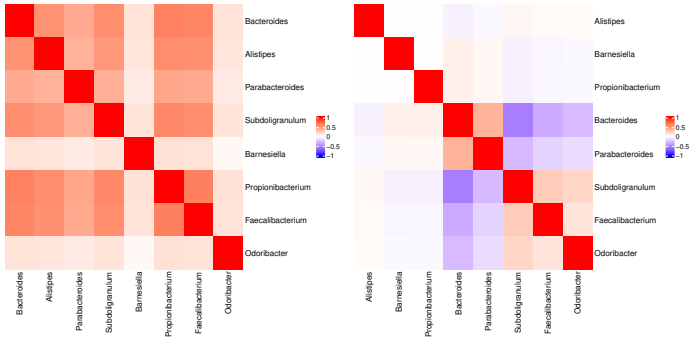
Data	Approach (Model)	Estimated G	Estimated q	Classification Table			ARI
					Infant	**Adult**	
	PLNM-FA	2	(1, 4)	Est. Group 1	18	1	**0.81**
				Est. Group 2	1	22	
					Infant	**Adult**	
`FerrettiP`	LNM-FA	2	1	Est. Group 1	18	1	**0.81**
				Est. Group 2	1	22	
	LNM-MM	-	-	-	-	-	-
					Infant	**Adult**	
	DMM	2	-	Est. Group 1	18	1	**0.81**
				Est. Group 2	1	22	
					PER	**SLV**	
	PLNM-FA	3	(2, 3, 3)	Est. Group 1	41	0	0.38
				Est. Group 2	33	34	
				Est. Group 3	3	80	
					PER	**SLV**	
`PehrssonE`	LNM-FA	3	1	Est. Group 1	42	0	**0.39**
				Est. Group 2	32	34	
				Est. Group 3	3	80	
	LNM-MM	-	-	-	-	-	-
					PER	**SLV**	
	DMM	4	-	Est. Group 1	34	0	0.33
				Est. Group 2	31	5	
				Est. Group 3	1	57	
				Est. Group 4	11	52	



(a) Observed correlation structures



(b) Recovered correlation structures using PLNM-FA



(c) Recovered correlation structures using LNM-FA

Figure 4: Heatmap of cluster-specific correlation structures in Infant and Adult for `FerrettiP` dataset

Note that not all taxa contribute to the latent factors and their weights vary from cluster to cluster. In cluster 1, the weights of *Buryrivibrio* and *Dorea* are 0 for both latent factors whereas in cluster 2, the weights of the *Escherichia*, *Desulfovibrio*, and *Bilophila* are 0 for all latent factors. The heatmaps of the observed and estimated correlations between the log-ratio transformed taxa from both clusters of `Pehrsson` dataset in Figure 6 show that our model recovered the underlying correlation structure fairly well.

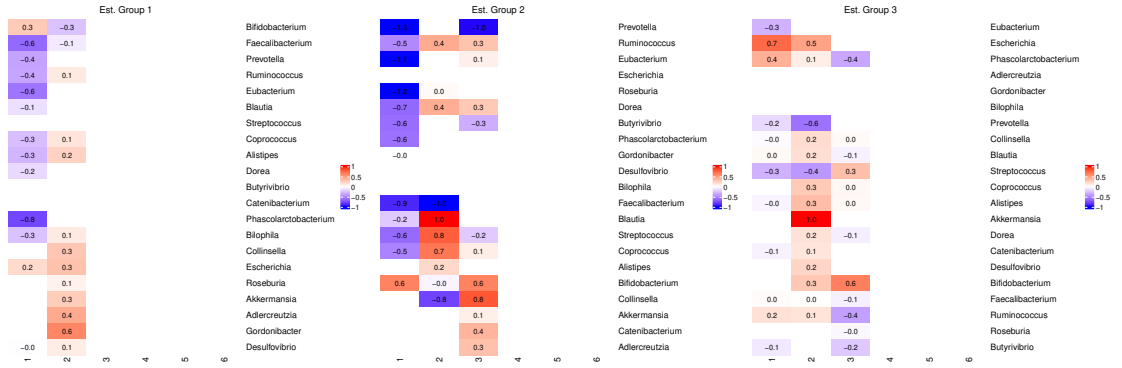
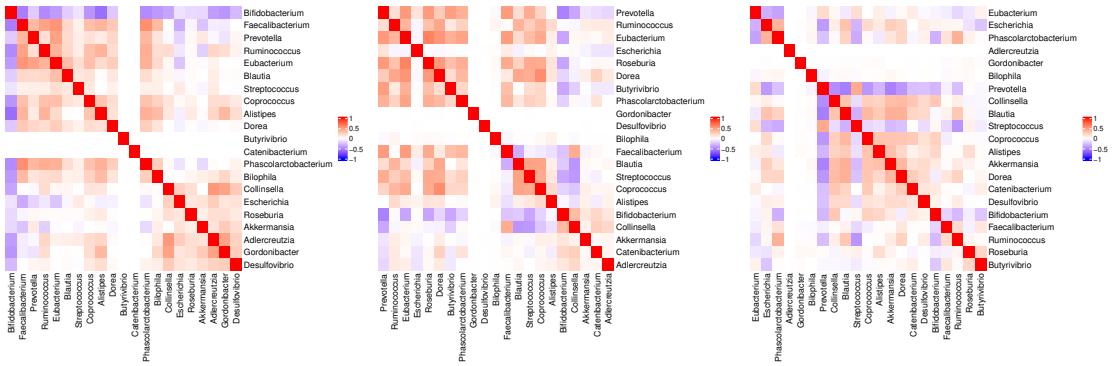


Figure 5: Heatmap of enties in Λ for `PehrssonE` dataset. The number stands for the value in $\mathbf{\Lambda}_g$, and red or blue colour stands for value greater than 0 or less than 0, while a blank cell without any number means that estimated value was exactly 0.

