9-26-2023 2:30 PM

Parameter Estimation for Normally Distributed Grouped Data and Clustering Single-Cell RNA Sequencing Data via the Expectation-Maximization Algorithm

Zahra Aghahosseinalishirazi, *Western University*

Supervisor: De Souza, Camila, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Statistics and Actuarial Sciences

Abstract

The Expectation-Maximization (EM) algorithm is an iterative algorithm for finding the maximum likelihood estimates in problems involving missing data or latent variables. The EM algorithm can be applied to problems consisting of evidently incomplete data or missingness situations, such as truncated distributions, censored or grouped observations, and also to problems in which the missingness of the data is not natural or evident, such as mixed-effects models, mixture models, log-linear models, and latent variables.

In Chapter 2 of this thesis, we apply the EM algorithm to grouped data, a problem in which incomplete data are evident. Nowadays, data confidentiality is of great importance for many companies and organizations. For this reason, they may prefer not to release exact data but instead to grant researchers access to approximate data. For example, rather than providing the exact measurements of their clients, they may only provide researchers with grouped data, that is, the number of clients falling in each of a set of non-overlapping measurement intervals. The challenge is to estimate the mean and variance structure of the hidden ungrouped data based on the observed grouped data. To tackle this problem, this work considers the exact observed data likelihood and applies the EM and Monte-Carlo EM (MCEM) algorithms for the cases where the hidden data follow a univariate, bivariate, or multivariate normal distribution. The well-known Galton data and simulated datasets are used to evaluate the statistical properties of the proposed EM and MCEM algorithms.

In Chapters 3, 4 and 5, we apply the EM algorithm to a case in which the missingness of the data is not evident by considering mixture models and latent variables to propose a novel model-based clustering approach for single-cell RNA sequencing data. In biology, cells can be distinguished by their phenotype, such as size and shape, or at the molecular level, based on their genome, epigenome, and transcriptome. In this thesis, we focus on the transcriptome, which includes all RNA transcripts in a given cell population, indicating the genes being expressed at a certain time. We consider single-cell RNA sequencing data and develop a novel model-based clustering method to group cells based on their transcriptome profiles. The proposed clustering approach takes into account the large proportion of zeros present in the data, which can be either true biological zeros or technological noise. The assumed model for clustering is a mixture of either zero-inflated Poisson or zero-inflated negative binomial distributions, and inference is conducted via the EM algorithm. The performance of the proposed methodology is evaluated via simulation studies and analyses of published real datasets.

Lay Summary

The Expectation-Maximization (EM) algorithm has many applications in Statistics for estimation purposes. In this thesis, we study the application of the EM algorithm from two perspectives. One, for the situations in which the incomplete data are evident, and the other for the cases that missingness of data is not evident.

In Chapter 2, we consider the application of the EM algorithm when the missingness in the data is evident such as grouped data, in which we know the intervals of the data and the frequencies over each interval. However, the exact raw data are not available. Assuming that the data follow a normal distribution, we find the mean and variance estimates of the normal distribution by applying the EM algorithm framework. We consider the cases of univariate, bivariate, and multivariate normal grouped data. We evaluate the performance of the proposed EM framework with simulated data and a publicly available dataset.

In the Chapters 3, 4, and 5, we study another application of the EM algorithm in which the incomplete data is not evident such as mixture models. We consider the finite mixtures of zero-inflated models to cluster cells based on their gene expression profiles by applying the EM algorithm to estimate the model parameters. Our proposed clustering approach considers the large proportion of zeros in the data, which can be either true biological zeros or technological noise. Simulation studies are implemented to evaluate the performance of our proposed method under different controlled scenarios. We also analyze publicly available biological datasets as examples of applications.

Co-Authorship Statement

I declare that this thesis incorporates materials that are the results of joint research conducted from September 2017 to present by myself as the main author under the supervision of Dr. Camila P. E. de Souza in the Department of Statistical and Actuarial Sciences at the University of Western Ontario.

The work presented in Chapter 2 is the accepted version of a manuscript published in the journal "Communication in Statistics-Simulation and Computation" by Zahra Aghahosseinalishirazi (student, main author), da Silva, J. P. (co-author), and Dr. Camila P. E. de Souza (supervisor). The manuscript title is "Parameter Estimation for Grouped Data Using EM and MCEM Algorithms." Dr. Camila P. E. de Souza presented the research problem, aided the development of the methodology and analysis, and contributed to reviewing, improving, and providing feedback on the text. João Pedro A. R. da Silva contributed to coding and running some of the data analyses. I certify that as the main author, I was responsible for all direct aspects of the research, including research question formulation, literature review, model formulation, coding, analysis, and preparing the first and final version of the manuscript. Full citation: Zahra AghahosseinaliShirazi, João Pedro A. R. da Silva & Camila P. E. de Souza (2022) Parameter estimation for grouped data using EM and MCEM algorithms, Communications in Statistics - Simulation, https://doi.org/10.1080/03610918.2022.2108843

The work presented in Chapters 3, 4, and 5 is co-authored with Pedro Assunção Rangel (co-author) and Dr. Camila P. E. de Souza (supervisor) and has not yet been submitted to a journal. Dr. Camila P. E. de Souza presented the research problem, aided the development of the methodology and analysis, and contributed to reviewing, improving, and providing feedback on the text. Pedro Assunção Rangel contributed to coding and running analyses using Sharcnet's Graham computer cluster. I certify that as the main author, I was responsible for all direct aspects of the research, including research question formulation, literature review, model formulation, coding, analysis, and writing.

Acknowledgments

It is a genuine pleasure to express my deep sense of thanks and gratitude to my supervisor, Dr. Camila de Souza, for her invaluable help, support, and guidance through this research and deeply inspired me with her motivation and sincerity. I learned a lot from her, not only in Statistics but also in dealing with the challenges in this journey.

I would like to thank Dr. Grace Yi, Dr. Hanna Jankowski, Dr. Yalda Mohsenzadeh, and Dr. Jay Gweon for agreeing to serve as members of my dissertation committee.

Another thank you to the professors in the Department of Statistical and Actuarial Sciences, Western University, for their support. Thank you, Pedro Assunção Rangel and João Pedro A. R. da Silva for the co-author contribution.

I am particularly grateful to my father, Mahmoud A.Shirazi, my mother, Simin Barkh, and my sisters, Maryam A.Shirazi and Mahsa A.Shirazi, for all their invaluable spiritual and moral support and encouragement during the tough and challenging days of this journey.

Finally, I also would like to give special thanks to my mentors and friends, Dr. Alireza Pirkhaefi, Dr. Farzaneh Safamivamesh, my friends Pegah Kebritchi, and Neda Mahali, and my aunt Soraya Barkh for their constant inspiration and support.

Contents

Bibliography **195**

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Overview

The Expectation-Maximization (EM) algorithm (Dempster et al., 1977, McLachlan and Krishnan, 2008) is an iterative optimization algorithm for solving maximum likelihood (ML) estimation problems in the presence of missing data or latent variables. The EM algorithm is an alternative to numerical optimization algorithms such as Newton-Raphson. Each iteration of the EM algorithm comprises two steps, the Expectation step (E-step) and the Maximization step (M-step), demonstrating the reason for calling the algorithm the EM algorithm. The idea of the EM algorithm is reformulating and associating the given incomplete-data problem with a more straightforward complete-data problem for which the computation of ML estimates is more amendable; i.e. the ML estimates of the complete-data problem have closed-form or can be computed by using the standard computer optimization packages. The E-step of the EM algorithm includes taking expectations of the complete-data log-likelihood given the observed data and current parameter values. Then, by replacing the unobserved data with its conditional expectations given the observed data and the current state of the parameters, the ML estimates of the complete data in the M-step can be easily computed. Starting from appropriate initial values, the E-step and M-step are repeated in each iteration of the EM algorithm until convergence. The parameter estimates obtained at the convergence of the EM algorithm are the ones that maximize the observed data log-likelihood. More details about the EM algorithm and its convergence property are presented in Section 1.3.

The EM algorithm can be applied to problems consisting of evidently incomplete data or missingness situations, such as truncated distributions, censored or grouped observations, and also to problems in which the missingness of the data is not natural or evident, such as mixed-effects models, mixture models, log-linear models, and latent class and latent variable structures (Dempster et al., 1977, Givens and Hoeting, 2013, McLachlan and Krishnan, 2008).

In this thesis, we consider applications of the EM algorithm framework to both cases: when the incomplete data situation is evident (grouped data studied in Chapter 2), and it is not evident (zero-inflated mixture models studied in Chapters 3, 4 and 5).

As mentioned above, one application of the EM algorithm is for situations in which incomplete data are evident, such as grouped observations or grouped data (also referred to as interval-based data). Some authors have used the EM algorithm to deal with the problem of parameter estimation for grouped data from different contexts (Cadez et al., 2002, Heitjan, 1991, McLachlan and Jones, 1988, Teimouri, 2020). However, none of these authors has presented the exact formulae of EM parameter estimates for the bivariate and multivariate normal grouped data. In addition, no previous study has considered the Monte-Carlo EM algorithm, which is an extension of the traditional EM, for estimating the parameters of normally grouped data for univariate, bivariate, and multivariate cases. Therefore, in Chapter 2 of this thesis, we develop a comprehensive approach to the parameter estimation of the normally distributed grouped data (univariate, bivariate, and multivariate grouped data) by applying the exact form of the likelihood and then finding the parameter estimates via 1) numerical optimization using the Newton-Raphson's algorithm (we call this approach exact MLE), 2) EM algorithm and the 3) Monte-Carlo EM (MCEM) algorithm. We also compute the standard errors of the EM and MCEM estimates within the EM framework for the mean parameters. We then apply and compare the performance of the proposed EM and MCEM algorithms with that of the exact MLE approach on both real and simulated data. For the real data application, we use the well-known Galton data (Galton, 1889) and calculate the estimated parameters using each of the three approaches, including exact MLE, EM and MCEM, when the data are considered as normally distributed grouped data for both univariate and bivariate cases. Simulation studies under various scenarios, including varying sample sizes and number of bins (intervals), are implemented to evaluate the statistical properties (bias and variance) of the proposed parameter estimates obtained by the EM and MCEM algorithms and compare them with the ones obtained from the exact MLE approach.

Another application of the EM algorithm is for cases in which the missingness of the data is not natural or evident, such as mixture models, which is the focus of Chapters 3, 4, and 5 of this thesis. In these chapters, we propose and present the results of a novel model-based clustering algorithm for zero-inflated count data motivated by the problem of clustering single-cell RNA-sequencing (scRNA-seq) data based on their transcriptome profiles. scRNA-seq data consist of a matrix where rows correspond to cells and columns to genes (or vice-versa) so that the (i, j) entry of the matrix contains the number of sequencing reads (read counts) aligned to the genomic coordinates of gene j in cell i. One important characteristic of scRNA-seq data is the excess of zeros (zero inflation), mainly due to biological zeros (non-expressed genes)

or technological noises (e.g., sequencing errors). Some studies have proposed model-based approaches to cluster or classify scRNA-seq data (Ji and Ji, 2016, Liu et al., 2019, Prabhakaran et al., 2016, Sun et al., 2018, Zhang et al., 2019). However, none of these studies has dealt with the zero-inflation feature of these data. Therefore, to cluster the cells of the scRNA-seq data, we consider mixtures of either zero-inflated Poisson (ZIP) or zero-inflated Negative Binomial (ZINB) distributions, which take into account the excess of zeros for this type of data. First, in Chapter 3, we describe the proposed EM algorithm to infer the cell-specific cluster assignments and model parameters for the zero-inflated Poisson and the zero-inflated negative binomial mixture models, considering the cases without covariates and with covariates. Then, in Chapter 4, we present the results of investigating the performance of our proposed model-based clustering approaches under a variety of simulation scenarios, including varying the number of cells (N), the number of genes (G), and the number of clusters (K), and also varying some other model parameters. Finally, in Chapter 5, we apply the proposed models to two publicly available data sets and compare the fitted models using the Aikaike Information Criterion (AIC).

1.2 Thesis Organization

This thesis consists of six chapters. Maximum likelihood estimation, the Newton-Raphson method and the Expectation-Maximization (EM) algorithm are briefly introduced in Sections 1.3, 1.4, and 1.5 in Chapter 1 since these inference techniques are used in Chapters 2 and 3. In Chapter 2, we address the problem of estimating the parameters of grouped data when they are normally distributed (both univariate and multivariate cases) using the EM and Monte Carlo EM (MCEM) algorithms (McLachlan and Krishnan, 2008). Chapter 2 corresponds to the accepted version of a published manuscript, for which the full citation is: Aghahosseinalishirazi, Z., da Silva, J. P., de Souza, C. P. E., Parameter Estimation for Grouped Data Using EM and MCEM Algorithms. (Aug 2022) Journal of Communication in Statistics-Simulation and Computation, hppt://doi.org/10.1080/03610918.2022.2108843. Chapter 3 describes our proposed model-based approach to cluster single-cell RNA sequencing data. Our model is based on a mixture of zero-inflated distributions (Poisson or negative binomial), and parameter inference is done via EM. In Chapter 4, simulation studies are conducted under various scenarios to evaluate the performance of the proposed zero-inflated Poisson and negative binomial (ZIP and ZINB) mixture models. In Chapter 5, the proposed model-based clustering algorithms introduced in Chapter 3 are applied to published data sets. Chapter 6 presents the conclusion and possible directions for future research work.

