Gene expression

Finite mixtures of matrix variate Poisson-log normal distributions for three-way count data

Anjali Silva¹,², Xiaoke Qin³, Steven J. Rothstein², Paul D. McNicholas⁴, Sanjeena Subedi *³

¹Department of Mathematics and Statistics, University of Guelph, Guelph, ON N1G 2W1, Canada
²Department of Molecular and Cellular Biology, University of Guelph, Guelph, ON N1G 2W1, Canada
³School of Mathematics and Statistics, Carleton University, Ottawa, ON K1S 5B6, Canada
⁴Department of Mathematics and Statistics, McMaster University, Hamilton, ON L8S 4L8, Canada

*Corresponding author. School of Mathematics and Statistics, Carleton University, Ottawa, ON K1S 5B6, Canada.
E-mail: sanjeena.dang@carleton.ca

Associate Editor: Alfonso Valencia

Received 24 June 2022; revised 9 February 2023; accepted 19 March 2023

Abstract

Motivation: Three-way data structures, characterized by three entities, the units, the variables and the occasions, are frequent in biological studies. In RNA sequencing, three-way data structures are obtained when high-throughput transcriptome sequencing data are collected for \( n \) genes across \( p \) conditions at \( r \) occasions. Matrix variate distributions offer a natural way to model three-way data and mixtures of matrix variate distributions can be used to cluster three-way data. Clustering of gene expression data is carried out as means of discovering gene co-expression networks.

Results: In this work, a mixture of matrix variate Poisson-log normal distributions is proposed for clustering read counts from RNA sequencing. By considering the matrix variate structure, full information on the conditions and occasions of the RNA sequencing dataset is simultaneously considered, and the number of covariance parameters to be estimated is reduced. We propose three different frameworks for parameter estimation: a Markov chain Monte Carlo-based approach, a variational Gaussian approximation-based approach, and a hybrid approach. Various information criteria are used for model selection. The models are applied to both real and simulated data, and we demonstrate that the proposed approaches can recover the underlying cluster structure in both cases. In simulation studies where the true model parameters are known, our proposed approach shows good parameter recovery.

Availability and implementation: The GitHub R package for this work is available at https://github.com/anjalisilva/mixMVPLN and is released under the open source MIT license.

1 Introduction

Finite mixture models are popular for clustering applications and are widely used on two-way data (McLachlan and Peel 2000; McNicholas 2016). Three-way data are becoming increasingly commonplace in several fields, including bioinformatics. Three-way data structures are characterized by three entities or modes: the units (rows), the variables (columns), and the occasions (layers). For two-way data, each observation is represented as a vector whereas, for three-way data, each observation can be regarded as a matrix. A random matrix \( T \) is said to contain \( k \in \{1, \ldots, p\} \) responses over \( i \in \{1, \ldots, r\} \) occasions and \( n = 1, \ldots, N \) such units are considered. This provides \( N \) independent and identically distributed random matrices \( T_1, T_2, \ldots, T_N \).

Matrix variate distributions offer a natural approach for modeling three-way data. Extensions of matrix variate distributions in the context of mixture models have given rise to mixtures of matrix variate distributions, which have been used to cluster three-way data (Viroli 2011; Anderlucci and Viroli 2015; Dogru et al. 2016; Gallaugher and McNicholas 2018). Here, the interest lies in clustering the \( N \) observed matrices into \( G \) clusters, while utilizing all information from the other two modes (Viroli 2011). It is assumed that matrices \( T_1, T_2, \ldots, T_N \) are conditionally independent and identically distributed observations coming from a mixture model with \( G \) possible groups in proportions \( \pi_1, \ldots, \pi_G \) (Viroli 2011). The density of the \( G \)-component mixture is

\[
f(T|\pi_1, \ldots, \pi_G, \theta_1, \ldots, \theta_G) = \sum_{g=1}^G \pi_g f^{(g)}(T | \theta_g).
\]

Here parameters of the distribution function \( f^{(g)}(\cdot) \) are represented by \( \theta_g \) and \( \pi_g > 0 \), such that \( \sum_{g=1}^G \pi_g = 1 \), is the mixing proportion of the \( g \)th component.

Three-way datasets are common in biological studies, including RNA sequencing (RNA-seq), where gene expression count data are collected for \( N \) genes across \( p \) conditions at \( r \) occasions. However, efficiently analyzing these complex data remains an ongoing project.

© The Author(s) 2023. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.
challenge. While some early work utilized the Poisson distribution to model such count data (Marioni et al. 2008; Bullard et al. 2010), this was not ideal because of its restrictive mean–variance relationship, and so the negative binomial distribution emerged as the univariate distribution of choice (Love et al. 2014; Dong et al. 2016).

However, the multivariate extensions of Poisson (Campbell 1934) and negative binomial distributions (Doss 1979) are seldom used in practice due to their computational complexity (Brijs et al. 2004). Silva et al. (2019) proposed a mixture model-based clustering methodology for overdispersed, multivariate count data based on the multivariate Poisson-log normal (MPLN) distribution for two-way RNA-seq data. For genes $n \in \{1, \ldots, N\}$ and samples $c \in \{1, \ldots, r\}$, the MPLN distribution is given by

$$Y_{nc} | \theta_{nc} \sim \mathcal{P}(\exp(\theta_{nc} + \log s_n))$$

$$(\theta_{n1}, \ldots, \theta_{nrp})' \sim \mathcal{N}(\mu, \Sigma),$$

where $\mathcal{P}$ denotes the Poisson distribution and the $\mathcal{N}(\mu, \Sigma)$ is an $rp$-dimensional normal distribution. To account for the differences in library sizes across each sample $c$ of an RNA-seq study, a fixed, known constant $s_n$ representing the normalized library sizes, is added to the mean of the Poisson distribution. In this work, mixtures of MPLN distributions and matrix variate normal distributions are extended to give rise to mixtures of matrix variate Poisson-log normal (MVPLN) distributions for clustering three-way count data. Details of parameter estimation are provided, and both real and simulated data illustrations are used to demonstrate the clustering ability.