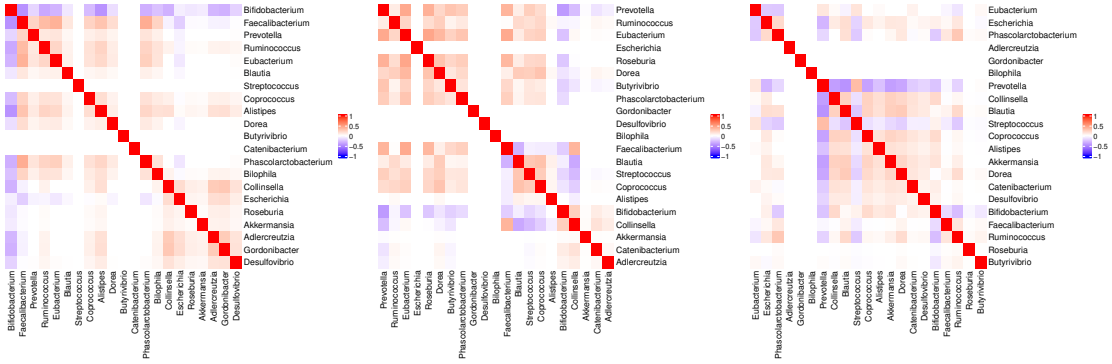
When the observed correlation is very small, our approach estimates the correlation as 0. For example, the estimated correlation between *Bilophila* and all other taxa are 0 in both Clusters 2 and 3. This agrees with the very small observed correlation between *Bilophila* and all other taxa in Cluster 2 and 3. However, in Cluster 1, *Bilophila* has a stronger positive or negative correlation with several taxa and therefore, the estimated correlations are non-zero. Thus, although in terms of clustering of the observations, both PLNM-FA and LNM-FA provide competitive performance on this dataset, the proposed PLNM-FA introduces sparsity in $\mathbf{\Lambda}_g$ and provides sparse estimations for $\mathbf{\Sigma}_g$ that could provide valuable insight into the underlying microbial community structure. Furthermore, as can be seen in Figures 4 and 6, allowing the number of latent variable (i.e., q) to vary among clusters and introducing sparsity on the elements of $\mathbf{\Lambda}_g$ resulted in a more accurate recovery of the underlying correlation structure compared to PLNM-FA for both real datasets.

5 Conclusion

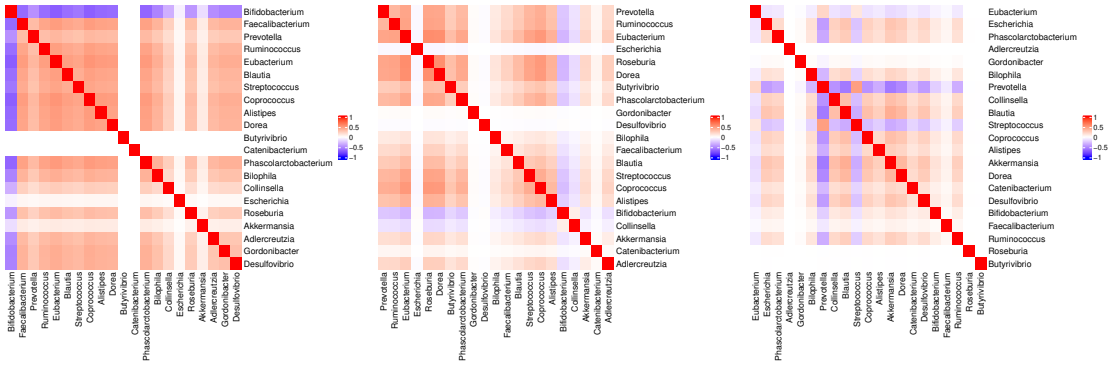
Here, we introduced a novel approach that provides a sparse covariance estimation of the covariance structure for mixtures of LNM-FA models via L_1 regularization of the elements of the factor loading matrix. Two models are proposed by imposing constraints on the factor loading matrix to be equal or different across groups. Due to the L_1 regularization of the loading matrix in factor analyzer structure, entries can be shrunk to zero when fitting data where observed covariances are close to 0, such that we can obtain a sparse estimation of $\mathbf{\Sigma}_g$. As shrinkage is applied to the entries, it allows for more flexibility in sparsity of the covariance structure. Through simulation studies, we demonstrated that our proposed approach provides excellent clustering performance and can recover sparsity in $\mathbf{\Sigma}_g$. In cases where the algorithm did not shrink entries to exactly 0, the estimated values were quite small and close to 0. Additionally, when the number of latent factors are over-specified, our approach can also estimate the optimal number of latent components by shrinking the weights of entire columns to 0. When a variable is not correlated with any other variables in the model, the entire row in $\mathbf{\Lambda}$ for that variable will shrink to 0, resulting in an estimated correlation of 0 for all variables. As compared to LNM-FA, which assumes the same number of latent components for each cluster and relies on BIC to select optimal q, PLNM-FA allows different q in each



(a) Observed correlation structures



(b) Recovered correlation structures using PLNM-FA



(c) Recovered correlation structures using LNM-FA

Figure 6: Heatmap of cluster-specific observed and recovered correlation structures for `PehrssonE` dataset

$\mathbf{\Lambda}$. Furthermore, introducing sparsity allows us to recover $\mathbf{\Sigma}_g$ more accurately than LNM-FA, and in some cases, improves the clustering performance as well. In our analysis, we utilized the BIC for selecting the optimal number of components, the model structure, and the tuning parameters. If the range of tuning

parameter is not wide and dense enough, BIC may not be able to optimal s. Some future direction will focus on investigating various approaches for tuning parameter selection. Our approach currently assumes that there is one tuning parameter for all components which can be restrictive, especially in the cases where the number of latent factors or the sparsity of Λ_g are different across different components. Considering different tuning parameter in different components can help in increasing the clustering performance and improvement in the recovery of the underlying correlation structure, but it is very computational intensive.