1.3 Maximum Likelihood Estimation

Let Y be a p-dimensional random vector with probability density function (or probability mass function) $f(y|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^T$ is a vector of d unknown parameters. To find the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}$, we consider the likelihood function of an observed random sample $y = (y_1^T, \ldots, y_n^T)^T$ of size n on the random vector Y, $L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(y_i|\boldsymbol{\theta})$, which is a function of $\boldsymbol{\theta}$ with each y_i fixed. Thus, we find $\hat{\boldsymbol{\theta}}$ as a solution of:

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0$$

or equivalently, the solution of:

$$\frac{\partial \log L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0.$$

Let $S(\boldsymbol{\theta}; y) = \frac{\partial \log L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ be the gradient vector (the score statistic) with the first-order derivatives of the log-likelihood function with respect to the parameter $\boldsymbol{\theta}$, and the Hessian matrix with second-order derivatives be denoted as:

$$H(\boldsymbol{\theta}; y) = \frac{\partial^2 \log L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}.$$

Under certain regularity conditions (Casella and Berger, 2002, McLachlan and Krishnan, 2008), the Fisher information matrix $\mathcal{I}(\boldsymbol{\theta})$ is given by:

$$\mathcal{I}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\big[S(\boldsymbol{\theta}; y) \times S^T(\boldsymbol{\theta}; y)\big] = -E_{\boldsymbol{\theta}}\big[H(\boldsymbol{\theta}; y)\big]. \tag{1.1}$$

By the asymptotic properties of maximum likelihood estimation (Casella and Berger, 2002, McLachlan and Krishnan, 2008), the standard error of θ_i can be approximated by:

$$\mathrm{SE}[(\hat{\theta}_i)] \approx (\mathcal{I}^{-1}(\hat{\boldsymbol{\theta}}))_{ii}^{\frac{1}{2}}$$

for $i = 1, \ldots, d$, where the notation $(A)_{ij}$ is denoting the element in the i-th row and j-th column of matrix A. It is common to further approximate $\mathrm{SE}(\hat{\theta}_i)$ using the observed information matrix $I(\hat{\boldsymbol{\theta}}; y) = -H(\hat{\boldsymbol{\theta}}; y)$ instead of the information matrix $\mathcal{I}(\boldsymbol{\theta})$ evaluated at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ (McLachlan and Krishnan, 2008).

Often, in practice, the MLE of the log-likelihood function cannot be found analytically and we have to compute $\hat{\boldsymbol{\theta}}$ as the MLE of $\boldsymbol{\theta}$ iteratively by using Newton-Raphson maximization algorithm or alternatively Expectation-Maximization (EM) algorithm. Both Newton-Raphson (NR) and EM algorithms are considered algorithms for finding the zeros of a function (McLachlan and Krishnan, 2008).

1.4 Newton-Raphson Method

The Newton-Raphson iterative method can be used for solving the likelihood equation (or alternatively the log-likelihood equation) $S(y;\boldsymbol{\theta}) = 0$. Using the linear Taylor series expansion about the current estimate $\boldsymbol{\theta}^{(k)}$ for $\boldsymbol{\theta}$, we can approximate the gradient vector as follows:

$$S(\boldsymbol{\theta};y) \approx S(\boldsymbol{\theta}^{(k)};y) + H(\boldsymbol{\theta}^{(k)};y)(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}) \tag{1.2}$$

Taking the right-hand side of (1.2) to be equal to zero leads to the new (updated) fit of $\boldsymbol{\theta}^{(k+1)}$ as:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - H^{-1}(\boldsymbol{\theta}^{(k)};y)S(\boldsymbol{\theta}^{(k)};y). \tag{1.3}$$

In practice,$-H(\boldsymbol{\theta}^{(k)};y)$ in (1.3) can be replaced by $\mathcal{I}^{-1}(\boldsymbol{\theta}^{(k)})$ in (1.1), thus resulting in the Fisher scoring update:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \mathcal{I}^{-1}(\boldsymbol{\theta}^{(k)};y)S(\boldsymbol{\theta}^{(k)};y).$$

When the log-likelihood function is a concave, unimodal, and quadrative function of $\boldsymbol{\theta}$, then the sequence of iterates $(\boldsymbol{\theta}^{(k)})$ converge to the MLE of $\boldsymbol{\theta}$ in one step; however, if the log-likelihood function is not concave, then the convergence of the Newton-Raphson algorithm from an arbitrary starting value cannot be guaranteed. In this case, under reasonable assumptions of $L(\boldsymbol{\theta})$ and choosing an appropriately accurate starting value, then the Newton-Raphson sequence of iterates $(\boldsymbol{\theta}^{(k)})$ has local quadratic convergence to a solution $\boldsymbol{\theta}^*$ of the equation $S(y;\boldsymbol{\theta}) = 0$. That means, given a norm $||.||$ on the parameter space, for a $\boldsymbol{\theta}^{(0)}$ sufficiently close to $\boldsymbol{\theta}^*$, there exists a constant ξ such that:

$$||\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*|| \leq \xi||\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*||^2$$

for all $k = 0, 1, 2, \ldots$. See McLachlan and Krishnan (2008) for more details.

1.5 Expectation-Maximization (EM) algorithm

Expectation-Maximization (EM) is an iterative method based on the maximum likelihood estimation framework when the observations are considered as incomplete data (Dempster et al., 1977). Indeed, it is assumed that the complete data are generated from a random variable $Y = (X, Z)$, where X is used for generating the observed (incomplete) data and Z for the missing or latent data. EM can be applied in the presence of both continuous or discrete random variables. In what follows, we describe the EM algorithm considering continuous variables and their density functions; however, the same results can be obtained for discrete variables with

their probability mass functions and integrals replaced by sums. The density of the missing data given the observed data can be written as follows:

$$f_{Z|X}(z|x,\boldsymbol{\theta}) = \frac{f_Y(y|\boldsymbol{\theta})}{f_X(x|\boldsymbol{\theta})},$$

where the observed data density is

$$f_X(x|\boldsymbol{\theta}) = \int_y f_Y(y|\boldsymbol{\theta})dy.$$

Let $L(\boldsymbol{\theta}|x)$ be the observed data likelihood, that is, $f_X(x|\boldsymbol{\theta})$ when x is fixed and we vary Θ. The goal is maximizing $L(\boldsymbol{\theta}|x)$ (or equivalently $\log L(\boldsymbol{\theta}|x)$) with respect to $\boldsymbol{\theta}$ using the EM algorithm to find $\hat{\boldsymbol{\theta}}$. We define $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ as the conditional expectation of the complete-data log-likelihood, $\log L(\boldsymbol{\theta}|y)$, given the observed data x and the current set of parameter estimates $\boldsymbol{\theta}^{(t)}$, that is:

$$\begin{aligned} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) &= E\Big\{ \log L(\boldsymbol{\theta}|y)\Big|x,\boldsymbol{\theta}^{(t)}\Big\} \\ &= E\Big\{ \log f_Y(y|\boldsymbol{\theta})\Big|x,\boldsymbol{\theta}^{(t)}\Big\} \\ &= \int \Big[\log f_Y(y|\boldsymbol{\theta})\Big] f_{Z|X}(z|x,\boldsymbol{\theta}^{(t)})dz. \end{aligned} \tag{1.4}$$

Starting from a initial value $\boldsymbol{\theta}^{(0)}$, the EM algorithm iterates between the Expectation (E) and Maximization (M) steps until convergence as follows:

1) **E-Step:** Compute $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$;

2) **M-Step:** Obtain the updated estimate $\boldsymbol{\theta}^{(t+1)}$ by maximizing $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ w.r.t. $\boldsymbol{\theta}$;

3) Return to Step 1 until convergence.

The convergence criterion can be built upon $\big[\log L(\boldsymbol{\theta}^{(t+1)}|x) - \log L(\boldsymbol{\theta}^{(t)}|x))\big]$.

At each step of the EM algorithm, the likelihood increases. Therefore, the parameter estimates $\hat{\boldsymbol{\theta}}$ obtained at the convergence of the EM algorithm are the ones that maximize $\log L(\boldsymbol{\theta}|x)$. In fact, using the EM algorithm for the problem of finding MLEs is so popular because of the simplicity in implementing the algorithm and being able to reach the global optimum after some uphill steps. To show this convergence property of the EM algorithm, we first express the observed data log-likelihood as:

$$\log L(\boldsymbol{\theta}|x) = \log f_Y(y|\boldsymbol{\theta}) - \log f_{Z|X}(z|x,\boldsymbol{\theta}). \tag{1.5}$$

Then by taking expectation from both sides of (1.5) with respect to the distribution of $Z|x, \boldsymbol{\theta}^{(t)}$, we have:

$$E\Big\{ \log L(\boldsymbol{\theta}|x)\Big|x, \boldsymbol{\theta}^{(t)}\Big\} = E\Big\{ \log f_Y(y|\boldsymbol{\theta})\Big|x, \boldsymbol{\theta}^{(t)}\Big\} - E\Big\{ \log f_{Z|X}(z|x, \boldsymbol{\theta})\Big|x, \boldsymbol{\theta}^{(t)}\Big\}$$

which leads to

$$\log L(\boldsymbol{\theta}|x) = Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) - G(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}),$$

where $G(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = E\Big\{ \log f_{Z|X}(z|x, \boldsymbol{\theta})\Big|x, \boldsymbol{\theta}^{(t)}\Big\}$.

Now, to show that the observed data log-likelihood increases as $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ increases in each EM iteration, we need to show that the G function decreases (or stays at the same value) for any $\boldsymbol{\theta} \neq \boldsymbol{\theta}^{(t)}$. By applying Jensen's inequality, we obtain:

$$\begin{aligned} G(\boldsymbol{\theta}^{(t)}; \boldsymbol{\theta}^{(t)}) - G(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) &= E\left\{ \log f_{Z|X}(z|x, \boldsymbol{\theta}^{(t)}) - \log f_{Z|X}(z|x, \boldsymbol{\theta})\Big|x, \boldsymbol{\theta}^{(t)}\right\} \\ &= \int -\log \frac{f_{Z|X}(z|x, \boldsymbol{\theta})}{f_{Z|X}(z|x, \boldsymbol{\theta}^{(t)})} f_{Z|X}(z|x, \boldsymbol{\theta}^{(t)}) dz \\ &\geq -\log \int f_{Z|X}(z|x, \boldsymbol{\theta}) dz = -\log 1 \\ &= 0. \end{aligned}$$

Then, choosing $\boldsymbol{\theta}^{(t+1)}$ as the maximizer of $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ with respect to $\boldsymbol{\theta}$, we have that:

$$\log L(\boldsymbol{\theta}^{(t+1)}|x) \geq \log L(\boldsymbol{\theta}^{(t)}|x)$$

as Q increases and G decreases (Dempster et al., 1977, Givens and Hoeting, 2013, McLachlan and Krishnan, 2008).

Sometimes the M-step of the EM algorithm is complicated and, therefore, Meng and Rubin (1993) propose an extension of the EM algorithm, which they call the Expectation-Conditional Maximization (ECM) algorithm. The idea of this extension is to simplify the M-step by undertaking the maximization conditionally on some of the parameters (or functions of the parameters). For example, suppose a parameter set $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$; one can apply the ECM algorithm by replacing the M-step with two CM steps. In the first CM step, fixing $\boldsymbol{\theta}_2^{(t)}$ at its current value, we find the new $\boldsymbol{\theta}_1^{(t+1)}$, then, in the second CM step, the new/updated estimate of $\boldsymbol{\theta}_2^{(t+1)}$ is found by fixing $\boldsymbol{\theta}_1$ at its new estimate $\boldsymbol{\theta}_1^{(t+1)}$. More details about the ECM algorithm can be found in McLachlan and Krishnan (2008).

Moreover, sometimes, it might be challenging to compute $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ in the E-step of the EM algorithm. In that situation, we can simulate a sample of size m of the missing data (or

latent variables) $z_1^{(t)}, \ldots, z_m^{(t)}$ from its conditional distribution $f(z|x;\boldsymbol{\theta}^{(t)})$ and replace $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ by its Monte-Carlo approximation given by:

$$\hat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) = \frac{1}{m}\sum_{j=1}^{m} \log L_c(\boldsymbol{\theta}; x, z_j^{(t)}).$$

Thus, the M-step at iteration $(t+1)$ can be computed by maximizing $\hat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ with respect to $\boldsymbol{\theta}$. This alternative version of the EM algorithm is called Monte Carlo EM (MCEM) algorithm (Givens and Hoeting, 2013, McLachlan and Krishnan, 2008, Wei and Tanner, 1990a,b).

Chapter 2

Parameter Estimation for Grouped Data Using EM and MCEM Algorithms

2.1 Introduction

Nowadays, protecting data confidentiality, security, and integrity is of great importance for governments, organizations, and companies (Chen and Miljkovic, 2018, Huang et al., 2016, Minoiu and Reddy, 2009, Wu and Perloff, 2007). For these reasons, these institutions might not release exact raw data to researchers, analysts, or even the public. Rather, they prefer to release data such as household income, house prices, insurance losses, profits, and age in an interval format. The interval format can contain either grouped data (Velez and Correa, 2015) or symbolic data (Bock and Diday, 2000). This work focuses on grouped data, where for a particular variable, only the intervals and the frequency of observations falling into each interval are known. Table 2.1 shows how univariate grouped data can be represented.

Table 2.1: Univariate grouped data representation.