2 Materials and methods

2.1 Matrix variate Poisson-log normal distribution

Mathematical properties of the matrix variate normal distribution can be found in Gupta and Nager (2000). By considering a matrix variate structure, the number of free covariance parameters to be estimated is reduced from $(1/2)rp(rp + 1)$ to $(1/2)(rp + r + p)(rp + p + 1)$. The matrix variate normal distribution can be extended to give rise to MVPLN distribution using a hierarchical structure. Consider $N$ independent and identically distributed random matrices $Y_{nk}, n = 1, \ldots, N$, each of dimension $r \times p$. In the MVPLN framework, $Y_{nk} | \theta_{nk}$ follows a Poisson distribution with mean $exp(\theta_{nk})$, and $\theta_{nk}$ follows a $r \times p$ matrix variate normal distribution $\mathcal{N}_{rp}(M, \Phi, \Omega)$, where $M$ is a $r \times p$ matrix of means, $\Phi$ is a $r \times r$ covariance matrix containing the variances and covariances between $r$ occasions, and $\Omega$ is a $p \times p$ covariance matrix containing the variance of the $p$ variables. Figure 1 provides a graphical representation of a mixture of MVPLN distributions.

The vectorization of $Y_{nk}$, denoted $\text{vec}(Y_{nk})$, is $rp$-dimensional. Given all $\text{vec}(Y_{nk})$, i.e., for $n = 1, \ldots, N$, the library sizes vec($s$) can be calculated. The vec($s$) and vec($\theta_{nk}$) are $rp$-dimensional. The covariance matrix of vec($Y_{nk}$) is $\Sigma = \Phi \otimes \Omega$, where $\otimes$ denotes the Kronecker product. Note that $\Sigma$ has dimension $rp \times rp$, and the probability mass function of the MVPLN distribution is

$$f(Y_{nk}, s | \theta) = \int \prod_{k=1}^{N} f(\text{vec}(Y_{nk}), \text{vec}(\theta_{nk}), \text{vec}(s)) g^{(r,p)}(\theta_{nk} | \theta) d\text{vec}(\theta_{nk})$$

and the complete-data log-likelihood is

$$l_c(\Theta) = \sum_{n=1}^{N} \sum_{g=1}^{G} n_{ng} \log \pi_g$$

$$- \sum_{n=1}^{N} \sum_{g=1}^{G} n_{ng} \log \text{vec}(\theta_{ng}) + \log \text{vec}(s)$$

$$+ \sum_{n=1}^{N} \sum_{g=1}^{G} \sum_{c=1}^{rp} n_{ng} \log(\text{vec}(\theta_{ng})) \text{vec}(Y_{nk})'$$

$$- \sum_{n=1}^{N} \sum_{g=1}^{G} \sum_{c=1}^{rp} \sum_{l=1}^{rp} \log(\text{vec}(Y_{nk})_l) \cdot \frac{nrp}{2} \log(2\pi)$$

$$- \frac{p}{2} \sum_{g=1}^{G} n_{ng} \log |\Phi_g| - \frac{r}{2} \sum_{g=1}^{G} n_{ng} \log |\Omega_g|$$

$$- \frac{1}{2} \sum_{g=1}^{G} \sum_{n=1}^{N} \text{vec}(\theta_{ng})' (\theta_{ng} - M_g) \Sigma_g^{-1} (\theta_{ng} - M_g)$$

where $\Theta = (M, \Phi, \Omega), f(\cdot)$ is the probability mass function of Poisson distribution, $\text{vec}(Y_{nk})_l$ represent the $l^{th}$ element of vec($Y_{nk}$), and $g^{(r,p)}(\cdot)$ is the probability density function of matrix variate normal distribution.

The unconditional mean and covariance of the MPLN distribution can be calculated using the properties of the log-normal distribution and of the conditional expectation (Aitchison and Ho 1989; Tunaru 2002). For the MVPLN distribution, the unconditional mean and covariance are

$$\mathbb{E}(Y_{nk}) = \mathbb{E}[\mathbb{E}(Y_{nk} | \theta_{nk})] = \exp \left\{ \mu_n + \frac{1}{2} (\Phi_n \Omega_n) \right\} = M_n$$

$$\text{Var}(Y_{nk}) = \text{Var}[\mathbb{E}(Y_{nk} | \theta_{nk})] + \text{Var}[\mathbb{E}(Y_{nk} | \theta_{nk})]$$

$$= M_n + M_n (\exp(\Phi_n \Omega_n) - 1),$$

respectively. The MVPLN distribution can account for both the correlations between variables and the correlations between occasions, as two different covariance matrices are used for the two modes. This makes the model ideal for modeling RNA-seq data when expression measurements for different conditions at different occasions, c.g. time-points or replicates, are available.