Acknowldgements

Funding This work is supported by NSERC Discovery Grant, Canada Research Chair Program, and Collaboration Grants for Mathematicians from Simons Foundation.

Conflict of interest None

Data sharing policy Datasets used in the manuscript are publicly available datasets. Details are provided in the manuscript.

References

Abdel-Aziz, M. I., Brinkman, P., Vijverberg, S. J., Neerincx, A. H., Riley, J. H., Bates, S., Hashimoto, S., Kermani, N. Z., Chung, K. F., Djukanovic, R., et al. (2021), "Sputum microbiome profiles identify severe asthma phenotypes of relative stability at 12-18 months," *Journal of Allergy and Clinical Immunology*, 147, 123–134.

Äijö, T., Müller, C. L., and Bonneau, R. (2018), "Temporal probabilistic modeling of bacterial compositions derived from 16S rRNA sequencing," *Bioinformatics*, 34, 372–380.

Aitchison, J. (1982), "The Statistical Analysis of Compositional Data," *Journal of the Royal Statistical Society: Series B (Methodological)*, 44, 139–160.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008), "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *The Journal of Machine Learning Research*, 9, 485–516.

Bien, J. and Tibshirani, R. (2011), "Sparse estimation of a covariance matrix," *Biometrika*, 98, 807–820.

Blei, D. and Lafferty, J. (2007), "A correlated topic model of Science," *The Annals of Applied Statistics*, 1, 17–35.

Chen, J. and Li, H. (2013), "Variable selection for sparse Dirichlet-multinomial regression with an application to microbiome data analysis," *The Annals of Applied Statistics*, 7.

Dempster, A. (1972), "Covariance selection," *Biometrics*, 28, 157–175.

Fang, Y. and Subedi, S. (2020), "Clustering microbiome data using mixtures of logistic normal multinomial models," ArXiv preprint arXiv:2011.06682.

Fernandes, A., Macklaim, J., Linn, T., Reid, G., and Gloor, G. (2013), "ANOVA-like differential gene expression analysis of single-organism and meta-RNA-seq," *PLoS ONE*, 8, e67019.

Fernandes, A. D., Reid, J. N., Macklaim, J. M., McMurrough, T. A., Edgell, D. R., and Gloor, G. B. (2014), "Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16S rRNA gene sequencing and selective growth experiments by compositional data analysis," *Microbiome*, 2, 1–13.

Ferretti, P., Pasolli, E., Tett, A., Asnicar, F., Gorfer, V., Fedi, S., Armanini, F., Truong, D. T., Manara, S., Zolfo, M., Beghini, F., Bertorelli, R., De Sanctis, V., Bariletti, I., Canto, R., Clementi, R., Cologna, M., Crifò, T., Cusumano, G., and Segata, N. (2018), "Mother-to-Infant Microbial Transmission from Different Body Sites Shapes the Developing Infant Gut Microbiome," *Cell Host & Microbe*, 24, 133–145.

Fraley, C. and Raftery, A. E. (1998), "How many clusters? Which clustering method? Answers via model-based cluster analysis," *The Computer Journal*, 41, 578–588.

Friedman, J., Hastie, T., and Tibshirani, R. (2008), "Sparse inverse covariance estimation with the graphical LASSO," *Biostatistics*, 9, 432–441.

Gloor, G., Macklaim, J., Pawlowsky-Glahn, V., and Egozcue, J. J. (2017), "Microbiome Datasets Are Compositional: And This Is Not Optional," *Frontiers in Microbiology*, 8, 2224.

Gloor, G. B., Macklaim, J. M., and Fernandes, A. D. (2016), "Displaying variation in large datasets: plotting a visual summary of effect sizes," *Journal of Computational and Graphical Statistics*, 25, 971–979.

Holmes, I., Harris, K., and Quince, C. (2012), "Dirichlet Multinomial Mixtures: Generative Models for Microbial Metagenomics," *PloS ONE*, 7, e30126.

Hotterbeekx, A., Xavier, B. B., Bielen, K., Lammens, C., Moons, P., Schepens, T., Ieven, M., Jorens, P. G., Goossens, H., Kumar-Singh, S., et al. (2016), "The endotracheal tube microbiome associated with *Pseudomonas aeruginosa* or *Staphylococcus epidermidis*," *Scientific Reports*, 6, 36507.

Hubert, L. and Arabie, P. (1985), "Comparing partitions," *Journal of Classification*, 2, 193–218.

Keribin, C. (2000), "Consistent Estimation of the Order of Mixture Models," *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 62, 49–66.

Koslovsky, M. D. and Vannucci, M. (2020), "MicroBVS: Dirichlet-tree multinomial regression models with Bayesian variable selection-an R package," *BMC Bioinformatics*, 21, 1–10.

La Rosa, P. S., Brooks, J. P., Deych, E., Boone, E. L., Edwards, D. J., Wang, Q., Sodergren, E., Weinstock, G., and Shannon, W. D. (2012), "Hypothesis testing and power calculations for taxonomic-based human microbiome data," *PLOS ONE*, 7, e52078.

Lawley, D. N. and Maxwell, A. E. (1962), "Factor Analysis as a Statistical Method," *Journal of the Royal Statistical Society. Series D (The Statistician)*, 12, 209–229.

McLachlan, G. and Peel, D. (2000), "Mixtures of factor analyzers," in *In Proceedings of the Seventeenth International Conference on Machine Learning*.

McNicholas, P. D., ElSherbiny, A., McDaid, A. F., and Murphy, T. B. (2019), *pgmm: Parsimonious Gaussian mixture models*, r package version 1.2.4.

McNicholas, P. D. and Murphy, T. B. (2008), "Parsimonious Gaussian mixture models," *Statistics and Computing*, 18, 285–296.

Meinshausen, N. and Bühlmann, P. (2006), "High dimensional graphs and variable selection with the LASSO," *The Annals of Statistics*, 34, 1436–1462.