Interval	Frequencies
$[a_0, a_1)$	n_1
$[a_1, a_2)$	n_2
$\vdots$	$\vdots$
$[a_{k-1}, a_k)$	n_k
Total	n

As can be seen from the grouped data representation in Table 2.1, these data are histogram-based and, therefore, continuous. Continuous data can follow different distributions, including normal, log-normal, and Weibull.

Many studies have been conducted on grouped data from different perspectives. Tallis (1967) has obtained approximate maximum likelihood estimates of the parameters for univariate and multivariate grouped data. Stewart (1983) has dealt with the problem of estimating the parameters of a linear model using data in which the dependent variable is only observed to fall in certain intervals on a continuous scale, with its actual values remaining unobserved. McLachlan and Jones (1988) have considered the fitting of finite mixture models to univariate grouped and truncated data using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977, McLachlan and Krishnan, 2008). Heitjan (1989) has considered Bayesian methods to analyze this type of data. In another study, Heitjan (1991) has applied Newton-Raphson's method and the EM algorithm to find parameter estimates of bivariate regression analysis for grouped data. Cadez et al. (2002) have extended the work in McLachlan and Jones (1988) to multivariate grouped data by using numerical techniques to evaluate the multidimensional integrals at each iteration of the EM algorithm. Wengrzik and Timm (2011) have studied the performance of different methods for fitting a two-component Gaussian mixture model to univariate grouped data. Velez and Correa (2015) have estimated the mean, variance, and coefficient of variation for univariate grouped data using their proposed bootstrap method. More recently, Teimouri (2020) has applied the EM algorithm on univariate grouped data arising from a mixture of skew-normal distributions.

The aim of this study is to find the parameter estimates for grouped data when they are normally distributed for the univariate, bivariate, and multivariate cases using the exact form of the likelihood. Therefore, the estimation approach of McLachlan and Jones (1988) and McLachlan and Krishnan (2008) for univariate grouped data with missing counts is considered and extended to the univariate, bivariate, and multivariate cases without missing counts using both the EM and Monte Carlo EM (Wei and Tanner, 1990a,b) algorithms. To the authors' knowledge, no other study has yet presented the exact formulae of EM parameter estimates for the bivariate and multivariate normal grouped data, as is done in this work. This work also contains the formulae to obtain standard errors for the EM and Monte Carlo EM (MCEM) mean estimates. In summary, three possible approaches for parameter estimation of grouped data are presented: 1) maximum likelihood estimation (MLE) by numerical optimization of the exact grouped data likelihood (Exact MLE), 2) maximizing the exact likelihood using the EM algorithm, and 3) same as (2), but using the MCEM algorithm. All three methods are implemented in R and available at `https://github.com/desouzalab/infgrouped`.

This study is organised as follows. In Section 2.2, the estimation methods for grouped data are presented. In Section 2.2.1, univariate normal grouped data are considered, and parameter estimates are provided for the three methods described in the previous paragraph. In Section 2.2.2, the proposed methods are applied to bivariate grouped data and extended to multivariate

normal grouped data. Standard errors for the EM and MCEM mean estimates are presented in Section 2.2.3. Section 2.3 deals with numerical applications. In Section 2.3.1, the proposed methods are applied to the well-known Galton data (Galton, 1889). Simulation studies for univariate and bivariate normal grouped data are described in Section 2.3.2. Finally, in Section 2.4, results and conclusions are discussed.

2.2 Methods

2.2.1 Univariate Normal Grouped Data

Exact MLE

It is assumed that the unobserved data come from a normal distribution with parameters $\theta = (\mu, \sigma)$ and denoted by $N(\mu, \sigma)$. Let $f(x; \theta)$ be the density function of $N(\mu, \sigma)$. According to $k+1$ pre-established partitioned points $a_0 < a_1 < \cdots < a_{k-1} < a_k$, let n_i be the number of observations that fall into the interval $\mathcal{X}_i = [a_{i-1}, a_i)$ for $1 \leq i \leq k$, $a_0 = -\infty$ and $a_k = +\infty$. Furthermore, it is assumed that the observed data $y = \{n_1, \ldots, n_k\}$ follow a multinomial distribution with $n = \sum_{i=1}^{k} n_i$ draws over k categories (intervals), with the probability of being in category i equal to $P_i(\theta)/P(\theta)$, where

$$P_i(\theta) = \int_{a_{i-1}}^{a_i} f(x; \theta) dx,$$

with $P(\theta) = \sum_{i=1}^{K} P_i(\theta) = 1$. Therefore, the log-likelihood function for the observed data y (also called the incomplete-data log-likelihood) can be written as:

$$\log L(\theta) = \sum_{i=1}^{k} n_i \log P_i(\theta) + C. \tag{2.1}$$

Let $\phi(\cdot)$ and $\Phi(\cdot)$ be the density and the cumulative distribution function (CDF), respectively, of a standard normal distribution. Therefore, the density of $N(\mu, \sigma)$ can be written as:

$$f(x; \mu, \sigma) = \frac{1}{\sigma} \phi\left(\frac{x - \mu}{\sigma}\right),$$

where $-\infty < \mu < \infty$ and $\sigma > 0$. By applying the reparametrization $\theta_1 = \mu/\sigma$ and $\theta_2 = 1/\sigma$, the parameters are changed from $\theta = (\mu, \sigma)$ to $\theta = (\theta_1, \theta_2)$. Now let the CDF of $N(\mu, \sigma)$ be $\Phi(\theta_2 t - \theta_1)$. Then the log-likelihood in (2.1) can be written as a function of θ_1 and θ_2 as follows

(see also Xia et al. (2009)):

$$\begin{aligned}\log L(\theta) &= n_1 \ln\Big[\Phi(\theta_2 a_1 - \theta_1)\Big] + n_k \ln\Big[1 - \Phi(\theta_2 a_{k-1} - \theta_1)\Big] + \\ &\quad \sum_{i=2}^{k-1} n_i \ln\Big[\Phi(\theta_2 a_i - \theta_1) - \Phi(\theta_2 a_{i-1} - \theta_1)\Big] + C, \end{aligned} \tag{2.2}$$

where C is a constant term that does not depend on θ. This reparametrization does not affect the results because of the invariance property of maximum likelihood estimators (Casella and Berger, 2002).

The parameter estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ can be obtained by maximizing (2.2) with respect to $\theta = \{\theta_1, \theta_2\}$ using Newton-Raphson numerical methods such as those implemented in the *optim()* function in R.

Parameter Estimation via the EM Algorithm

In a similar manner to McLachlan and Jones (1988), McLachlan and Krishnan (2008), Park (2006), to find $\hat{\theta}$ that maximizes $\log L(\theta)$ in (2.1) within the EM framework, the vector of $x_i = (x_{i1}, x_{i2}, \ldots, x_{in_i})^T$, for $i = 1, \ldots, k$, should be introduced as missing (unobservable) data. In fact, for each interval $\mathcal{X}_i = [a_{i-1}, a_i)$, x_i consists of n_i independent unobservable data points falling into that interval. Hence, the complete-data vector can be written as $w = (y^T, x_1^T, \ldots, x_k^T)^T$. Furthermore, given y, each x_{il} has a density function $f(x_{il}|y) = f(x_{il};\theta)/P_i(\theta)$ for $l = 1, \ldots, n_i$ and $i = 1, \ldots, k$, where $P_i(\theta)$ is introduced above. Therefore, the complete-data likelihood can be written as

$$\begin{aligned} L_c(\theta) &\equiv f(w;\theta) \\ &= f(x|y;\theta)p(y;\theta) \\ &= \prod_{i=1}^{k}\prod_{l=1}^{n_i} \frac{f(x_{il};\theta)}{P_i(\theta)} \times \prod_{i=1}^{k}(P_i(\theta))^{n_i} \times C \\ &\propto \prod_i \prod_l f(x_{il};\theta), \end{aligned}$$

and its corresponding log-likelihood as

$$\log L_c(\theta) = \sum_{i=1}^{k}\sum_{l=1}^{n_i} \log f(x_{il};\theta) + C. \tag{2.3}$$

The Expectation-Maximization (EM) algorithm (Dempster et al., 1977) is a common iterative approach for computing the ML estimates in the case of incomplete or missing data problems. Implementing the EM algorithm needs starting values and alternatively iterates be-

tween two steps, the Expectation (E)- and Maximization (M)-steps, until convergence occurs. The following describes the E and M steps of the proposed EM approach.

E-Step:
The E-step calculates the expectation of the complete-data log-likelihood in (2.3) conditional on y and the current parameter estimates ($\theta^{(p)}$). Disregarding the constant term, the expectation of the $\log L_c(\theta)$ conditional on y and $\theta^{(p)}$ is given by:

$$Q(\theta, \theta^{(p)}) \equiv E_{\theta^{(p)}}\Big[\log L_c(\theta)|y\Big] = \sum_{i=1}^{k} n_i E_{\theta^{(p)}}\Big[\log f(X;\theta)|X \in \mathcal{X}_i\Big],$$

where the expectation is taken with respect to the density $f(x;\theta^{(p)})/P_i(\theta^{(p)})$.

Therefore, for the normally distributed grouped data, we can write:

$$Q(\theta, \theta^{(p)}) = -\frac{1}{2}n\{\log(2\pi) + \log\sigma^2\} - \frac{1}{2}\sigma^2 \sum_{i=1}^{k} n_i E_{\theta^{(p)}}\left\{(X-\mu)^2|X \in \mathcal{X}_i\right\}.$$

M-Step:

The M-step of the EM algorithm maximizes $Q(\theta, \theta^{(p)})$ with respect to θ at iteration $p+1$ to produce new estimates $\theta^{(p+1)} = (\mu^{(p+1)}, \sigma^{(p+1)})^T$. By using the idea of interchanging the differentiation and the expectation (the Leibniz integral rule), $Q(\theta, \theta^{(p)})$ can be differentiated with respect to $\theta = (\mu, \sigma)$ to obtain the following updated estimates:

$$\mu^{(p+1)} = \frac{\sum_{i=1}^{k} n_i E_{\theta^{(p)}}(X|X \in \mathcal{X}_i)}{n} \tag{2.4}$$

and

$$\sigma^{2(p+1)} = \frac{\sum_{i=1}^{k} n_i E_{\theta^{(p)}}\Big[(X-\mu^{(p+1)})^2|X \in \mathcal{X}_i\Big]}{n}, \tag{2.5}$$

where $n = \sum_{i=1}^{k} n_i$. The derivation of the expectations in (2.4) and (2.5) can be found in Appendix A.1.

Parameter Estimation via the MCEM Algorithm

Instead of calculating the exact form of the expectations in (2.4) and (2.5), one can apply the Monte Carlo EM (MCEM) algorithm (McLachlan and Krishnan, 2008, Wei and Tanner, 1990a,b), in which the required expectations are replaced with an average over simulations. The unobserved data x_i, for $i = 1, \ldots, k$, can be simulated (sampled) from the truncated univari-

Table 2.2: Bivariate grouped data representation.

$x_1 \backslash x_2$	$[b_0, b_1)$	$[b_1, b_2)$	$\cdots$	$[b_{s-1}, b_s)$	Total
$[a_0, a_1)$	n_{11}	n_{12}	$\cdots$	n_{1s}	$n_{1.}$
$[a_1, a_2)$	n_{21}	n_{22}	$\cdots$	n_{2s}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$[a_{r-1}, a_r)$	n_{r1}	n_{r2}	$\cdots$	n_{rs}	$n_{r.}$
Total	$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.s}$	n

ate normal distribution $f(x;\theta^{(p)})/P_i(\theta^{(p)})$ over each specific i-th interval. Now, considering M as the number of observations generated for each interval in the Monte Carlo simulation, the simulated sample for the i-th interval can be written as $(x_{i1},\ldots,x_{iM})$, and the MCEM updates are:

$$\mu^{(p+1)} = \frac{1}{n}\sum_{i=1}^{k} n_i \frac{1}{M}\sum_{h=1}^{M} x_{ih}$$

and

$$\sigma^{2(p+1)} = \frac{1}{n}\sum_{i=1}^{k} n_i \frac{1}{M}\sum_{h=1}^{M} (x_{ih} - \mu^{(p+1)})^2.$$

2.2.2 Bivariate and Multivariate Normal Grouped Data

Exact MLE for Bivariate Normal

The derivation of the exact MLE for bivariate normal grouped data is much like that for the univariate case, except that the multinomial probabilities depend on the bivariate normal CDF calculated over rectangles instead of intervals. The probability of a bivariate random variable $X = (X_1, X_2)$ belonging to a rectangle $\mathcal{X}_1 \times \mathcal{X}_2$ of the form $[a_{i-1}, a_i) \times [b_{j-1}, b_j)$; for $i = 1,\ldots,r$ and $j = 1,\ldots,s$, is

$$\begin{aligned} P_{ij}(\theta) &\equiv P(a_{i-1} \le X_1 < a_i, b_{j-1} \le X_2 < b_j) \\ &= \int_{a_{i-1}}^{a_i}\int_{b_{j-1}}^{b_j} f(x_1, x_2; \theta)\, dx_1 dx_2 \\ &= F_\theta(a_i, b_j) - F_\theta(a_{i-1}, b_j) - F_\theta(a_i, b_{j-1}) + F_\theta(a_{i-1}, b_{j-1}), \end{aligned}$$

where $f(.;\theta)$ and $F_\theta(.)$ are the bivariate normal density function and cumulative distribution function, respectively; with parameters $\theta = (\mu_{x_1}, \mu_{x_2}, \sigma_{x_1}, \sigma_{x_2}, \rho)$.