2.2 Finite mixtures of MVPLN distributions

In the mixture model context, a random matrix $Y_{nk}$ is assumed to come from a population with $G$ subgroups each distributed according to an MVPLN distribution. Then $N$ such matrices $Y_1, Y_2, \ldots, Y_N$ are observed, each of which belongs to one of $g \in \{1, \ldots, G\}$ different sub-populations with mixing proportions $\pi_1, \ldots, \pi_G$. Then the probability density function of a G-component mixture of MVPLN distributions can be written as

$$f(Y; \Theta) = \sum_{g=1}^{G} \pi_g f(Y | M_g, \Phi_g, \Omega_g)$$

$$= \sum_{g=1}^{G} \prod_{n=1}^{N} \left[ \prod_{c=1}^{rp} f(\text{vec}(Y_{nk}), \text{vec}(\theta_{ng}), \text{vec}(s)) \times g^{(r,p)}(\theta_{ng} | M_g, \Phi_g, \Omega_g) \right]$$

where $\Theta = (\pi_1, \ldots, \pi_G, M_1, \ldots, M_G, \Phi_1, \ldots, \Phi_G, \Omega_1, \ldots, \Omega_G)$, $f(\cdot)$ is the probability mass function of a Poisson distribution and the $g^{(r,p)}(\cdot)$ is the probability density function of matrix variate normal distribution. The cluster membership of all units is assumed to be unknown and $z_{ng} = 1$ if $Y_{nk}$ is in component $g$ and $z_{ng} = 0$ otherwise. The complete data consist of the observed and missing data, i.e. $(Y_1, \ldots, Y_N, z_1, \ldots, z_N, \theta_1, \ldots, \theta_N)$. The complete-data likelihood is

$$L_c(\theta) = \prod_{n=1}^{N} \prod_{g=1}^{G} \pi_g \left[ \prod_{c=1}^{rp} f(\text{vec}(Y_{nk}), \text{vec}(\theta_{ng}), \text{vec}(s)) \times g^{(r,p)}(\theta_{ng} | M_g, \Phi_g, \Omega_g) \right]$$

and the complete-data log-likelihood is

$$l_c(\theta) = \sum_{n=1}^{N} \sum_{g=1}^{G} n_{ng} \log \pi_g$$

$$- \sum_{n=1}^{N} \sum_{g=1}^{G} z_{ng} \log \text{vec}(\theta_{ng}) + \log \text{vec}(s)$$

$$+ \sum_{n=1}^{N} \sum_{g=1}^{G} z_{ng} \log(\text{vec}(\theta_{ng})) \text{vec}(Y_{nk})'$$

$$- \sum_{n=1}^{N} \sum_{g=1}^{G} \sum_{c=1}^{rp} \sum_{l=1}^{rp} \log(\text{vec}(Y_{nk})_l) \cdot \frac{nrp}{2} \log(2\pi)$$

$$- \frac{p}{2} \sum_{g=1}^{G} n_{ng} \log |\Phi_g| - \frac{r}{2} \sum_{g=1}^{G} n_{ng} \log |\Omega_g|$$

$$- \frac{1}{2} \sum_{g=1}^{G} \sum_{n=1}^{N} \text{vec}(\theta_{ng})' (\theta_{ng} - M_g) \Sigma_g^{-1} (\theta_{ng} - M_g).$$

Figure 1. Graphical representation of the MVPLN mixture model.
where $n_g = \sum_{n=1}^{N} z_{ng}$ and $\log(\text{vec}(Y_n), \theta)$ is the log of the factorial of the $z_{ng}$ elements of $\text{vec}(Y_n)$. Compared to the mixtures of MLPN distribution, the number of free parameters to be estimated is reduced by considering a matrix variate structure (see Figs 2 and 3). For the mixtures of MLPN model, the number of free parameters is $K = (G - 1) + (Grp) + \frac{1}{2} Grp(rp + 1)$, whereas for mixtures of MVPLN model it is $K = (G - 1) + (Grp) + \frac{1}{2} G(rp + 1) + p(rp + 1)]$.

2.3 Parameter estimation

Three different frameworks for parameter estimation for the mixtures of MVPLN models are proposed: one based on Markov chain Monte Carlo (MCMC) methods, one based on variational Gaussian approximation (VGA) as well as a hybrid approach. MCMC-based approaches are computationally intensive; hence, we also provide a computationally efficient variational approximation framework for parameter estimation. Finally, a hybrid approach combines the variational approximation-based approach and MCMC-based approach.

2.3.1 MCMC-based approach

In the MCMC-based approach, the MCMC expectation–maximization (MCMC–EM) algorithm is used to estimate the model parameters (see Silva et al. 2019, for details). Using an MCMC–EM algorithm, the expected value of the $\theta_g$ and the $Z_{ng}$ conditional on the parameter updates from the $n$th iteration, respectively, are updated in the expectation (E-) step as follows:

\[
\begin{align*}
E(\theta_g|Y_n) & \approx \frac{1}{W} \sum_{f=1}^{W} \theta_g^{(f)} = \theta_g^{(0)} , \\
E(Z_{ng}|Y_n, \theta_0, s) & = \sum_{\omega=1}^{\omega_{ng}} \omega_{ng} := z_{ng},
\end{align*}
\]

(1)

where $\omega_{ng} = \pi_g^{(t)} \left( \prod_{c=1}^{C} \left( \text{vec}(Y_n c), \text{vec}(\theta_{ng} c), \text{vec}(s) \right) \right) \times g^{(t)}(r_p (\theta_g^{(t)} M^{(t)}, \Phi_g^{(t)} \Omega_g^{(t)})$ and $\theta_g^{(t)}$ is a random sample simulated via the RStan package for iterations $f = 1, \ldots, B$. In the E-step, the expectation is taken conditional on the current parameter estimates; hence, the use of $(t)$ on parameters in (1). As the values from initial iterations are discarded from further analysis to minimize bias, the number of iterations used for parameter estimation is $W$, where $W < B$

The conditional expected-value of the complete-data log-likelihood is

\[
Q(\Theta) \approx \mathbb{E}[l(\Theta)] \approx C - \frac{p}{2} \sum_{g=1}^{G} \log(\Phi_g) - \frac{1}{2} \sum_{g=1}^{G} n_g \log(\Omega_g)
\]

\[
= \sum_{n=1}^{N} \sum_{g=1}^{G} z_{ng} \left[ \log(\Phi_g) \right] + \log(\Omega_g)
\]