Meng, X.-L. and Van Dyk, D. (1997), "The EM algorithm—an old folk-song sung to a fast new tune," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59, 511–567.

Metwally, A. A., Aldirawi, H., and Yang, J. (2018), "A review on probabilistic models used in microbiome studies." *Communications in Information and Systems*, 18, 173–191.

Pasolli, E., Schiffer, L., Manghi, P., Renson, A., Obenchain, V., Truong, D. T., Beghini, F., Malik, F., Ramos, M., Dowd, J. B., Huttenhower, C., Morgan, M., Segata, N., and Waldron, L. (2017), "Accessible, curated metagenomic data through ExperimentHub," *Nature Methods*, 14, 1023–1024.

Pawlowsky-Glahn, V., Egozcue, J. J., and Tolosana-Delgado, R. (2007), "Lecture Notes on Compositional Data Analysis," .

Pehrsson, E., Tsukayama, P., Patel, S., Mejía-Bautista, M., Sosa-Soto, G., Navarrete, K., Calderon, M., Cabrera, L., Hoyos-Arango, W., Bertoli, M., Berg, D., Gilman, R., and Dantas, G. (2016), "Interconnected microbiomes and resistomes in low-income human habitats," *Nature*, 533, 212–216.

Rothman, A. J., Levina, E., and Zhu, J. (2010), "A new approach to Cholesky-based covariance regularization in high dimensions," *Biometrika*, 97, 539–550.

Schwarz, G. (1978), "Estimating the dimension of a model," *The Annals of Statistics*, 6, 461–464.

Spearman, C. (1904), "The proof and measurement of association between two things," *The American Journal of Psychology*, 15, 72–101.

Subedi, S., Neish, D., Bak, S., and Feng, Z. (2020), "Cluster analysis of microbiome data via mixtures of Dirichlet-multinomial regression models," *Journal of Royal Statistical Society: Series C*, 69, 1163–1187.

Subedi, S., Punzo, A., Ingrassia, S., and Mcnicholas, P. D. (2013), "Clustering and classification via cluster-weighted factor analyzers," *Advances in Data Analysis and Classification*, 7, 5–40.

Taie, W. S., Omar, Y., and Badr, A. (2018), "Clustering of human intestine microbiomes with k-means," in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, IEEE, pp. 1–6.

Tibshirani, R. (1996), "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, 58, 267–288.

Tu, W. and Subedi, S. (2021), "Logistic Normal Multinomial Factor Analyzers for Clustering Microbiome Data," ArXiv preprint arXiv:2101.01871.

Wadsworth, W. D., Argiento, R., Guindani, M., Galloway-Pena, J., Shelburne, S. A., and Vannucci, M. (2017), "An integrative Bayesian Dirichlet-multinomial regression model for the analysis of taxonomic abundances in microbiome data," *BMC Bioinformatics*, 18, 1–12.

Wainwright, M. J., Jordan, M. I., et al. (2008), "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, 1, 1–305.

Wu, G. D., Chen, J., Hoffmann, C., Bittinger, K., Chen, Y.-Y., Keilbaugh, S. A., Bewtra, M., Knights, D., Walters, W. A., Knight, R., et al. (2011), "Linking long-term dietary patterns with gut microbial enterotypes," *Science*, 334, 105–108.

Xia, F., Chen, J., Fung, W. K., and Li, H. (2013), "A logistic normal multinomial regression model for microbiome compositional data analysis," *Biometrics*, 69, 1053—1063.

Xie, B., Pan, W., and Shen, X. (2010), "Penalized mixtures of factor analyzers with application to clustering high-dimensional microarray data," *Bioinformatics*, 26, 501–508.

Received: June 9, 2022

Accepted: February 8, 2023

A Mathematical Details

Fang and Subedi (2020) first introduced the ELBO for the LNM mixture model and Tu and Subedi (2021) simplified this further. While the ELBO for the LNM model is the same as Tu and Subedi (2021), we are providing the details here in the Appendix for completeness.

A.1 ELBO for LNM model

First, we decompose $F(q(\boldsymbol{y}), \boldsymbol{w})$ into 3 parts:

$$F(q(\boldsymbol{y}), \boldsymbol{w}) = \int q(\boldsymbol{y}) \log f(\boldsymbol{w}|\boldsymbol{y}) d\boldsymbol{y} + \int q(\boldsymbol{y}) \log f(\boldsymbol{y}) d\boldsymbol{y} - \int q(\boldsymbol{y}) \log q(\boldsymbol{y}) d\boldsymbol{y}.$$

where $q(\boldsymbol{y}) \sim N(\mathbf{m}, \mathbf{V})$. The second and third integral (i.e. $E_{q(\boldsymbol{y})}(\log f(\boldsymbol{y}))$ and $E_{q(\boldsymbol{y})}(\log q(\boldsymbol{y}))$) have explicit solutions such that

$$E_{q(\boldsymbol{y})}(\log f(\boldsymbol{y})) = -\frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{m}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}-\boldsymbol{\mu}) - \frac{1}{2}\text{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{V})$$

and

$$-E_{q(\boldsymbol{y})}(\log q(\boldsymbol{y})) = \frac{1}{2}\log|\mathbf{V}| + \frac{K}{2} + \frac{K}{2}\log(2\pi).$$