For each rectangle (or cell in Table 2.2), the frequencies n_{ij}, for $i = 1,\ldots,r$ and $j = 1,\ldots,s$,

are known, and therefore the following multinomial likelihood can be assumed for them:

$$L(\theta) = \frac{n!}{\prod_{i=1}^{r}\prod_{j=1}^{s} n_{ij}} \prod_{i=1}^{r}\prod_{j=1}^{s}\Big[\frac{P_{ij}(\theta)}{P(\theta)}\Big]^{n_{ij}},$$

where $n = \sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij}$ and $P(\theta) = \sum_{i=1}^{r}\sum_{j=1}^{s} P_{ij}(\theta) = 1$. Hence, the exact log-likelihood function is:

$$\begin{aligned}\log L(\theta) &= \sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij}\log P_{ij}(\theta) + C \\ &= \sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij}\log\Big[F_\theta(a_i,b_j) - F_\theta(a_{i-1},b_j) - F_\theta(a_i,b_{j-1}) + F_\theta(a_{i-1},b_{j-1})\Big] + C \end{aligned} \tag{2.6}$$

To find the MLEs of the parameters in θ, the log-likelihood function in (2.6) is maximized using numerical methods implemented by the *nlm()* function in R.

Parameter Estimation via the EM Algorithm

Extending the ideas of the univariate case, the goal is to maximize the exact log-likelihood for bivariate grouped data (see Equation (2.6)); using the EM approach. Therefore, the first step is to introduce x as missing observations in array form as:

$$x = \Big\{(x_{1ik}, x_{2jk}) \text{ for } i = 1,\ldots,r;\ j = 1,\ldots,s;\ k = 1,\ldots,n_{ij}\Big\}.$$

Then the complete-data $w = \{y, x\}$ can be defined over the rectangles, and their log-likelihood can be written as:

$$\begin{aligned}\log L_c(\theta) &= \log L(\theta) + \sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{n_{ij}}\log\frac{f(x_{1ik},x_{2jk};\theta)}{P_{ij}(\theta)} \\ &= \sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij}\log P_{ij}(\theta) + C + \sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{n_{ij}}\log\frac{f(x_{1ik},x_{2jk};\theta)}{P_{ij}(\theta)} \\ &= \sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{n_{ij}}\log f(x_{1ik},x_{2jk};\theta) + C \end{aligned} \tag{2.7}$$

The following presents the proposed E and M steps of the EM algorithm.

E-Step:

The E-step calculates the expected value of (2.7) given y and the current $\theta^{(p)}$, that is,

$$Q(\theta,\theta^{(p)}) \equiv \sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij} Q_{ij}(\theta,\theta^{(p)}), \tag{2.8}$$

where

$$Q_{ij}(\theta, \theta^{(p)}) = E_{\theta^{(p)}}\left\{ \log f\Big((X_1, X_2); \theta\Big) | (X_1, X_2) \in (\mathcal{X}_{i1} \times \mathcal{X}_{2j}) \right\},$$

with the expectation taken with respect to the density $f((x_1, x_2); \theta^{(p)})/P_{ij}(\theta^{(p)})$.

M-Step:

The M-step aims to find the parameter updates that maximize (2.8). Using a similar framework as in Section 2.1.2, the results are:

$$\mu_{x_1}^{(p+1)} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij} E_{\theta^{(p)}}\Big(X_1 | (X_1, X_2) \in (\mathcal{X}_{i1} \times \mathcal{X}_{2j}) \Big)}{n} \tag{2.9}$$

$$\mu_{x_2}^{(p+1)} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij} E_{\theta^{(p)}}\Big(X_2 | (X_1, X_2) \in (\mathcal{X}_{i1} \times \mathcal{X}_{2j}) \Big)}{n} \tag{2.10}$$

$$\sigma_{x_1}^{2(p+1)} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij} E_{\theta^{(p)}}\Big((X_1 - \mu_{x_1}^{(p+1)})^2 | (X_1, X_2) \in (\mathcal{X}_{i1} \times \mathcal{X}_{2j}) \Big)}{n} \tag{2.11}$$

$$\sigma_{x_2}^{2(p+1)} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij} E_{\theta^{(p)}}\Big((X_2 - \mu_{x_2}^{(p+1)})^2 | (X_1, X_2) \in (\mathcal{X}_{i1} \times \mathcal{X}_{2j}) \Big)}{n} \tag{2.12}$$

$$\rho^{(p+1)} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} n_{ij} E_{\theta^{(p)}}\Big((X_1 - \mu_{x_1}^{(p+1)})(X_2 - \mu_{x_2}^{(p+1)}) | (X_1, X_2) \in (\mathcal{X}_{i1} \times \mathcal{X}_{2j}) \Big)}{n} \tag{2.13}$$

The expectations in (2.9) to (2.13) are the moments of a truncated bivariate normal distribution ($f(x_1, x_2; \theta^{(p)})/P_{ij}(\theta^{(p)})$) and, therefore, can be calculated using the results of Manjunath and Wilhelm (2021) (for details, see Appendix A.2). To compute these expectations in R, we use the package *tmvtnorm* available at https://cran.r-project.org/.

MCEM for Bivariate Grouped Data

The MCEM algorithm can be used to replace the expectations in (2.9) to (2.13) by the average of simulated values. That means that M random samples of (X_1, X_2) are simulated (sampled) from the truncated bivariate normal distribution $f((x_1, x_2); \theta^{(p)})/P_{ij}(\theta^{(p)})$ over the rectangles,

and then their averages are used to replace the expectations in the EM parameter updates, obtaining the following MCEM-based parameter estimates:

$$\mu_{x_1}^{(p+1)} = \frac{\sum_i \sum_j n_{ij} \frac{1}{M} \sum_{h=1}^{M} x_{1ih}}{n},$$
$$\mu_{x_2}^{(p+1)} = \frac{\sum_i \sum_j n_{ij} \frac{1}{M} \sum_{h=1}^{M} x_{2jh}}{n},$$
$$\sigma_{x_1}^{2(p+1)} = \frac{\sum_i \sum_j n_{ij} \frac{1}{M} \sum_{h=1}^{M} (x_{1ih} - \mu_{x_1}^{(p+1)})^2}{n},$$
$$\sigma_{x_2}^{2(p+1)} = \frac{\sum_i \sum_j n_{ij} \frac{1}{M} \sum_{h=1}^{M} (x_{2jh} - \mu_{x_2}^{(p+1)})^2}{n}, \text{ and}$$
$$\rho^{(p+1)} = \frac{\sum_i \sum_j n_{ij} \frac{1}{M} \sum_{h=1}^{M} (x_{1ih} - \mu_{x_1}^{(p+1)})(x_{2jh} - \mu_{x_2}^{(p+1)})}{n}.$$

Extension of EM and MCEM to Multivariate Normal Grouped Data

By extending the ideas of univariate and bivariate normal grouped data, it is possible to find the parameter estimates (mean vector and covariance matrix) for multivariate normal grouped data using a matrix notation. Let $(x_1, \ldots, x_d)$ be an unobservable vector arising from a d-dimensional multivariate normal distribution with parameters of $(\underline{\mu}, \Sigma)$. Consider $r_1 r_2 \cdots r_d$ as the number of d-dimensional surfaces of the form $\mathcal{X}_{1i_1} \times \mathcal{X}_{2i_2} \times \cdots \times \mathcal{X}_{di_d} = [a_{1i_1-1}, a_{1i_1}] \times [a_{2i_2-1}, a_{2i_2}] \times \cdots \times [a_{di_d-1}, a_{di_d}]$ for $i_1 = 1, \ldots, r_1;\ i_2 = 1, \ldots, r_2;\ \ldots, i_d = 1, \ldots, r_d$. Let $n_{i_1,\ldots,i_d}$ be the observed number (count) of data points falling in each surface. These observed counts form a multinomial likelihood as follows:

$$L(\underline{\mu}, \Sigma) \equiv \frac{n!}{\prod_{i_1=1}^{r_1} \cdots \prod_{i_d=1}^{r_d} (n_{i_1,i_2,\ldots,i_d})!} \prod_{i_1=1}^{r_1} \cdots \prod_{i_d=1}^{r_d} \left(\frac{P_{i_1,i_2,\ldots,i_d}(\underline{\mu}, \Sigma)}{P(\underline{\mu}, \Sigma)} \right)^{n_{i_1,\ldots,i_d}}$$

where $n = \sum_{i_1} \cdots \sum_{i_d} n_{i_1,\ldots,i_d}$,

$$P_{i_1,i_2,\ldots,i_d}(\underline{\mu}, \Sigma) \equiv \int_{a_{1i_1-1}}^{a_{1i_1}} \cdots \int_{a_{di_d-1}}^{a_{di_d}} f(x_1, \ldots, x_d) dx_d \ldots dx_1,$$

and

$$P(\underline{\mu}, \Sigma) \equiv \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} P_{i_1,i_2,\ldots,i_d}(\underline{\mu}, \Sigma) = 1,$$

with $f(x_1, \ldots, x_d)$ being the probability density function of a multivariate normal distribution with parameters $(\underline{\mu}, \Sigma)$. Representing the observed data as $y = \left\{ n_{i_1,\ldots,i_d} \text{ for } i_1 = 1, \ldots, r_1; \ldots; i_d = \right.$

$1, \ldots, r_d\}$, the goal is to maximize

$$\log L(\underline{\mu}, \Sigma) = \sum_{i_1} \cdots \sum_{i_d} n_{i_1,\ldots,i_d} P_{i_1,\ldots,i_d}(\underline{\mu}, \Sigma) + C,$$

with respect to $(\underline{\mu}, \Sigma)$ using the EM framework.

Let $x = \{(x_{1i_1k}, x_{2i_2k}, \ldots, x_{di_dk})$ for $i_1 = 1, \ldots, r_1; \ldots; i_d = 1, \ldots, r_d; k = 1, 2, \ldots, n_{i_1,\ldots,i_d}\}$ be the missing vectors of observations. Thus, considering the complete data as $w = \{y, x\}$, the complete-data log-likelihood function can be written as:

$$\begin{aligned}
\log L_c(\underline{\mu}, \Sigma) &= \log L(\underline{\mu}, \Sigma) + \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} \sum_{k=1}^{n_{i_1,\ldots,i_d}} \log \frac{f\Big((x_{1i_1k}, x_{2i_2k}, \ldots, x_{di_dk}); (\underline{\mu}, \Sigma)\Big)}{P_{i_1,\ldots,i_d}(\underline{\mu}, \Sigma)} \\
&= \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} n_{i_1,\ldots,i_d} \log P_{i_1,\ldots,i_d}(\underline{\mu}, \Sigma) + C \\
&+ \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} \sum_{k=1}^{n_{i_1,\ldots,i_d}} \log \frac{f\Big((x_{1i_1k}, x_{2i_2k}, \ldots, x_{di_dk}); (\underline{\mu}, \Sigma)\Big)}{P_{i_1,\ldots,i_d}(\underline{\mu}, \Sigma)} \\
&= \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} \sum_{k=1}^{n_{i_1,\ldots,i_d}} \log f\Big((x_{1i_1k}, x_{2i_2k}, \ldots, x_{di_dk}); (\underline{\mu}, \Sigma)\Big) + C
\end{aligned}$$

The E-step and M-step of the EM algorithm are described as follows.

E-Step:

The E-step calculates:

$$Q\Big((\underline{\mu}, \Sigma); (\underline{\mu}, \Sigma)^{(p)}\Big) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} n_{i_1,\ldots,i_d} Q_{i_1,\ldots,i_d}\Big((\underline{\mu}, \Sigma); (\underline{\mu}, \Sigma)^{(p)}\Big),$$

where

$$\begin{aligned}
&Q_{i_1,\ldots,i_d}\Big((\underline{\mu}, \Sigma); (\underline{\mu}, \Sigma)^{(p)}\Big) = \\
&E_{(\underline{\mu},\Sigma)^{(p)}}\left\{ \log f\Big((X_1, \ldots, X_d); (\underline{\mu}, \Sigma)\Big) \Big| (X_1, \ldots, X_d) \in (\mathcal{X}_{1i_1} \times \cdots \times \mathcal{X}_{di_d}) \right\}.
\end{aligned}$$

Hence,

$$Q\Big((\underline{\mu},\Sigma);(\underline{\mu},\Sigma)^{(p)}\Big) = \sum_{i_1}\cdots\sum_{i_d} n_{i_1,\ldots,i_d}$$
$$E_{(\underline{\mu},\Sigma)^{(p)}}\left\{\Big[-\frac{d}{2}\log(2\pi)+\frac{1}{2}\log(|\Sigma|^{-1})-\frac{1}{2}(X-\underline{\mu})^T\Sigma^{-1}(X-\underline{\mu})\Big]\Big|(X_1,\ldots,X_d)\in(\mathcal{X}_{1i_1}\times\cdots\times\mathcal{X}_{di_d})\right\}$$
$$=\sum_{i_1}\cdots\sum_{i_d} n_{i_1,\ldots,i_d}\left\{-\frac{d}{2}\log(2\pi)+\frac{1}{2}\log(|\Sigma|^{-1})\right.$$
$$\left.-\frac{1}{2}tr\left[\Sigma^{-1}E_{(\underline{\mu},\Sigma)^{(p)}}\Big((X-\underline{\mu})(X-\underline{\mu})^T\Big|(X_1,\ldots,X_d)\in(\mathcal{X}_{1i_1}\times\cdots\times\mathcal{X}_{di_d})\Big)\right]\right\}.$$

M-Step:

The M-step maximizes $Q\Big((\underline{\mu},\Sigma);(\underline{\mu},\Sigma)^{(p)}\Big)$ w.r.t $(\underline{\mu},\Sigma)$, obtaining:

$$\underline{\mu}^{(p+1)} = \frac{1}{n}\sum_{i_1=1}^{r_1}\cdots\sum_{i_d=1}^{r_d} n_{i_1,\ldots,i_d}E_{(\underline{\mu},\Sigma)^{(p)}}\left\{X\Big|(X_1,\ldots,X_d)\in(\mathcal{X}_{1i_1}\times\cdots\times\mathcal{X}_{di_d})\right\} \tag{2.14}$$

$$\Sigma^{(p+1)} = \frac{1}{n}\sum_{i_1=1}^{r_1}\cdots\sum_{i_d=1}^{r_d} n_{i_1,\ldots,i_d}\times$$
$$E_{(\underline{\mu},\Sigma)^{(p)}}\left\{\Big((X-\underline{\mu}^{(p+1)})(X-\underline{\mu}^{(p+1)})^T\Big)\Big|(X_1,\ldots,X_d)\in(\mathcal{X}_{1i_1}\times\cdots\times\mathcal{X}_{di_d})\right\}. \tag{2.15}$$

The expectations in (2.14) and (2.15) are the moments of a truncated multivariate normal $\frac{f(x_1,\ldots,x_d;(\underline{\mu},\Sigma))}{P_{i_1,\ldots,i_d}(\underline{\mu},\Sigma)}$ and as in the bivariate case, the results in Manjunath and Wilhelm (2021) can be used to calculate these moments, as shown in Appendix A.2. In our R code, these expectations are computed using the package *tmvtnorm*.