(2)

where $C$ is a constant with respect to $M_g$, $\Phi_g$ and $\Omega_g$, and $n_g = \sum_{n=1}^{N} z_{ng}$. During the M-step, the updates for the parameters are obtained as follows:

\[
\begin{align*}
\pi_g^{(t+1)} & = \frac{h_{ng}^{(t)}}{N}, \\
M_g^{(t+1)} & = \frac{\sum_{n=1}^{N} z_{ng} \log(\theta_{ng})}{\sum_{n=1}^{N} z_{ng}}, \\
\Phi_g^{(t+1)} & = \frac{\sum_{n=1}^{N} z_{ng} \log(\theta_{ng} - M_g^{(t+1)})}{\sum_{n=1}^{N} z_{ng}} + M_g^{(t+1)}, \\
\Omega_g^{(t+1)} & = \frac{\sum_{n=1}^{N} z_{ng} \log(\theta_{ng} - M_g^{(t+1)})}{\sum_{n=1}^{N} z_{ng}} + M_g^{(t+1)}
\end{align*}
\]

2.3.2 VGA-based approach

Variational approximations (Wainwright and Jordan 2007) are approximate inference techniques in which a computationally convenient approximating density is used in place of a more complex but “true” posterior density. The approximating density is obtained by minimizing the Kullback–Leibler (KL) divergence between the true and the approximating densities. Suppose we have an approximating density $q(\Theta)$, then the marginal log of the probability mass function can be written

\[
\log f(Y) = f(q, Y) + D_{KL}(q||f),
\]

where $D_{KL}(q||f) = \int q(\Theta) \log \frac{d(\Theta)}{d(q(\Theta))} d\Theta$ is the KL divergence between $f(\Theta|Y)$ and approximating distribution $q(\Theta)$, and $f(Y, q) = \int \log f(Y, \Theta) - \log q(\Theta) d\Theta$ is our evidence lower bound (ELBO). Thus, to minimize the KL divergence, we maximize our ELBO. For VGA, $q(\Theta)$ is assumed to be a Gaussian distribution.

The complete-data log-likelihood for the mixtures of MVPLN distributions can be written

\[
\begin{align*}
L(\Theta) & = \sum_{n=1}^{N} \sum_{g=1}^{G} \pi_g \log(\pi_g) + \sum_{n=1}^{N} \sum_{g=1}^{G} \log f(Y_n | M_g, \Phi_g, \Omega_g) \\
& = \sum_{g=1}^{G} \sum_{n=1}^{N} \pi_g \log(\pi_g) + \sum_{n=1}^{N} \sum_{g=1}^{G} \log f(Y_n, q_{ng}) + D_{KL}(q_{ng}||f_{ng})
\end{align*}
\]

where $D_{KL}(q_{ng}||f_{ng}) = \int q_{ng}(\Theta) \log \frac{d(\Theta)}{d(q_{ng}(\Theta))} d\Theta$ is the KL divergence between $f(\Theta|Y_n, Z_{ng} = 1)$ and approximating distribution $q(\Theta)$. Assuming $q(\Theta) = N_f(r_p (\xi_{ng}, \Theta_{ng}, s_{ng})$, the ELBO for each observation $y_n$ becomes
The variational parameters that maximize the ELBO will minimize the KL divergence between the true posterior and the approximation. Thus, parameter estimation can be done in an iterative EM-type approach such that the following steps are iterated. At the (t + 1)th step:

1. Conditional on the variational parameters $\xi_{ng}$, $\Delta_{ng}$, and $\kappa_{ng}$ and on $M_t$, $\Phi_t$, and $\Omega_t$, the $E(Z_{ng}|Y_t)$ is computed. Given $\pi_g f(M_t, \Phi_t, \Omega_t)$, and $E(Z_{ng}|Y_t)$:

$$E(Z_{ng}|Y_t) = \frac{\pi_g f(Y_t|M_t, \Phi_t, \Omega_t)}{\sum_{g=1}^{G} \pi_g f(Y_t|M_t, \Phi_t, \Omega_t)}$$

Note that this involves the marginal distribution of $Y$ which is difficult to compute. Hence, we use an approximation of $E(Z_{ng}|Y_t)$, where we replace the marginal density of the exponent of ELBO such that

$$\hat{z}_{ng}^{(t+1)} \equiv \frac{\sum_{g=1}^{G} \pi_g \exp[F(Y_t|Y_{ng})]}{\sum_{g=1}^{G} \pi_g \exp[F(Y_t|Y_{ng})]}$$

This approximation is computationally convenient and a similar framework has been previously utilized (Gollini and Murphy 2014; Tang et al. 2015). This approximation works well in simulation studies and real data analysis.

2. Given $\hat{z}_{ng}^{(t+1)}$, variational parameters $\xi_{ng}$, $\Delta_{ng}$, and $\kappa_{ng}$ are updated conditional on $M_t^{(t)}$, $\Phi_t^{(t)}$, and $\Omega_t^{(t)}$.
   a. A fixed-point method is used for updating $\Delta_{ng}$:

$$\Delta_{ng}^{(t+1)} = \left[ \sum_{g=1}^{G} \pi_g \left( \exp \left( \xi_{ng}^{(t)} \right) + \log \pi_g \right) \right]^{-1} \left[ \sum_{g=1}^{G} \pi_g \left( \exp \left( \xi_{ng}^{(t)} \right) + \log \pi_g \right) \right]^{-1}$$

where the vector function $\exp[a] = (e^{a_1}, \ldots, e^{a_p})'$ is a vector of the exponential of each element of the $p$-dimensional vector $a$, $\pi_g(\mathbf{k}) = (\pi_1, \ldots, \pi_p)$ puts the diagonal elements of the $p \times p$ matrix $\mathbf{k}$ into a $p$-dimensional vector, and $\odot$ is the Hadamard product.