Note that $\mathbf{V}$ is a diagonal matrix. As for the first integral, it has no explicit solution because of the expectation of log sum exponential term:

$$E_{q(\boldsymbol{y})}(\log f(\boldsymbol{w}|\boldsymbol{y})) = C + \boldsymbol{w}^{*T}\mathbf{m} - \left(\sum_{k=1}^{K+1} w_k\right) E_{q(\boldsymbol{y})}\left[\log \sum_{k=1}^{K+1} \exp y_k\right]$$

where $\boldsymbol{w}^*$ represents a K dimension vector with first K elements of $\boldsymbol{w}$, y_{K+1} is set to 0 and C stands for $\log \frac{\mathbf{1}^T \boldsymbol{w}!}{\prod_{k=1}^{K} \boldsymbol{w}_k!}$. Blei and Lafferty (2007) proposed an upper bound for $E_{q(\boldsymbol{y})}\left[\log\left(\sum_{k=1}^{K+1}\exp y_k\right)\right]$ as

$$E_{q(\boldsymbol{y}|\mathbf{m},\mathbf{V})}\left[\log\left(\sum_{k=1}^{K+1}\exp y_k\right)\right] \leq \xi^{-1}\left\{\sum_{k=1}^{K+1} E_{q(\boldsymbol{y}|\mathbf{m},\mathbf{V})}\left[\exp(y_k)\right]\right\} - 1 + \log(\xi), \tag{A.1}$$

where $\xi \in \mathbb{R}$ is introduced as a new variational parameter. Fang and Subedi (2020) utilized this upper bound to find a lower bound for $E_{q(\boldsymbol{y})}(\log f(\boldsymbol{w}|\boldsymbol{y}))$. Here we further simplify the lower bound by Blei and Lafferty (2007). Let $\mathbf{Z} = \sum_{k=1}^{K+1}\exp(y_k)$, then we have:

$$E_{q(\boldsymbol{y})}\left[\log\left(\sum_{k=1}^{K+1}\exp y_k\right)\right] \leq \log E_{q(\boldsymbol{y})}\left(\sum_{k=1}^{K+1}\exp y_k\right) = \log\left[\sum_{k=1}^{K}\exp\left(m_k + \frac{v_k^2}{2}\right) + 1\right],$$

where m_k, v_k^2 stands for k^{th} entry of $\mathbf{m}$ and the k^{th} diagonal entry of $\mathbf{V}$. The two upper bounds are equal when we minimize A.1 with respect to ξ. .
Combining all 3 parts together, we have the approximate lower bound for $\log f(\boldsymbol{w})$:

$$\begin{aligned}\tilde{F}(q(\boldsymbol{y}), \boldsymbol{w}) = C + \boldsymbol{w}^{*T}\mathbf{m} - \left(\sum_{k=1}^{K+1} w_k\right)\left\{\log\left[\sum_{k=1}^{K}\exp\left(m_k + \frac{v_k^2}{2}\right) + 1\right]\right\} + \\ \frac{1}{2}\log|\mathbf{V}| + \frac{K}{2} - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{m}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{m}-\boldsymbol{\mu}) - \frac{1}{2}\text{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{V})\end{aligned}$$

A.2 ELBO for cycle 2

Same as the LNM-FA by Tu and Subedi (2021), in the second cycle, we have

$$\begin{aligned}F(q(\boldsymbol{u},\boldsymbol{y}), \boldsymbol{w}) &= \int q(\boldsymbol{u},\boldsymbol{y})\log\frac{f(\boldsymbol{w},\boldsymbol{u},\boldsymbol{y})}{q(\boldsymbol{u},\boldsymbol{y})}d\boldsymbol{y}d\boldsymbol{u} \\ &= \int q(\boldsymbol{u},\boldsymbol{y})\log f(\boldsymbol{w}|\boldsymbol{u},\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u} + \int q(\boldsymbol{u},\boldsymbol{y})\log f(\boldsymbol{u},\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u} \\ &\quad - \int q(\boldsymbol{u},\boldsymbol{y})\log q(\boldsymbol{u},\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u}.\end{aligned}$$

Furthermore, we assume that $q(\boldsymbol{u},\boldsymbol{y}) = q(\boldsymbol{u})q(\boldsymbol{y})$, $\boldsymbol{u} \sim N(\tilde{\mathbf{m}}, \tilde{\mathbf{V}})$ and $\boldsymbol{y} \sim N(\mathbf{m}, \mathbf{V})$. Thus, the first term can be written as:

$$\begin{aligned}\int q(\boldsymbol{u},\boldsymbol{y})\log f(\boldsymbol{w}|\boldsymbol{u},\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u} &= \int q(\boldsymbol{u})q(\boldsymbol{y})\log f(\boldsymbol{w}|\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u} \\ &= \int q(\boldsymbol{y})\log f(\boldsymbol{w}|\boldsymbol{y})d\boldsymbol{y}\end{aligned}$$

This is identical to the first term in the ELBO in the first cycle and thus its lower bound is

$$\int q(\boldsymbol{u},\boldsymbol{y})\log f(\boldsymbol{w}|\boldsymbol{u},\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u} \geq C + \boldsymbol{w}^{*T}\mathbf{m} - \left(\sum_{k=1}^{K+1}\boldsymbol{w}_k\right)\left\{\log\left(\sum_{k=1}^{K}\exp\left(m_k + \frac{v_k^2}{2}\right) + 1\right)\right\}$$

The third term is

$$-\int q(\boldsymbol{u},\boldsymbol{y})\log q(\boldsymbol{u},\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u}=\frac{1}{2}\left(\log|\mathbf{V}|+\log|\tilde{\mathbf{V}}|+q+K+(K+q)\log 2\pi\right).$$