An alternative approach to calculating these expectations is to use the MCEM algorithm. The MCEM approach first simulates (samples) M multivariate random samples of $X=(X_1,\ldots,X_d)$ from the truncated multivariate normal distribution $\frac{f(x_1,\ldots,x_d;(\underline{\mu},\Sigma))}{P_{i_1,\ldots,i_d}(\underline{\mu},\Sigma)}$ over all surfaces and then replaces the expectations in (2.14) and (2.15) with the averages of the simulated sample vectors obtaining the following parameter updates:

$$\underline{\mu}^{(p+1)} = \frac{1}{n}\left[\sum_{i_1}\cdots\sum_{i_d} n_{i_1,\ldots,i_d}\frac{1}{M}\sum_{h=1}^{M}x_{ih}\right]$$

and

$$\Sigma^{(p+1)} = \frac{1}{n}\left[\sum_{i_1}\cdots\sum_{i_d} n_{i_1,\ldots,i_d}\frac{1}{M}\sum_{h=1}^{M}(x_{ih}-\mu^{(p+1)})(x_{ih}-\mu^{(p+1)})^T\right].$$

2.2.3 Standard Errors for the EM and MCEM mean Estimates

Following the ideas in Chapter 4 of McLachlan and Krishnan (2008), standard errors for the EM estimates for grouped data can be obtained using an approximation of the observed information matrix, which is called the empirical observed information matrix, $I_{e,g}$. For the univariate grouped data, $I_{e,g}$ can be calculated as:

$$I_{e,g}(\hat{\theta}; y) = \sum_{i=1}^{r} n_i s_i(\hat{\theta}) s_i^T(\hat{\theta}) - n\bar{s}(\hat{\theta})\bar{s}^T(\hat{\theta}), \tag{2.16}$$

where $\bar{s}(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{r} n_i s_i(\hat{\theta})$, $s_i(\hat{\theta}) = \frac{\partial Q_i(\theta,\hat{\theta})}{\partial\theta}\big|_{\theta=\hat{\theta}}$, and $\hat{\theta}$ contains the EM estimates.

Similarly, the empirical observed information matrix for multivariate grouped data is as follows:

$$I_{e,g}(\hat{\underline{\mu}}, \hat{\Sigma}; y) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} n_{i_1,\ldots,i_d} s_{i_1,\ldots,i_d}(\hat{\underline{\mu}}, \hat{\Sigma}) s_{i_1,\ldots,i_d}^T(\hat{\underline{\mu}}, \hat{\Sigma}) - n\bar{s}(\hat{\underline{\mu}}, \hat{\Sigma})\bar{s}^T(\hat{\underline{\mu}}, \hat{\Sigma}) \tag{2.17}$$

where $s_{i_1,\ldots,i_d}(\hat{\underline{\mu}}, \hat{\Sigma}) = \dfrac{\partial Q_{i_1,\ldots,i_d}\Big((\underline{\mu}, \Sigma); (\hat{\underline{\mu}}, \hat{\Sigma})\Big)}{\partial(\underline{\mu}, \Sigma)}\Big|_{(\underline{\mu},\Sigma)=(\hat{\underline{\mu}},\hat{\Sigma})}$.

The inverse of $I_{e,g}$ demonstrates an approximation of the covariance matrix of the EM estimates, with the diagonal containing the standard errors.

For our study, we calculate the standard error for the EM estimates of μ and $\underline{\mu}$ using equations (2.16) and (2.17), respectively, and fixing the variance-covariance parameter values to the ones obtained by the EM algorithm. Using the notation from the previous sections, we can show that the score function for μ for univariate grouped data is:

$$s_i(\hat{\mu}, \hat{\sigma}^2) = \frac{1}{\hat{\sigma}^2} E\Big\{(X - \hat{\mu}) | X \in \mathcal{X}_i\Big\},$$

and for the multivariate case, it is:

$$s_{i_1,\ldots,i_d}(\hat{\underline{\mu}}, \hat{\Sigma}^2) = E\Big\{(X - \hat{\underline{\mu}}) | X \in (\mathcal{X}_{1i_1} \times \cdots \times \mathcal{X}_{di_d})\Big\}^T \hat{\Sigma}^{-1}.$$

Using these score functions, we also obtain standard errors for the mean MCEM estimates using the Louis' approach (Louis, 1982) as described in Chapter 6 of McLachlan and Krishnan (2008) with all expectations replaced by the averages of observations simulated using the final MCEM estimates.

We use the standard errors (se) proposed above to construct 95% confidence intervals of the form: $\hat{\mu} \pm 1.96 se(\hat{\mu})$.

2.3 Results

2.3.1 Galton Data

The Galton dataset was first introduced by Francis Galton in 1886 (Galton, 1889, Hanley, 2004) and consists of a two-way frequency table containing the number of parents and children falling into different possible height intervals. The individual height observations are not available; only the frequencies (grouped data) are available. Moreover, for each interval, the midpoints (as the averages of the lower and upper limits of the intervals) are also available. This data set is a well-known example of normally distributed grouped data. The Galton data are electronically and publicly available in the R package *HistData*. In this study, each of the variables (parent's height and child's height) was first analyzed separately as univariate normal grouped data before considering the bivariate case. The results are provided in the following.

Univariate Case

First, the exact MLE of the parameters with the assumption of normal distribution of both parent height and child height data were obtained using the approach described in Section 2.2.1. As described, for the grouped data, the exact likelihood estimation was conducted numerically using the R functions *optim()* and *nlm* (L-BFGS-B method); the results are shown in Table 2.3 under Exact MLE. Note that the numerical maximization of the exact likelihood is highly sensitive to initial values. The parameter estimates using the EM algorithm to maximize the exact likelihood were then found, along with those using the MCEM algorithm. The results for both EM and MCEM algorithms are also presented in Table 2.3. As can be expected by the convergence properties of the EM algorithm (McLachlan and Krishnan, 2008), its estimates were close to those obtained by direct maximization of the exact likelihood (mean absolute relative difference (MARD) across parameters = 0.005672%). The MCEM estimates were also close to the Exact MLE results (MARD = 0.020222%), but not as close as the EM results, which was also expected from the properties of the MCEM (McLachlan and Krishnan, 2008).

Table 2.3: Estimates of the mean and variance (Var) of parent and child height variables (considering the univariate case) from the Galton data using the three proposed methods. The standard error (se) for each mean estimate is also provided. For EM and MCEM, standard errors are obtained using the methods in Section 2.2.3. For exact MLE, standard errors are found using the observed information matrix and the delta method.

Method	Mean parent (se)	Mean child (se)	Var parent	Var child
Exact MLE	68.30030 (0.05967)	68.09834 (0.084364)	3.24432	6.50945
EM	68.30026 (0.03818)	68.09834 (0.05232)	3.24482	6.50971
MCEM	68.30070 (0.05992)	68.09600 (0.08435)	3.24312	6.50763

Bivariate Case

In this case, the Galton data were considered as bivariate grouped data and the methods proposed in Section 2.2.2 were used to find the parameter estimates. The results for all five parameters (including mean of parents, mean of children, variance of parents, variance of children, and correlation of heights between parents and children) are shown in Table 2.4. Note that as mentioned in Section 2.2.2, for parameter estimates using the exact MLE method for bivariate data, the *nlm()* and *optim()* functions in R were used. The EM estimates were closest to those from the exact MLE method, with mean absolute relative difference over the five parameters of 0.0012%.

Table 2.4: Estimates of mean, variance (Var) and correlation (Corr) parameters for bivariate Galton data using the three proposed methods. Standard errors (se) for the mean estimates are also provided. For EM and MCEM, standard errors are obtained using the methods in Section 2.2.3. For exact MLE, standard errors are found using the observed information matrix.

Method	Mean parent	Mean child	Var parent	Var child	Corr
Exact MLE	68.300475(0.059918)	68.098651(0.084394)	3.243895	6.513746	0.470162
EM	68.300495(0.059656)	68.098736(0.084259)	3.243960	6.513621	0.470171
MCEM	68.302157(0.058073)	68.098961(0.070917)	3.248326	6.514850	0.469763

2.3.2 Simulation Studies

In this section, the parameter estimation methods for normally distributed grouped data are applied to simulated data for both the univariate and bivariate cases.

Univariate Simulation

In this study, we conducted simulations on 15 different scenarios obtained by varying the sample size n (50, 100, 300, 600 and 1000) and the number of equal-sized intervals (or bins, $k = 8, 15$, and 30). For each scenario, 500 univariate datasets (in total 7500 datasets) are simulated. All simulated data are from a univariate normal distribution with parameters $\mu = 68$ and $\sigma = 2.5$ ($\sigma^2 = 6.25$). Moreover, according to Booth and Hobert (1999) and McCulloch (1997), as the number of MCEM iterations for the univariate data was between 10 to 30, we fix the number of Monte Carlo simulations for MCEM estimates to $M = 1000$. We use the mean absolute difference between current and updated estimates as a stop criterion for both EM and MCEM algorithms.

The parameters (μ and σ) are estimated using the three methods described in Section 2.2.1: Exact MLE, EM algorithm and MCEM algorithm. For all the methods, we set the initial

values of the parameters as $\mu = 67, \sigma = 2$. The root mean squared error (RMSE) of μ and σ over 500 samples are presented in Tables 2.5 and 2.7. Box plots of the parameter estimates obtained across all different scenarios are shown in Figures 2.1 and 2.2. We can observe that for all parameters and all bin sizes the RMSE of the estimates of Exact MLE, EM, and MCEM decrease as the sample size n increases.

To evaluate the performance of our proposed standard errors for the mean estimates, we calculate the empirical coverage (EC) of 95% confidence intervals of the form $\hat{\mu} \pm 1.96se(\hat{\mu})$. We observe in Table 2.6 that most of the ECs are close to the established confidence level of 95%. In addition, we can observe that the standard deviations of the mean estimates are close to the mean of the proposed standard errors as expected.

Bivariate Simulation

For bivariate data we simulated 500 datasets for each sample size n of 50, 100, 300, 600, and 1000 with 10 equal intervals for each variable (X_1 and X_2) resulting in 100 rectangles and 2500 datasets. Datasets are simulated from a bivariate normal with parameters $\mu_{x_1} = 68$, $\mu_{x_2} = 68$, $\sigma^2_{x_1} = 3$, $\sigma^2_{x_2} = 6$, $\text{cov}(X_1, X_2) = 2$. The initial values selected for exact MLE, EM and MCEM methods are $\mu_{x_1} = 67$, $\mu_{x_2} = 67$, $\sigma^2_{x_1} = 3.2$, $\sigma^2_{x_2} = 6.2$, $\text{cov}(X_1, X_2) = 2.227106$. According to Booth and Hobert (1999) and McCulloch (1997), the number of Monte Carlo simulations to obtain the MCEM estimates was fixed to $M = 5000$ as the number of MCEM iterations for bivariate data was more than 40. As in the univariate case, we use the mean absolute difference between current and updated estimates as a stop criterion for both EM and MCEM algorithms.

Figures 2.3 to 2.7 present the box plots of the parameter estimates for each method and different sample sizes. Our results also show that the Exact MLE, EM and MCEM yielded very similar estimates as expected even for the smaller n of 50. In addition, we can observe in Table 2.8 that the root mean squared error (RMSE) of the estimates decreases as the sample size n increases for all parameters and methods. Table 2.9 shows the ECs for 95% confidence intervals of the form $\hat{\mu} \pm 1.96se(\hat{\mu})$ for both μ_{x_1} and μ_{x_2}. We observe that in most cases the ECs are close to the established 95% level of confidence for both EM and MCEM methods.