b. A fixed-point method is used for updating $\kappa_{ng}$:

$$\kappa_{ng}^{(t+1)} = \left[ \sum_{g=1}^{G} \pi_g \left( \exp \left( \xi_{ng}^{(t)} \right) + \log \pi_g \right) \right]^{-1} \left[ \sum_{g=1}^{G} \pi_g \left( \exp \left( \xi_{ng}^{(t)} \right) + \log \pi_g \right) \right]^{-1}$$

where the vector function $\exp[a] = (e^{a_1}, \ldots, e^{a_p})'$ is a vector of the exponential of each element of the $p$-dimensional vector $a$, $\pi_g(\mathbf{k}) = (\pi_1, \ldots, \pi_p)$ puts the diagonal elements of the $p \times p$ matrix $\mathbf{k}$ into a $p$-dimensional vector, and $\odot$ is the Hadamard product.

c. Newton’s method is used to update $\xi_{ng}^{(t+1)}$:

$$\xi_{ng}^{(t+1)} = \left( \sum_{g=1}^{G} \pi_g \left( \exp \left( \xi_{ng}^{(t)} \right) + \log \pi_g \right) \right)$$

3. Given $\hat{z}_{ng}^{(t+1)}$, $\Delta_{ng}^{(t+1)}$, and $\kappa_{ng}^{(t+1)}$, the parameters $\pi_g$, $\Phi_t$, and $\Omega_t$ can be solved for as

$$\pi_g^{(t+1)} = \frac{\pi_g^{(t+1)}}{N}$$

$$\Phi_t^{(t+1)} = \left[ \sum_{g=1}^{G} \hat{z}_{ng}^{(t+1)} \pi_g^{(t+1)} - \Phi_t^{(t+1)} \Omega_t^{(t+1)} \right]$$

2.3.3 Hybrid approach

While the MCMC-based approach can generate exact results, fitting such models can be computationally intensive because we need to evaluate the expected complete-data log-likelihood with respect to the posterior distribution of the latent variables at every iteration of the EM algorithm. For example, for datasets with $N = 1000$, $rp = 6$, and $G = 2$, in simulation setting 2, the MCMC-based approach took on average of $\sim 41h$. On the other hand, the VGA approach is computationally efficient—even on the same set of datasets from simulation 2 with $N = 1000$, $rp = 6$, and $G = 2$, fitting such a model took an average of $\sim 2$ min (see Table 7 of Supplementary File 1 for complete details). However, it does not guarantee an exact posterior (Ghahramani and Beal 1999). Thus, we provide a computationally efficient hybrid approach in which:

- Step 1: Fit the model using the VGA-based approach.
- Step 2: Estimate the component indicator variable $z_{ng}$ conditional on the parameter estimates from the VGA-based approach.
- Step 3: Using the parameter estimates from Step 1 as the initial values for the parameters and using the classification from Step 2, compute the MCMC-based expectation for the latent variable $\theta_{ng}$ and obtain the final estimates of the model parameters.

The hybrid approach comes with a substantial reduction in computational overhead compared to a traditional MCMC-EM but it can generate samples from the exact posterior distribution. Fitting such a model using the hybrid approach on the same set of datasets from Simulation setting 2 with $N = 1000$, $rp = 6$, and $G = 2$ took on average just under 10 min. Complete details of computational times for all three simulation settings are provided in Table 7 of Supplementary File 1. When the primary goal is to detect the underlying clusters (which is the case for the real data analysis), the VGA-based approach is sufficient. However, when the primary goal is posterior inference, we recommend the hybrid approach as it can better yield an exact posterior similar to the MCMC-EM approach but is computationally efficient.

Details on the convergence criteria, initialization, and parallel implementation for an MCMC-EM approach is provided in Supplementary File 3.

2.4 Identifiability

Model identifiability is vital to obtain unique and consistent parameter estimates. Identifiability of univariate and multivariate finite mixtures of normal distributions has been proved (Teicher 1963; Yakowitz and Spragins 1968). With regard to the mixtures of MVPNL distributions, the estimates for $\Phi_t$ and $\Omega_t$ are only unique up to a strictly positive constant. To eliminate identifiability issues, a constraint needs to be imposed, e.g. the trace of $\Omega_t$ can be set equal to $p$, the trace of $\Phi_t$ can be set equal to $r$, or the first diagonal element of $\Phi_t$ can be set equal to 1. The latter solution, which is used by Gallaugher and McNicholas (2018), is used for all analyses.
in this paper. To obtain final parameter estimates, the resulting $\Phi^G$ is divided by the first diagonal element of $\Phi^G$, and $\Omega^G_z$ is multiplied by the first diagonal element of $\Phi^G$.

2.5 Model selection and performance assessment

Four model selection criteria are offered, which include the Akaike information criterion (AIC; Akaike 1973), the Bayesian information criterion (BIC; Schwarz 1978), a variation of the AIC used by Bozdogan (1994) called AIC3, and the integrated completed likelihood (ICL; Biernacki et al. 2000). These criteria are given by $\text{AIC} = 2 \log L(\theta^{MLE}; y) + 2K$, $\text{BIC} = 2 \log L(\theta^{MLE}; y) - K \log N$, $\text{AIC3} = 2 \log L(\theta^{MLE}; y) + 3K$, and $\text{ICL} = \text{BIC} + 2 \sum_{g=1}^{G} \sum_{t=1}^{N_g} \text{MAP}[z_{ng}(t)] \log z_{ng}(t)$, respectively, where $L(\theta^{MLE}; y)$ represents maximized log-likelihood, $\theta^{MLE}$ is the maximum likelihood estimate of the model parameters, $N$ is the number of observations, and $\text{MAP}[z_{ng}(t)]$ is the maximum a posteriori classification given $z_{ng}(t)$.