The second term is

$$\begin{aligned}\int q(\boldsymbol{u},\boldsymbol{y})\log f(\boldsymbol{u},\boldsymbol{y})d\boldsymbol{y}d\boldsymbol{u}&=\int q(\boldsymbol{u})q(\boldsymbol{y})\log[f(\boldsymbol{y}|\boldsymbol{u})f(\boldsymbol{u})]d\boldsymbol{y}d\boldsymbol{u}\\&=E_{q(\boldsymbol{u})}E_{q(\boldsymbol{y})}(\log f(\boldsymbol{y}|\boldsymbol{u})f(\boldsymbol{u}))\\&=-\frac{1}{2}\Big\{(q+K)\log(2\pi)-\log|\boldsymbol{D}|-\tilde{\mathbf{m}}^T\tilde{\mathbf{m}}-\text{tr}(\tilde{\mathbf{V}})-\text{tr}(\boldsymbol{\Lambda}^T\boldsymbol{D}^{-1}\boldsymbol{\Lambda}\tilde{\mathbf{V}})\\&\quad-\text{tr}\left(\boldsymbol{D}^{-1}(\boldsymbol{V}+(\boldsymbol{m}-\boldsymbol{\mu})^T(\boldsymbol{m}-\boldsymbol{\mu}))\right)+2(\boldsymbol{m}-\boldsymbol{\mu})^T\boldsymbol{D}^{-1}\boldsymbol{\Lambda}\tilde{\mathbf{m}}\\&\quad-\tilde{\mathbf{m}}^T\boldsymbol{\Lambda}^T\boldsymbol{D}^{-1}\boldsymbol{\Lambda}\tilde{\mathbf{m}}\Big\}.\end{aligned}$$

Overall, the ELBO in second cycle is:

$$\begin{aligned}F(q(\boldsymbol{u},\boldsymbol{y}),\boldsymbol{w})\geq C+\boldsymbol{w}^T\mathbf{m}-\left(\sum_{i=1}^{K+1}\boldsymbol{w}_i\right)\left\{\log\left(\sum_{k=1}^{K}\exp\left(m_k+\frac{v_k^2}{2}\right)+1\right)\right\}+\\\frac{1}{2}(\log|\mathbf{V}|+\log|\tilde{\mathbf{V}}|+q+K-\log|\boldsymbol{D}|-\tilde{\mathbf{m}}^T\tilde{\mathbf{m}}-tr(\tilde{\mathbf{V}})-\\tr(\boldsymbol{D}^{-1}(\boldsymbol{V}+(\boldsymbol{m}-\boldsymbol{\mu})^T(\boldsymbol{m}-\boldsymbol{\mu})))+2(\boldsymbol{m}-\boldsymbol{\mu})^T\boldsymbol{D}^{-1}\boldsymbol{\Lambda}\tilde{\mathbf{m}}-\\\tilde{\mathbf{m}}^T\boldsymbol{\Lambda}^T\boldsymbol{D}^{-1}\boldsymbol{\Lambda}\tilde{\mathbf{m}}-tr(\boldsymbol{\Lambda}^T\boldsymbol{D}^{-1}\boldsymbol{\Lambda}\tilde{\mathbf{V}}))\end{aligned}$$

where $\mathbf{m}$ and $\mathbf{V}$ are calculated from first stage.

A.3 Parameter estimation for constrained PLMN-FA model

Here, we will obtain the parameter estimates for constrained PLMN-FA model with the constraint $\boldsymbol{\Lambda}_g=\boldsymbol{\Lambda}$. Let $[\boldsymbol{\Lambda}_g]_{ij}$ represents the i^{th} row and j^{th} column of $\boldsymbol{\Lambda}_g$, and $i=1\dots K, j=1\dots q$. $[\boldsymbol{\Lambda}_g]_{i(-j)}$ represents the entire i^{th} row of $\boldsymbol{\Lambda}_g$ but without the j^{th} entry.

1. When $[\boldsymbol{\Lambda}]_{ij}>0$, taking derivative of the variational penalized log likelihood $\tilde{l}_2$ with respect to $[\boldsymbol{\Lambda}]_{ij}$ gives us:

$$\frac{\partial\tilde{l}_2}{\partial[\boldsymbol{\Lambda}]_{ij}}=[\sum_{g=1}^{G}n_g(\mathbf{D}_g^{-1}\mathbf{S}_g\boldsymbol{\beta}_g^T-\mathbf{D}_g^{-1}\boldsymbol{\Lambda}\boldsymbol{\theta}_g)]_{ij}-\frac{s}{1-s},\quad\text{where}\quad n_g=\sum_{i=1}^{n}z_{ig}.$$

   After simplification, the solution is:

$$\frac{\sum_{g=1}^{G}n_g[\boldsymbol{D}_g^{-1}]_{ii}\left([\boldsymbol{S}_g\boldsymbol{\beta}_g^T]_{ij}-[\boldsymbol{\Lambda}]_{i(-j)}[\boldsymbol{\theta}_g]_{(-j)j}\right)-\frac{s}{1-s}}{\sum_{g=1}^{G}n_g[\boldsymbol{D}_g^{-1}]_{ii}[\boldsymbol{\theta}_g]_{jj}}$$

   If the solution less than 0, then we assign $[\boldsymbol{\Lambda}]_{ij}=0$.

2. When $[\mathbf{\Lambda}]_{ij} < 0$, the solution is

$$\frac{\sum_{g=1}^{G} n_g [\boldsymbol{D}_g^{-1}]_{ii} \left([\boldsymbol{S}_g \boldsymbol{\beta}_g^T]_{ij} - [\mathbf{\Lambda}]_{i(-j)} [\boldsymbol{\theta}_g]_{(-j)j} \right) + \frac{s}{1-s}}{\sum_{g=1}^{G} n_g [\boldsymbol{D}_g^{-1}]_{ii} [\boldsymbol{\theta}_g]_{jj}}$$

If the solution greater than 0, then we assign $[\mathbf{\Lambda}]_{ij} = 0$. After setting $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2 = \ldots = \mathbf{\Lambda}_g = \mathbf{\Lambda}$, the estimates of the rest of the parameter remains unchanged.