2.4 Discussion and Conclusion

We have proposed three approaches, namely, Exact MLE, EM and MCEM algorithms, to estimate the parameters of normally distributed grouped data. The univariate, bivariate and multivariate normal cases were considered, and parameter estimates using each method were presented. For the exact MLE approach, by considering the counts' distribution to be multi-

nomial, with probabilities based on the normal CDFs, the exact data log-likelihood could be formulated, and the MLE values could be found using numerical methods. For EM and MCEM algorithms, using the exact observed-data log-likelihood, the complete-data log-likelihood was computed, and the parameter estimates were obtained in closed forms using the formulas presented in Sections 2.2.1 and 2.2.2.

To compare the methods, first, we considered the well-known Galton data, and parameter estimates were found for the cases of univariate and bivariate grouped data. Next, the mean absolute relative differences between the estimates obtained by Exact MLE and the other methods (EM and MCEM) were calculated. They showed that EM led to the closest results to the exact MLE. Then, simulation studies were implemented for the univariate and bivariate cases for different scenarios. For most parameters, the results from the EM and MCEM algorithms were similar to the ones from the exact MLE, as expected by their convergence properties shown in Chapters 1 and 3 of McLachlan and Krishnan (2008).

Based on our results, we conclude that there are some advantages and drawbacks regarding the three methods. The exact MLE method leads to efficient and unbiased estimates; however, there is no closed-form for the parameter estimates, and they are found using numerical optimization methods. Moreover, this method is susceptible to the optimization method and initial values. In comparison, in our analyses, the EM and MCEM methods were not as sensitive to initial values as the Exact MLE method. In addition, for both EM and MCEM algorithms, there are specific and closed formulae for the parameter estimates. We have not extensively studied the behaviour of the methods when changing the ratio of n over k by fixing n and varying k; however, based on the available simulation results, we noticed that when k (number of bins/intervals) becomes larger, while n (sample size) is small, the exact MLE method does not perform as well as the EM and MCEM approaches. That is because we might have some empty intervals or intervals with a very small number of observations in those situations, and that affects the performance of the exact MLE; however, as the EM and MCEM use the expectations (or simulated averages) over the intervals, we can see better behaviour of these approaches. This could be further investigated in future work on this topic.

Table 2.5: *Simulation results: univariate case.* RMSE of mean estimates of 500 simulated samples for $n = 50, 100, 300, 600$, and 1000 and number of intervals (bins) $k = 8, 15$, and 30 over three estimation methods.

		RMSE for Means		
Method	n	$k = 8$	$k = 15$	$k = 30$
Exact MLE	50	0.34368	0.34848	0.80577
	100	0.25917	0.25484	0.73227
	300	0.13859	0.15677	0.62772
	600	0.10698	0.10678	0.33962
	1000	0.07972	0.08207	0.22194
EM	50	0.34369	0.34849	0.37453
	100	0.25917	0.25485	0.25459
	300	0.13859	0.15678	0.14536
	600	0.10697	0.10678	0.10202
	1000	0.07972	0.08207	0.07917
MCEM	50	0.34369	0.34845	0.37479
	100	0.25922	0.25501	0.25457
	300	0.13893	0.15668	0.14526
	600	0.10705	0.10687	0.1022
	1000	0.07976	0.08243	0.07925

Table 2.6: *Simulation results: univariate case.* Average standard error (SE) and empirical coverage (EC) (over 500 simulated datasets) for the EM and MCEM estimates of μ for $n = 50, 100, 300, 600, 1000$, and $k = 15$ number of intervals (bins).

		Standard Errors for Mean Estimates		
n	Method	Ave. $\hat{\mu}$ (std $\hat{\mu}$)	Ave. SE for $\hat{\mu}$	EC
50	EM	68.00287872 (0.34882416)	0.36509683	94.8
	MCEM	68.00295812 (0.34879046)	0.35490471	94.6
100	EM	67.99932895 (0.25510392)	0.25597085	94.4
	MCEM	67.99935779 (0.25526480)	0.25237501	93.8
300	EM	68.00202742 (0.15692531)	0.14679516	92.8
	MCEM	68.00153299 (0.15683075)	0.14610848	92.4
600	EM	67.99406868 (0.10672629)	0.10381674	94.2
	MCEM	67.99370576 (0.10679113)	0.10354026	94.0
1000	EM	67.99913644 (0.08215109)	0.080350265	93.8
	MCEM	67.99914431 (0.08250915)	0.080233290	93.8

Table 2.7: *Simulation results: univariate case.* RMSE of variance estimates of 500 simulated samples for $n = 50, 100, 300, 600$, and 1000 and number of intervals (bins) $k = 8, 15$, and 30 over three estimation methods.

		RMSE for Variances		
Method	n	$k = 8$	$k = 15$	$k = 30$
Exact MLE	50	1.40998	1.26072	1.90078
	100	0.93567	0.89101	1.2376
	300	0.54813	0.50938	1.00808
	600	0.39345	0.35808	0.49013
	1000	0.30994	0.29212	0.45055
EM	50	1.40998	1.2607	1.28548
	100	0.93576	0.89092	0.86004
	300	0.54812	0.50927	0.5132
	600	0.39336	0.35814	0.35481
	1000	0.31006	0.29215	0.29758
MCEM	50	1.40989	1.26103	1.2848
	100	0.93697	0.89189	0.86105
	300	0.55068	0.50973	0.51382
	600	0.39327	0.35742	0.3557
	1000	0.31167	0.29247	0.29938

Table 2.8: Root mean squared errors (RMSE) of bivariate parameters (μ_{x_1}, μ_{x_2}, $\sigma^2_{x_1}$, $\sigma^2_{x_2}$, ρ) across 500 data sets for each sample size $n = 50, 100, 300, 600, 1000$ with 10 intervals for each variable (100 rectangles) and three methods used.

Parameter	Sample size	Exact MLE	EM	MCEM
μ_{x_1}	50	0.252397	0.252381	0.252402
	100	0.17686	0.176857	0.176825
	300	0.099135	0.099115	0.099109
	600	0.06756	0.067556	0.067578
	1000	0.054078	0.054073	0.054092
μ_{x_2}	50	0.337363	0.337353	0.337348
	100	0.250034	0.250038	0.250025
	300	0.140965	0.140715	0.140641
	600	0.101405	0.101408	0.101358
	1000	0.075395	0.075394	0.075369
$\sigma^2_{x_1}$	50	0.636113	0.635723	0.63575
	100	0.438604	0.438275	0.438731
	300	0.244176	0.244165	0.244096
	600	0.187619	0.187514	0.187396
	1000	0.138219	0.13813	0.138106
$\sigma^2_{x_2}$	50	1.306861	1.305404	1.30616
	100	0.953621	0.952149	0.953117
	300	0.519407	0.519655	0.520149
	600	0.378151	0.376675	0.376777
	1000	0.287702	0.286377	0.287136
ρ	50	0.115697	0.115659	0.115676
	100	0.081728	0.081703	0.081697
	300	0.044546	0.044568	0.044561
	600	0.033404	0.033389	0.033383
	1000	0.026724	0.026709	0.026731

Table 2.9: *Simulation results: bivariate case.* Average standard error (SE) and empirical coverage (EC) (over 500 simulated datasets) for the EM and MCEM estimates of μ_{x_1} and μ_{x_2} for n = 50, 100, 300, 600, 1000, and 100 rectangles.

Parameter	n	Method	Ave. $\hat{\mu}$ (sd $\hat{\mu}$)	Ave. SE for $\hat{\mu}$	EC
μ_{x_1}	50	EM	67.99029176 (0.25244727)	0.24601741	0.942
		MCEM	67.99024615 (0.25246634)	0.23884756	0.934
	100	EM	68.01838946 (0.17607445)	0.17393201	0.936
		MCEM	68.01840550 (0.17604050)	0.16872550	0.934
	300	EM	67.99715989 (0.09917395)	0.10089726	0.966
		MCEM	67.99715451 (0.09916721)	0.09782386	0.958
	600	EM	67.99817344 (0.06759916)	0.07171556	0.966
		MCEM	67.99813458 (0.06762007)	0.06942490	0.960
	1000	EM	67.99940268 (0.05412395)	0.05552854	0.946
		MCEM	67.99927880 (0.05414088)	0.05376547	0.942
μ_{x_2}	50	EM	67.98844687 (0.33749244)	0.34712279	0.934
		MCEM	67.98851100 (0.33749043)	0.28934334	0.892
	100	EM	68.01415707 (0.24988641)	0.24504268	0.950
		MCEM	68.01411880 (0.24987648)	0.20503459	0.898
	300	EM	67.99908176 (0.14085275)	0.14285627	0.958
		MCEM	67.99912894 (0.14077926)	0.11950836	0.900
	600	EM	67.99156129 (0.10115777)	0.10078140	0.948
		MCEM	67.99151371 (0.10110339)	0.08452820	0.892
	1000	EM	67.99395405 (0.07522677)	0.07822683	0.954
		MCEM	67.99405000 (0.07520861)	0.06556686	0.918

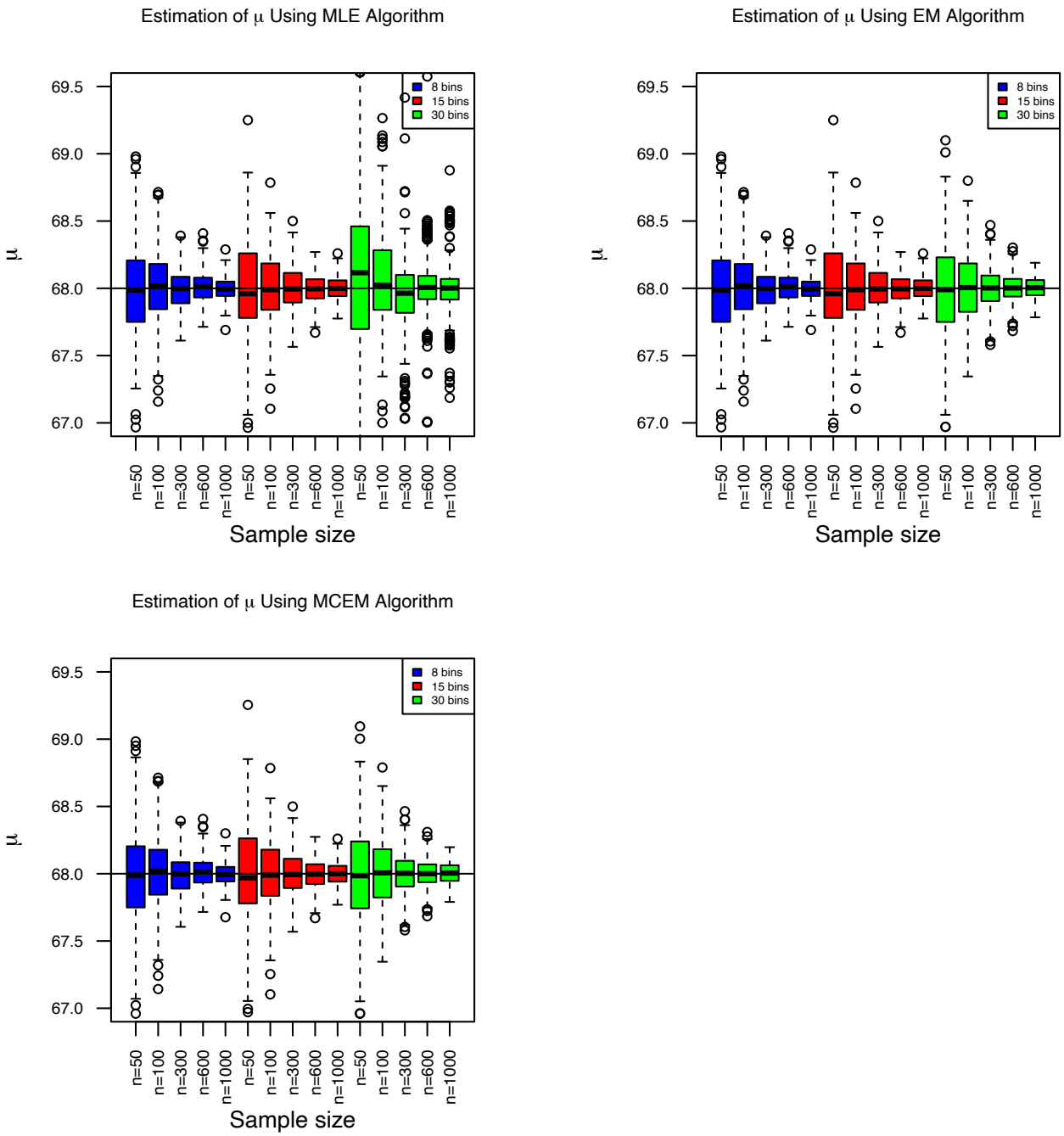

Figure 2.1: *Simulation results: univariate case.* Mean estimates for k = 8, 15 and 30 intervals (bins) for sample sizes $n = 50, 100, 300, 600, 1000$. True mean value $\mu = 68$.