In situations where the true classes are known but, for clustering purposes, are ignored, the adjusted Rand index (ARI; Hubert and Arabie 1985) can be used for performance assessment. The ARI takes a value 1 under perfect class agreement and has expected value 0 under random classification.

3 Results

3.1 Simulations

Simulation studies were conducted to illustrate the ability to recover the true underlying parameters for the mixtures of MVPLN algorithm. For Simulation 1, datasets with $G = 1$ component were generated with $N = 1000$ observations, $r = 2$ and $p = 3$. For Simulation 2, datasets with $G = 2$ components and $\pi_1 = 0.79$ were generated with $N = 1000$ observations, $r = 2$ and $p = 3$. For Simulation 3, datasets with $G = 2$ components and $\pi_1 = 0.6$ were generated with $N = 1000$ observations, $r = 2$ and $p = 3$. Further, only diagonal covariance structures for both $\Phi_g$ and $\Omega_g$ were considered in Simulation 3. Each of the simulation settings consisted of 25 different datasets. The count range in the simulated datasets closely represented the range observed in the RNA-seq data (Freixas-Coutin et al. 2017). The covariance matrices $\Phi_g$ and $\Omega_g$ for each setting are generated using the clusterGeneration package (Qiu and Joe 2015). Initialization of $z_{ng}$ was done using one hundred different runs of the k-means algorithm. Clustering was performed on each dataset for values $G = 1, \ldots, 5$. We also compared the performance of our approach on count datasets generated from competitive models. For Simulation 4, we generated 25 datasets from a mixture of six independent Poisson distributions with $G = 2$ components, $\pi_1 = 0.45$ and $N = 1000$ and we analyzed the data as a $2 \times 3$ matrix when using mixtures of MVPLN distributions. For Simulation 5, we generated 25 datasets from a mixture of six independent negative binomial distributions with $G = 2$ components, $\pi_1 = 0.79$ and $N = 2000$ and we analyzed the data as a $2 \times 3$ matrix when using mixtures of MVPLN distributions. For Simulation 6, to show performance on a dataset with similar number of components to real data, we generated 25 datasets from a mixture of MVPLN distributions with $G = 8$ components. We set $N = 1500$, $r = 2$, $p = 3$, and $\pi_1 = \cdots = \pi_7 = 0.125$.

Comparative studies were also conducted on datasets from five simulation settings. Because no comparable methods capable of clustering three-way count data were found in the current literature, datasets from Simulations 1, 2, 3, and 6 were vectorized and analyzed with clustering methods designed for two-way data. For comparison purpose, a model-based clustering technique for count data, HTSCluster (Raher et al. 2011, 2015), and method k-means clustering (MacQueen 1967) were used. For HTSCluster, initialization and clustering ranges were same as those used for mixtures of MVPLN algorithm. In a classification EM framework (CEM; Celeux and Govaert 1992), k-means has been shown to be equivalent to an isotropic Gaussian mixture model with equal variance across all components and with equal mixing proportions. Here, we fitted an equal variance isotropic Gaussian mixture model using R package mclust (Scrucca et al. 2016) on the log-transformed data which would be similar to fitting a fuzzy version of the k-means on the latent space and allow us to compute ICL that relies on the estimated soft $Z$. We then utilized model selection criteria to select the optimal number of components.

The clustering results along with ARI values of our proposed method and other comparative methods are provided in Table 1. As evident in Table 1, in all six simulations, our proposed approach was able to recover the underlying cluster in all 25 datasets using both BIC and ICL, including when the Simulations 4 and 5, where the datasets were generated from mixtures of Poisson and negative binomial distributions. However, overfitting was evident with AIC and AIC3 as in various datasets, it favored models with larger number of components (i.e. $G = 3$, 4, and 5 for Simulations 1–5 and $G = 9$ and $G = 10$ for Simulation 6). The AIC and AIC3 penalize log-likelihood only for the number of free parameters in the model and are constant with respect to the sample size. When the number of observations is large, these model selection criteria tend to favor more complex models and are known to overestimate number of components (Shibata 1976; Katz 1981). Note that we only provide the ARI values from the VAG approach. The ARI from the hybrid approach is the same as that from the VAG approach because the cluster membership indicator variable is determined in the VAG step in the hybrid approach. While we provide the estimated parameters for using MCMC approach, due to the extreme computational cost that comes with fitting these models, we only fitted the model with correct number of components, and thus, do not provide the summarized results in Table 1. For each simulation setting, we also provided a measure of cluster separation. For each simulated dataset, we computed the normalized separation index, $I \in [0,1]$, by Hennig (2019) using the R package fpc (Hennig 2020), where higher values indicate that clusters have good separation. As Euclidean distance is used as a distance measure for computing the separation index, and we computed the separation index using the log-transformed data, which would be equivalent to assessing separation in the latent space. The average value of the separation index along with the standard deviation are provided in Table 1. The parameter estimation results for $M_g$, $\Phi_g$, and $\Omega_g$ via the mixtures of MVPLN algorithm for Simulations 1–3 using all three approaches are summarized in Supplementary File 1. Overall, the simulation experiments illustrated that our approach for parameter estimation (Section 2.3) is effective at parameter recovery for the mixtures of MVPLN distributions.