A.4 True parameters for $\mathbf{\Sigma}_g$ in simulation studies

True $\mathbf{\Lambda}$ for $\mathbf{\Sigma}$ in simulation study 1:

$$\mathbf{\Lambda}^T = \begin{bmatrix} \sqrt{0.5} & \sqrt{0.5} & \sqrt{0.5} & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & \sqrt{0.5} & \sqrt{0.5} & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & \sqrt{0.5} & \sqrt{0.5} \end{bmatrix}$$

True $\mathbf{\Lambda}_g$ for $\mathbf{\Sigma}_g$ in simulation study 2:

$$\mathbf{\Lambda}_1 = \begin{bmatrix} \sqrt{0.4} & 0.00 & 0.00 \\ \sqrt{0.4} & 0.00 & 0.00 \\ 0.00 & \sqrt{0.4} & 0.00 \\ 0.00 & \sqrt{0.4} & 0.00 \\ 0.00 & 0.00 & \sqrt{0.4} \\ 0.00 & 0.00 & \sqrt{0.4} \\ 0.00 & 0.00 & \sqrt{0.4} \end{bmatrix}, \mathbf{\Lambda}_2 = \begin{bmatrix} \sqrt{0.5} & 0.00 & 0.00 \\ \sqrt{0.5} & 0.00 & 0.00 \\ \sqrt{0.5} & 0.00 & 0.00 \\ 0.00 & \sqrt{0.5} & 0.00 \\ 0.00 & \sqrt{0.5} & 0.00 \\ 0.00 & 0.00 & \sqrt{0.5} \\ 0.00 & 0.00 & \sqrt{0.5} \end{bmatrix}, \mathbf{\Lambda}_3 = \begin{bmatrix} \sqrt{0.6} & 0.00 \\ \sqrt{0.6} & 0.00 \\ \sqrt{0.6} & 0.00 \\ \sqrt{0.6} & 0.00 \\ 0.00 & \sqrt{0.6} \\ 0.00 & \sqrt{0.6} \\ 0.00 & \sqrt{0.6} \end{bmatrix}$$

True $\boldsymbol{D}_g$ in Simulation 2.

$$\boldsymbol{D}_1 = \text{diag}\,[0.28, 0.1, 0.22, 0.09, 0.23, 0.14, 0.11]$$
$$\boldsymbol{D}_2 = \text{diag}\,[0.08, 0.2, 0.1, 0.24, 0.18, 0.2, 0.08]$$
$$\boldsymbol{D}_3 = \text{diag}\,[0.2, 0.25, 0.05, 0.16, 0.1, 0.09, 0.22]$$

True parameter generation for simulation study 3:

1. $\boldsymbol{\mu}_1 \sim N(-0.3, 0.1)$, $\boldsymbol{\mu}_2 \sim N(0.1, 0.1)$, and $\boldsymbol{\mu}_3 \sim N(0.5, 0.1)$.
2. $\mathbf{\Lambda}_1 \sim N(0, 1)$, $\mathbf{\Lambda}_2 \sim N(0, 0.1)$, and $\mathbf{\Lambda}_3 \sim N(0, 0.1)$.
3. Diagonal elements of $\boldsymbol{D}_g \sim$ Uniform$(0, 0.4)$

True parameter generation for simulation study 4:

1. $\boldsymbol{\mu}_1 \sim N(0.2, 1)$, $\boldsymbol{\mu}_2 \sim N(-0.2, 1)$, and $\boldsymbol{\mu}_3 \sim N(0, 1)$.
2. $\mathbf{\Lambda}_g \sim N(0, 0.1)$.
3. Diagonal elements of $\boldsymbol{D}_g \sim$ Uniform$(0, 0.05)$.

A.5 Estimated $\mathbf{\Sigma}_g$ for simulation study 3

Table 6: Generating parameters along with the averages and standard errors of the estimated values of the parameters from the 100 datasets from Simulation 3.

	True parameters	Average of estimated parameters (standard errors)
Component 1($n_1 = 500$)		
$\boldsymbol{\Sigma}_1$	$\begin{bmatrix} 1.75 & -1.15 & 0 & 0 & 0 & 1.90 & 0 & 0 \\ -1.15 & 1.71 & 0 & 0.63 & -0.47 & -1.27 & 0 & 0.09 \\ 0 & 0 & 0.21 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.63 & 0 & 0.68 & -0.39 & 0 & 0 & 0.07 \\ 0 & -0.47 & 0 & -0.39 & 0.74 & 0 & 0 & 0.26 \\ 1.90 & -1.27 & 0 & 0 & 0 & 2.22 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.23 & 0 \\ 0 & 0.09 & 0 & 0.07 & 0.26 & 0 & 0 & 1.55 \end{bmatrix}$	$\begin{bmatrix} 0.86 & -0.50 & 0.02 & 0.03 & 0.01 & 0.90 & 0.03 & 0.02 \\ -0.50 & 0.87 & 0.01 & 0.30 & -0.20 & -0.56 & 0.01 & 0.02 \\ 0.02 & 0.01 & 0.22 & 0.02 & 0.01 & 0.01 & 0.02 & 0.01 \\ 0.03 & 0.30 & 0.02 & 0.42 & -0.16 & 0.03 & 0.02 & 0.03 \\ 0.01 & -0.20 & 0.01 & -0.16 & 0.43 & 0.01 & 0.02 & 0.12 \\ 0.90 & -0.56 & 0.01 & 0.03 & 0.01 & 1.11 & 0.02 & 0.02 \\ 0.03 & 0.01 & 0.02 & 0.02 & 0.02 & 0.02 & 0.26 & 0.02 \\ 0.02 & 0.02 & 0.01 & 0.03 & 0.12 & 0.02 & 0.02 & 0.91 \end{bmatrix}$
Proportion of times the elements of $\hat{\boldsymbol{\Sigma}}_1$ is non-zero		$\begin{pmatrix} 1.00 & 1.00 & 0.65 & 0.87 & 0.74 & 1.00 & 0.80 & 0.67 \\ 1.00 & 1.00 & 0.76 & 1.00 & 1.00 & 1.00 & 0.80 & 0.76 \\ 0.65 & 0.76 & 1.00 & 0.70 & 0.67 & 0.76 & 0.70 & 0.57 \\ 0.87 & 1.00 & 0.70 & 1.00 & 1.00 & 0.91 & 0.78 & 0.74 \\ 0.74 & 1.00 & 0.67 & 1.00 & 1.00 & 0.85 & 0.83 & 0.96 \\ 1.00 & 1.00 & 0.76 & 0.91 & 0.85 & 1.00 & 0.80 & 0.63 \\ 0.80 & 0.80 & 0.70 & 0.78 & 0.83 & 0.80 & 1.00 & 0.72 \\ 0.67 & 0.76 & 0.57 & 0.74 & 0.96 & 0.63 & 0.72 & 1.00 \end{pmatrix}$
Component 2($n_2 = 300$)		