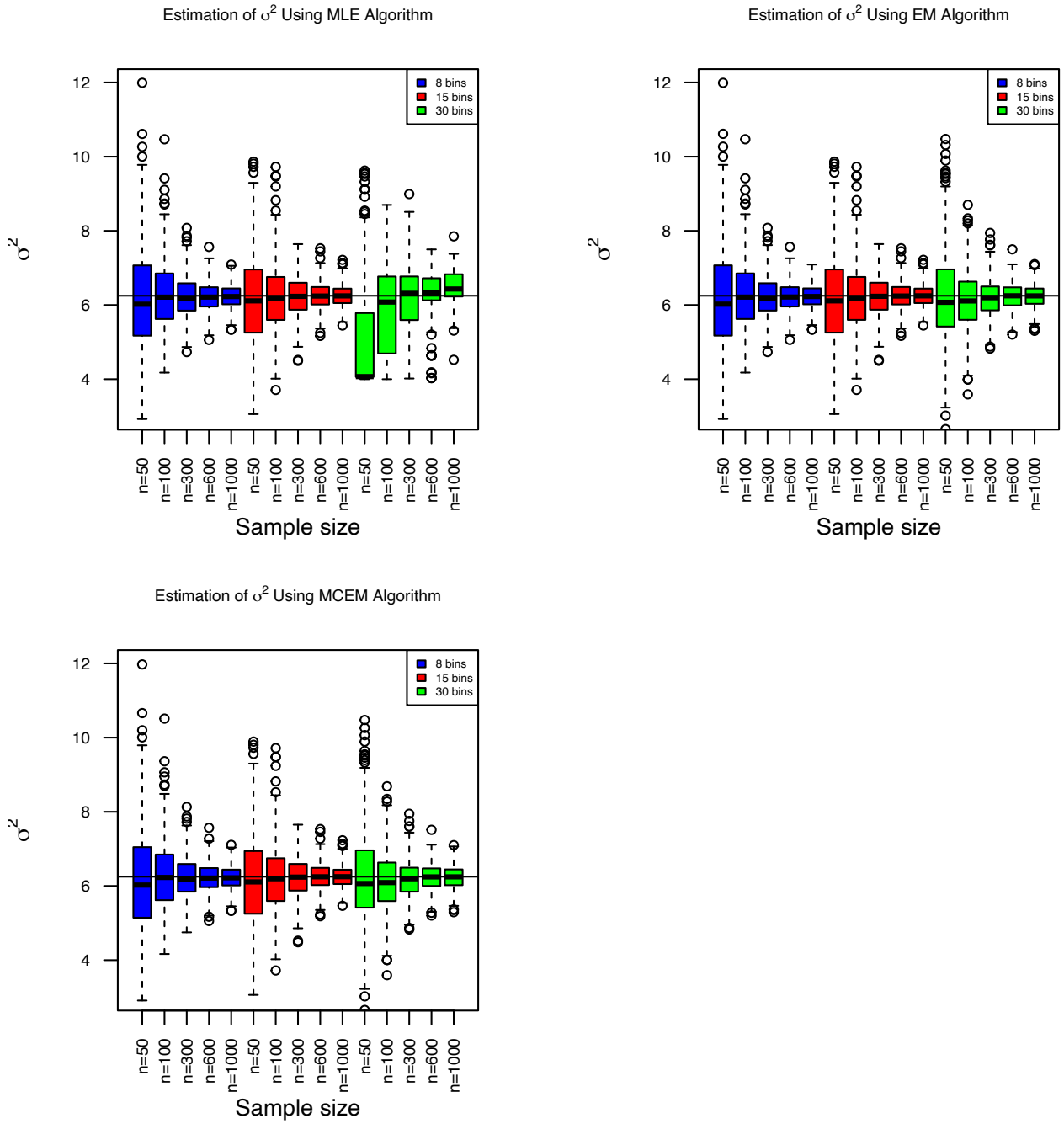

Figure 2.2: *Simulation results: univariate case.* Variance estimates for $k = 8$, 15 and 30 intervals (bins) for sample sizes $n = 50, 100, 300, 600, 1000$. True variance value $\sigma^2 = 6.25$.

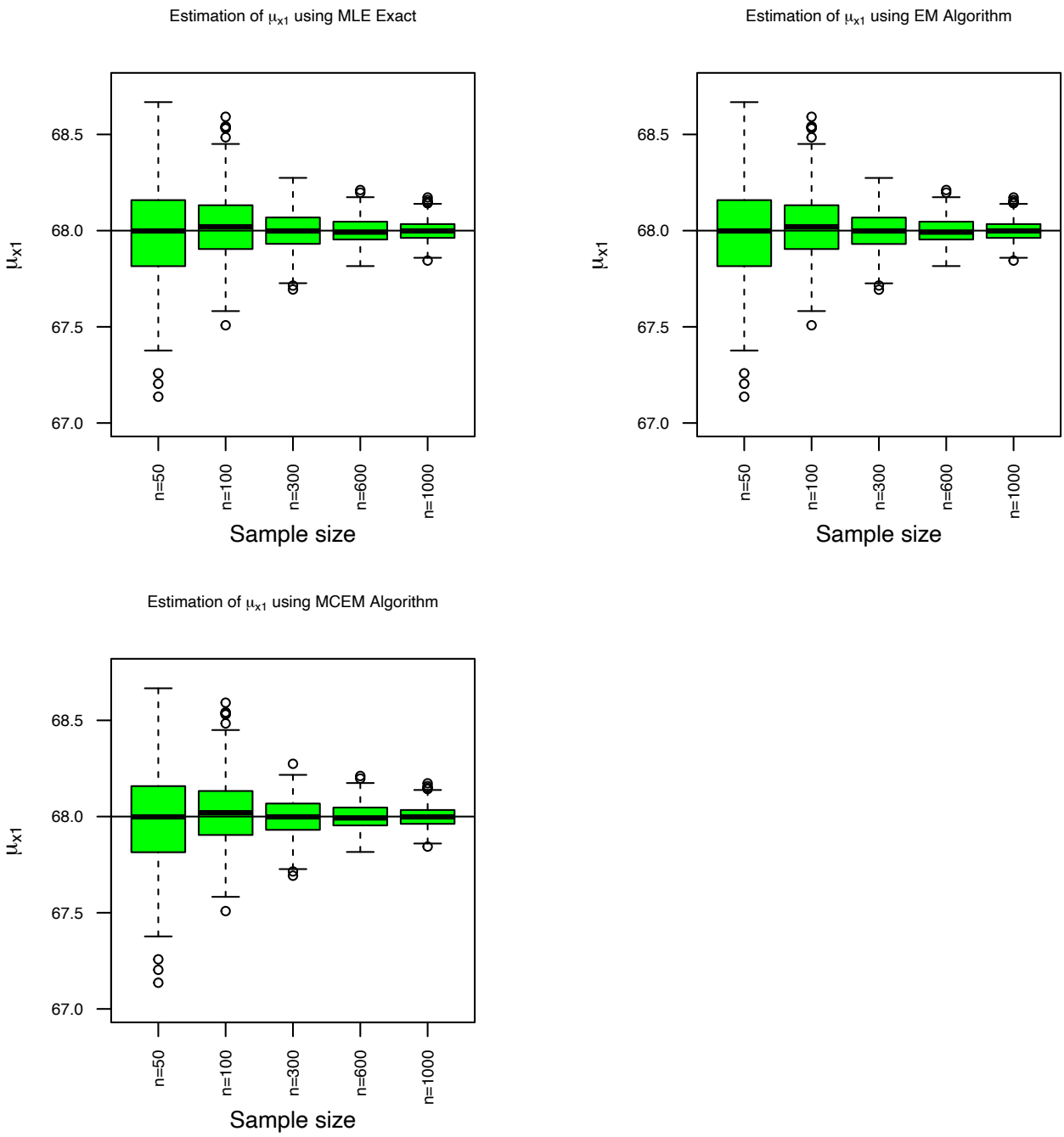

Figure 2.3: *Simulation results: bivariate case.* Estimates of μ_{x_1} for sample sizes of n = 50, 100, 300, 600, 1000, and k = 10 intervals for each variable. The horizontal solid line corresponds to the true value μ_{x_1} = 68.

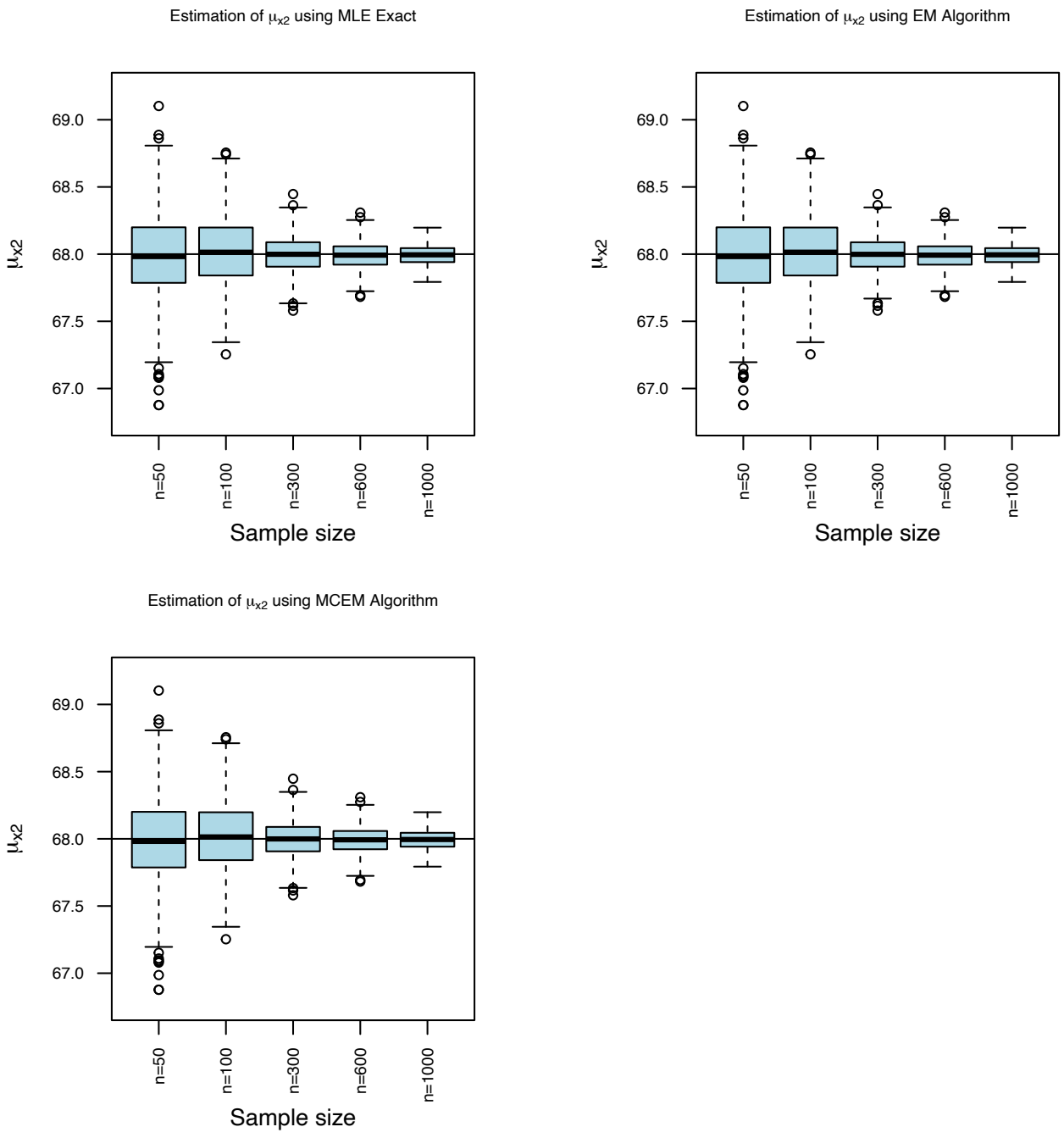

Figure 2.4: *Simulation results: bivariate case.* Estimates of μ_{x_2} for sample sizes of n = 50, 100, 300, 600, 1000, and k = 10 intervals for each variable. The horizontal solid line corresponds to the true value μ_{x_2} = 68.