In the two simulation settings where the datasets were generated from mixtures of independent Poisson distributions and mixtures of independent negative binomial distributions (Simulation 4 and 5), our approach was able to recover the underlying cluster structure and estimate the mean and variances of samples fairly well. With regard to HTSCluster, a model with $G = 3$, the highest value for $G$ considered, was selected for in Simulations 1, 2, 3, and 5 and a model with $G = 10$, the highest value for $G$ considered, was selected for in Simulation 6. Furthermore, the ARI values were low across these simulation settings, indicating that observations were not assigned to the correct clusters. However, in simulation setting 4 where the datasets are generated from mixtures of independent Poisson distribution, HTSCluster is able to recover the underlying cluster structure in all 25 datasets using BIC and ICL with perfect classification. It is worth noting that several studies have
Table 1. Mean of normalized cluster separation index (standard deviation) for each simulation setting and the number of clusters selected (average ARI, standard deviation) from all 25 datasets for each simulation setting using different model selection criteria.

| Method          | Simulation setting | Mean of normalized separation index (SD) | BIC          | ICL          | AIC          | AIC3         |
|-----------------|--------------------|----------------------------------------|--------------|--------------|--------------|--------------|
| Mix. of MVPLN   | 1 –                | 1 (1.00, 0.00)                          | 1 (1.00, 0.00) | 1, 2 (0.88, 0.33) | 1 (1.00, 0.00) |
| (VGA-based      | 2 0.44 (0.02)      | 2 (1.00, 0.00)                          | 2 (1.00, 0.00) | 2, 3, 4, 5 (0.89, 0.22) | 2 (1.00, 0.00) |
| approach)       | 3 0.37 (0.02)      | 2 (1.00, 0.00)                          | 2 (1.00, 0.00) | 2, 3, 4, 5 (0.86, 0.21) | 2, 3, 4 (0.96, 0.12) |
|                 | 4 0.69 (0.01)      | 2 (1.00, 0.00)                          | 2 (1.00, 0.00) | 2, 3, 4, 5 (0.89, 0.22) | 2 (1.00, 0.00) |
|                 | 5 0.44 (0.02)      | 2 (1.00, 0.00)                          | 2 (1.00, 0.00) | 2, 3, 4, 5 (0.99, 0.06) | 2 (1.00, 0.00) |
|                 | 6 0.14 (0.01)      | 8 (0.99, 0.01)                          | 8 (0.99, 0.00) | 8, 9, 10 (0.99, 0.01) | 8, 9 (0.99, 0.01) |
| HTSCluster      | 1 –                | 5 (0.00, 0.00)                          | 5 (0.00, 0.00) | 5 (0.00, 0.00) | 5 (0.00, 0.00) |
|                 | 2 0.44 (0.02)      | 5 (0.00, 0.00)                          | 5 (0.00, 0.00) | 5 (0.00, 0.00) | 5 (0.00, 0.00) |
|                 | 3 0.37 (0.02)      | 5 (0.00, 0.00)                          | 5 (0.00, 0.00) | 5 (0.00, 0.00) | 5 (0.00, 0.00) |
|                 | 4 0.69 (0.01)      | 2 (1.00, 0.00)                          | 2 (1.00, 0.00) | 2, 3, 4 (0.97, 0.06) | 2, 3, 4 (0.97, 0.06) |
|                 | 5 0.44 (0.02)      | 3 (0.22, 0.01)                          | 3 (0.22, 0.01) | 3 (0.22, 0.01) | 3 (0.22, 0.01) |
|                 | 6 0.14 (0.01)      | 10 (0.03, 0.13)                         | 10 (0.03, 0.13) | 10 (0.03, 0.13) | 10 (0.03, 0.13) |
| Fuzzy-version of | 1 –                | 5 (0.00, 0.00)                          | 4, 5 (0.00, 0.00) | 5 (0.00, 0.00) | 5 (0.00, 0.00) |
| k-means         | 2 0.44 (0.02)      | 3 (0.24, 0.01)                          | 3 (0.24, 0.01) | 3 (0.24, 0.01) | 3 (0.24, 0.01) |
|                 | 3 0.37 (0.02)      | 3, 4, 5 (0.55, 0.06)                   | 2, 3, 4, 5 (0.81, 0.23) | 4, 5 (0.54, 0.04) | 4, 5 (0.54, 0.04) |
|                 | 4 0.69 (0.01)      | 2 (1.00, 0.00)                          | 2 (1.00, 0.00) | 2, 3, 4 (0.97, 0.06) | 2, 3, 4 (0.97, 0.06) |
|                 | 5 0.44 (0.02)      | 2 (1.00, 0.00)                          | 2 (1.00, 0.00) | 2, 3, 4, 5 (0.82, 0.27) | 2, 3, 4 (0.82, 0.27) |
|                 | 6 0.14 (0.01)      | 8, 9, 10 (0.93, 0.03)                  | 8, 9, 10 (0.96, 0.02) | 10 (0.90, 0.01) | 10 (0.90, 0.01) |

3.2 Clustering transcriptome data

To illustrate the applicability of mixtures of MVPLN distributions for detecting the underlying cluster structure, the VEGA-based approach was applied to an RNA-seq dataset. Typically, only a subset of genes from the experiment are used for cluster analysis, in order to reduce noise. For this analysis, only the differentially expressed genes were used for clustering. Freixas-Coutin et al. (2017) used RNA-seq to monitor the transcriptional dynamics in the seed coats of darkening and nondarkening cranberry beans (Phaseolus vulgaris) at three developmental stages: early, intermediate and mature. The aim of the study was to evaluate if the changes in the seed coat transcriptome were associated with proanthocyanidin levels as a function of seed development in cranberry beans. The RNA-seq data are available on the National Center for Biotechnology Information (NCBI) Sequence Read Archive (SRA) under the BioProject PRJNA380220. The study identified 1336 differentially expressed genes, which were used for clustering. The raw read counts for genes were obtained from Binary Alignment/Map files using samtools (Li et al. 2009) and HTSeq (Anders et al. 2015). The median value from the three replicates per each developmental stage was used. Each observation or gene (the unit) was structured as a 2 × 3 matrix, such that it contained counts for the two varieties (the variables): darkening and nondarkening, across the three developmental stages (the occasions): early, intermediate and mature. On the three-way data of dimensions 1336 × 2 × 3, a clustering range of G = 1,...,10 was considered using k-means initialization (100 runs). Furthermore, we repeated the analysis 10 times. Since BIC and ICL both performed well in recovering the underlying cluster structure in the simulated data, here we also used BIC and ICL for model selection. Both BIC and ICL selected a model with G = 8. In this model, Clusters 1–8 were composed of 206 (15.4%), 161 (12.2%), 104 (7.8%), 162 (12.1%), 126 (9.4%), 147 (11.0%), 162 (12.3%), and 266 (19.9%) genes, respectively. See Supplementary File 2 for gene composition of each cluster. Expression patterns across the clusters were visualized using a heatmap. The log-transformed expression patterns of the clusters are illustrated using the heatmap in Fig. 4.