$\boldsymbol{\Sigma}_2$

$$\begin{bmatrix} 0.03 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.17 & 0.00 & 0 & 0.00 & 0 & 0.01 & 0 \\ 0 & 0.00 & 0.21 & 0 & 0.00 & 0 & 0.00 & 0 \\ 0 & 0 & 0 & 0.16 & 0 & 0 & 0 & 0 \\ 0 & 0.00 & 0.00 & 0 & 0.09 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.12 & 0 & 0 \\ 0 & 0.01 & 0.00 & 0 & 0.01 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.20 \end{bmatrix}$$

Proportion of times the elements of $\hat{\boldsymbol{\Sigma}}_2$ is non-zero

$$\begin{pmatrix} 1.00 & 0.87 & 0.74 & 0.76 & 0.72 & 0.87 & 0.74 & 0.63 \\ 0.87 & 1.00 & 0.76 & 0.87 & 0.87 & 0.87 & 0.78 & 0.65 \\ 0.74 & 0.76 & 1.00 & 0.74 & 0.70 & 0.72 & 0.63 & 0.61 \\ 0.76 & 0.87 & 0.74 & 1.00 & 0.87 & 0.83 & 0.74 & 0.70 \\ 0.72 & 0.87 & 0.70 & 0.87 & 1.00 & 0.78 & 0.76 & 0.83 \\ 0.87 & 0.87 & 0.72 & 0.83 & 0.78 & 1.00 & 0.76 & 0.65 \\ 0.74 & 0.78 & 0.63 & 0.74 & 0.76 & 0.76 & 1.00 & 0.63 \\ 0.63 & 0.65 & 0.61 & 0.70 & 0.83 & 0.65 & 0.63 & 1.00 \end{pmatrix}$$

$$\begin{bmatrix} 0.67 & -0.37 & 0.02 & 0.03 & 0.01 & 0.68 & 0.02 & 0.02 \\ -0.37 & 0.71 & 0.01 & 0.24 & -0.15 & -0.42 & 0.01 & 0.02 \\ 0.02 & 0.01 & 0.22 & 0.02 & 0.01 & 0.02 & 0.02 & 0.01 \\ 0.03 & 0.24 & 0.02 & 0.37 & -0.12 & 0.03 & 0.02 & 0.03 \\ 0.01 & -0.15 & 0.01 & -0.12 & 0.35 & 0.01 & 0.02 & 0.09 \\ 0.68 & -0.42 & 0.02 & 0.03 & 0.01 & 0.86 & 0.02 & 0.02 \\ 0.02 & 0.01 & 0.02 & 0.02 & 0.02 & 0.02 & 0.26 & 0.02 \\ 0.02 & 0.02 & 0.01 & 0.03 & 0.09 & 0.02 & 0.02 & 0.73 \end{bmatrix}$$

Component 3($n_3 = 200$)

$\boldsymbol{\Sigma}_3$

$$\begin{bmatrix} 0.03 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.19 & 0.02 & 0 & -0.00 & 0 & 0.01 & 0 \\ 0 & 0.02 & 0.21 & 0 & -0.00 & 0 & 0.01 & 0.00 \\ 0 & 0 & 0 & 0.16 & 0 & 0 & 0 & 0 \\ 0 & -0.00 & -0.00 & 0 & 0.09 & 0 & -0.00 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.12 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0 & -0.00 & 0 & 0.24 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.20 \end{bmatrix}$$

$$\begin{bmatrix} 0.08 & -0.01 & -0.00 & -0.00 & 0.00 & 0.04 & -0.00 & -0.00 \\ -0.01 & 0.22 & 0.00 & 0.00 & -0.00 & -0.01 & 0.00 & 0.00 \\ -0.00 & 0.00 & 0.13 & 0.00 & -0.00 & 0.00 & 0.00 & 0.00 \\ -0.00 & 0.00 & 0.00 & 0.12 & -0.00 & -0.00 & 0.00 & 0.00 \\ 0.00 & -0.00 & -0.00 & -0.00 & 0.20 & 0.00 & 0.00 & 0.00 \\ 0.04 & -0.01 & 0.00 & -0.00 & 0.00 & 0.16 & -0.00 & -0.00 \\ -0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.00 & 0.15 & 0.00 \\ -0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -0.00 & 0.00 & 0.77 \end{bmatrix}$$

Proportion of times the elements of $\hat{\boldsymbol{\Sigma}}_3$ is non-zero

$$\begin{pmatrix}
1.00 & 0.11 & 0.02 & 0.04 & 0.02 & 0.15 & 0.04 & 0.02 \\
0.11 & 1.00 & 0.02 & 0.04 & 0.02 & 0.09 & 0.02 & 0.02 \\
0.02 & 0.02 & 1.00 & 0.02 & 0.02 & 0.00 & 0.00 & 0.00 \\
0.04 & 0.04 & 0.02 & 1.00 & 0.02 & 0.02 & 0.02 & 0.00 \\
0.02 & 0.02 & 0.02 & 0.02 & 1.00 & 0.00 & 0.00 & 0.00 \\
0.15 & 0.09 & 0.00 & 0.02 & 0.00 & 1.00 & 0.04 & 0.02 \\
0.04 & 0.02 & 0.00 & 0.02 & 0.00 & 0.04 & 1.00 & 0.00 \\
0.02 & 0.02 & 0.00 & 0.00 & 0.00 & 0.02 & 0.00 & 1.00
\end{pmatrix}$$