In simulation studies where the true labels are available and ARI of true and predicted class label can be used to assess clustering performance. Because no true labels are available in real data, we use visualization of the data using heatmap to assess the trends in the clusters obtained on the transcriptome data in Fig. 4. We also visualize the cluster-specific μg which relates to mean trends of the expression levels of genes in different clusters in Fig. 5. As evident in Figs 4 and 5, each cluster has its distinctive expression signatures. In some clusters (for example, Cluster 4), the means of the gene expression signatures are similar between the darkening and nondarkening beans but their mean expression pattern varies between developmental stages; whereas, in Cluster 8, the means of the gene expression signatures are similar across the developmental stages but varies slightly between the darkening and nondarkening beans. On the other hand, in Cluster 3, the mean gene expression signatures vary across both the developmental stages as well as between darkening and nondarkening beans.

For all simulation and transcriptome data analyses, the normalization factors representing library sizes for samples were obtained using the trimmed mean of M values from calcNormFactors function of edgeR package (Robinson and Oshlack 2010; McCarthy et al. 2012). A table of mathematical notation is provided in Supplementary File 4 and a table of abbreviations is provided in Supplementary File 5.

4 Discussion

A mixture of MVPLN distributions is introduced for clustering three-way count data, targeted at expression data arising from RNA-seq experiments. This is the first use of a mixture of MVPLN distributions for clustering within the literature. By allowing for a direct analysis of three-way data structures, matrix variate distributions permit the estimation of correlations within and between variables and occasions. This makes them very attractive for analyzing...
matrix data in the context of clustering. Further, by considering a matrix variate structure, the number of free covariance parameters to be estimated is greatly reduced under high dimensional settings. Herein, three different parameter estimation frameworks are proposed: an approach based on MCMC, one based on VGA, and one based on a hybrid approach. When posterior inference is of interest, the MCMC-based approach is favorable but it can be computationally intensive. On the other hand, the VGA-based approach only approximates the posterior distribution that relies on approximation but it is computationally efficient. Therefore, here we also propose a hybrid approach that is computationally efficient and it samples from the true posterior. Through simulation studies, we show that the VGA approach provides good clustering performance even when the datasets are generated from mixtures of other discrete distributions.

Using simulated data, it was illustrated that the algorithm for mixtures of MVPLN distributions is effective and returned favorable clustering results. The transcriptome data analysis showed the applicability of the mixture model-based clustering method on RNA-seq count data. A possible future direction of this work would be to make use of subspace clustering methods and to develop the matrix variate factor analyzers model. This would permit clustering of data in low-dimensional subspaces as high-dimensional RNA-seq datasets become frequent. Another path is to consider restrictions on the matrices $\Phi_g$ and $\Omega_g$, as done by Viroli (2011). Also, constraints on $\Phi_g$ similar to those introduced by McNicholas and Murphy (2010), and used by Anderlucci and Viroli (2015), could be beneficial when analyzing longitudinal RNA-seq data.

Acknowledgements
The authors acknowledge the computational support by Dr Marcelo Ponce at the SciNet HPC Consortium, University of Toronto, ON, Canada. This research was enabled in part by support provided by Research Computing Services (https://carleton.ca/rcs) at Carleton University.

Supplementary data
Supplementary data are available at Bioinformatics online.

Conflict of interest
None declared.

Funding
This work was supported by Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) [to S.S. Grant no. RGPIN-2021-03812 and to P.M. Grant no. RGPIN-2021-06030], Canada Research Chairs program [to S.S. Grant no. 2020-00303 and P.M. Grant no. 950-230797, respectively], Queen Elizabeth II Graduate Scholarships in Science & Technology [to A.S.], Ontario Graduate Fellowship [to A.S.], and an E.W.R. Steacie Memorial Fellowship from NSERC [to P.M. Grant no. 537596-2019]. No funding body played a role in the design of the study, analysis and interpretation of data, or in writing the manuscript.

Data availability
The RNA-seq data used in the manuscript are publicly available on the National Center for Biotechnology Information (NCBI) Sequence Read Archive (SRA) under the BioProject PRJNA380220. The GitHub R package for this work is available at https://github.com/anjalisilva/mixMVPLN.

References
Aitchison J, Ho CH. The multivariate Poisson-log normal distribution. Biometrika 1989;76:641–53.
Akaike H. Information theory and an extension of the maximum likelihood principle, In: Petrov B.N. and Csaki F. (eds) Second International Symposium on Information Theory, New York: Springer Verlag, 1973, 267–81.
Anderlucci L, Viroli C. Covariance pattern mixture models for the analysis of multivariate heterogeneous longitudinal data. Ann Appl Stat 2015;9: 777–800.
Anders S, Pyl PT, Huber W. HTSeq—a python framework to work with high-throughput sequencing data. Bioinformatics 2015;31:166–9.