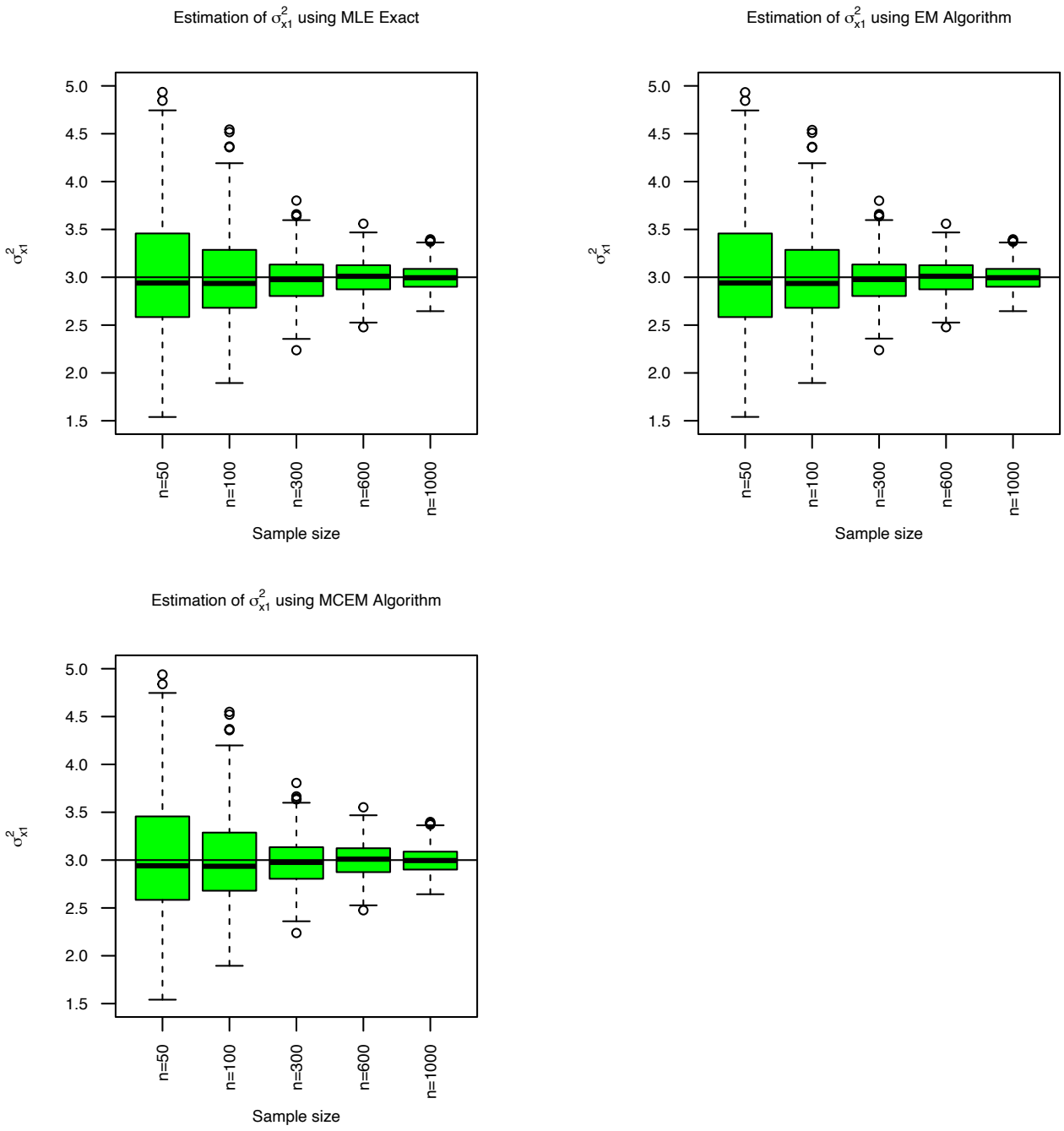

Figure 2.5: *Simulation results: bivariate case.* Estimates of $\sigma^2_{x_1}$ for sample sizes of n = 50, 100, 300, 600, 1000, and k = 10 intervals for each variable. The horizontal solid line corresponds to the true value $\sigma^2_{x_1}$ = 3.

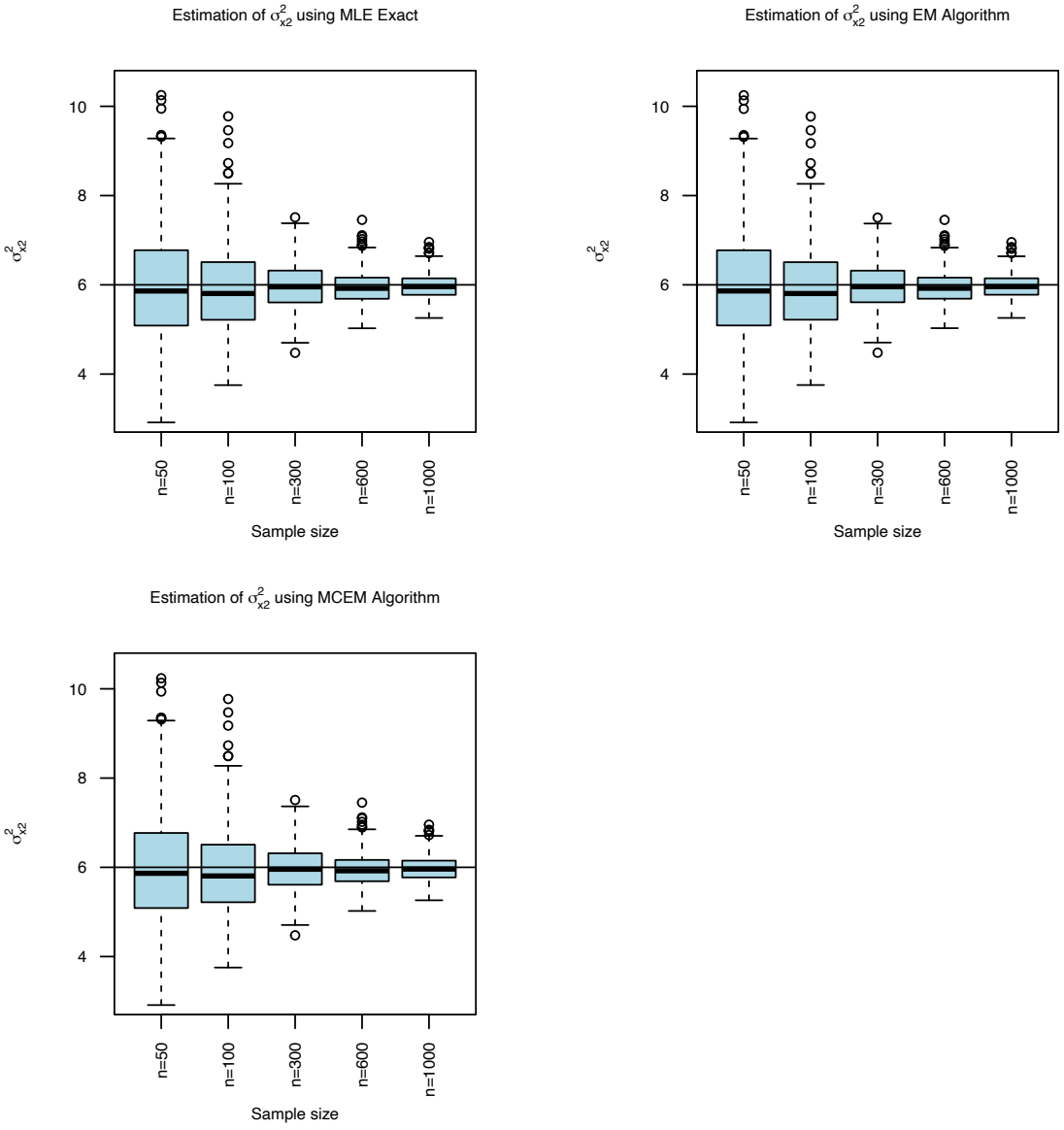

Figure 2.6: *Simulation results: bivariate case.* Estimates of $\sigma^2_{x_2}$ for sample sizes of n = 50, 100, 300, 600, 1000, and k = 10 intervals for each variable. The horizontal solid line corresponds to the true value $\sigma^2_{x_2} = 6$.

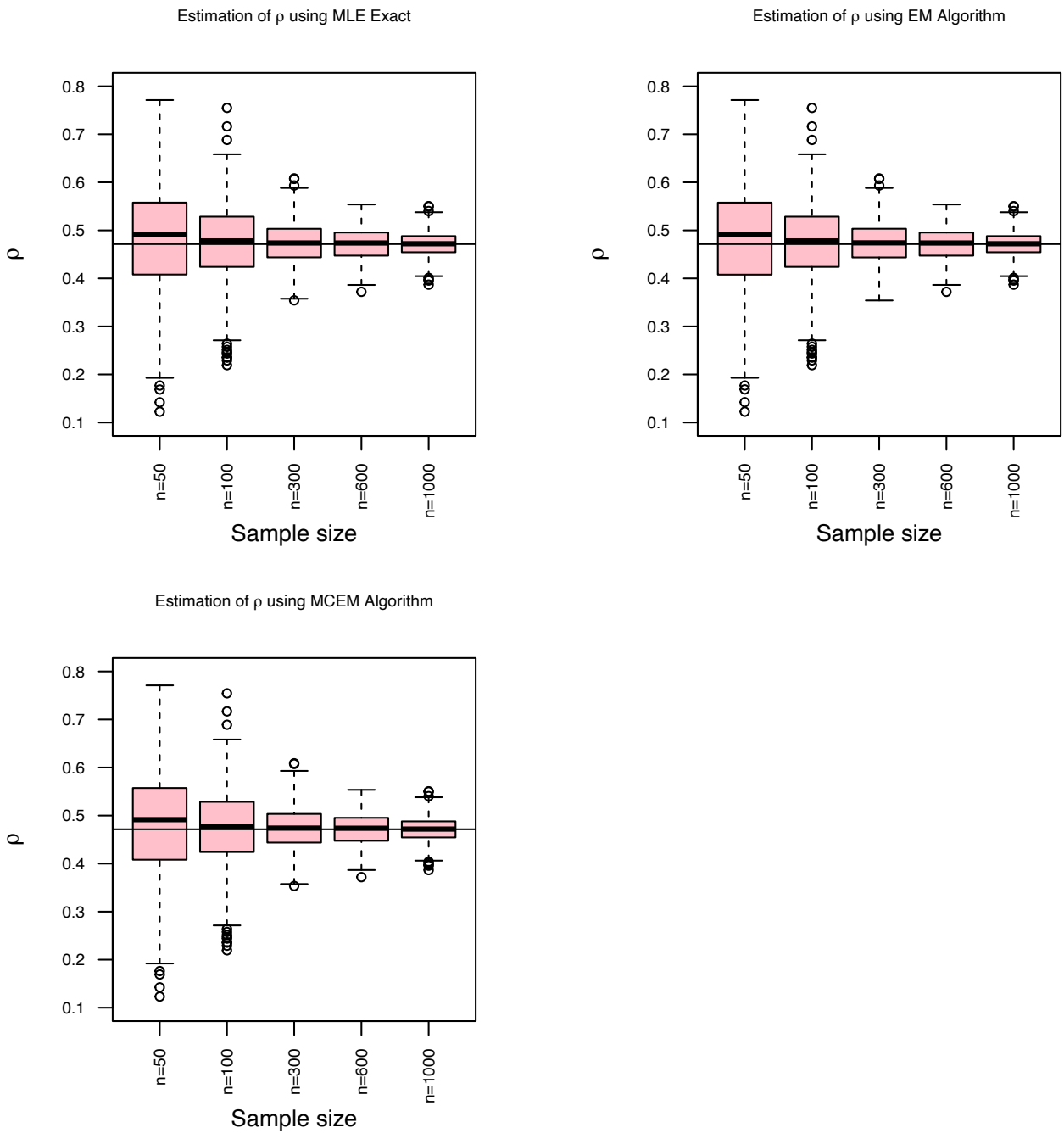

Figure 2.7: *Simulation results: bivariate case.* Estimates of ρ for sample sizes of n = 50, 100, 300, 600, 1000, and k = 10 intervals for each variable. The horizontal solid line corresponds to the true value of ρ.

Chapter 3

Model-based Clustering of Single-Cell RNA Sequencing Data

3.1 Introduction to single-cell sequencing

Cells are the essential units in biology. For many years, biologists have been interested in discovering more about the distinct cell types in multi-cellular organisms. Cells can be distinguished by their phenotype such as size and shape or at the molecular level, based on their genome, epigenome, and transcriptome. In this thesis, we focus on the transcriptome, which includes all ribonucleic acid (RNA) transcripts present in a given cell population indicating the genes that are being expressed at any given time. So far, most of the technologies and analyses have studied the expression of RNA at the population (bulk) level, in which the transcriptome of thousands or millions of cell are measured and averaged simultaneously. Although studies at the bulk level of gene expression (via bulk RNA sequencing, (Cloonan et al., 2008, Mortazavi et al., 2008)) are informative, any heterogeneity within a population of cells is largely concealed in these types of studies (Trapnell, 2015). Thanks to advances in sequencing technologies and the need for dissecting the cells of more complex tissues such as the brain, gene expression profiling at the individual-cell (single-cell) level can be carried out, as a powerful, high resolution tool for biological and disease discoveries (Saliba et al., 2014, Tanay and Regev, 2017). Statistical and computational methods to analyze single-cell RNA sequencing (scRNA-seq) data provide an opportunity for researchers to study the heterogeneity between individual cells and identify cell types based on their transcriptome (Andrews and Hemberg, 2018, Macaulay and Voet, 2014).

Table 3.1: Example of a raw count table from scRNA-Seq data.

	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	⋯	Gene 24,175
Cell 1	0	1	0	1	4	0	⋯	0
Cell 2	3	3	3	0	1	0	⋯	0
Cell 3	1	2	0	1	3	1	⋯	0
Cell 4	0	1	0	1	1	0	⋯	0
Cell 5	0	0	0	0	0	0	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Cell 1616	0	0	0	0	0	0	⋯	0

Indeed, single-cell genomics is bringing many new insights into the discoveries of complex and rare cell types including cancer stem cell and progenitor cells as well as many unrecognized cell types in various tissues such as neural immune and digestion systems, and studying diseases and cell development/lineage processes.

The process of single-cell RNA sequencing (scRNA-seq) includes four main steps.

A) Creating single-cell suspension using an isolation method.

B) Cell lysis and whole genome amplification.

C) Library preparation and sequencing.

D) Mapping the reads to the reference genome. Thus the expression of a gene can be measured by the number of reads aligned within its genomic coordinates (the so-called read counts or counts).

Because the amount of RNA material per cell is limited, some transcripts are not detected during sequencing leading to a high number of dropouts (zero counts) in the generated scRNA-seq data (Kharchenko et al., 2014). In addition, some of the observed zero counts correspond to true biological zeros (no expression). Therefore, it is not surprising to have a count matrix with more than 50% of its entries equal to zero. This leads to an excess of zeros (zero-inflation) in the count matrix of scRNA-seq data which is an important characteristic of these data. Table 3.1 demonstrates an example of raw read counts from single-cell RNA sequencing data (mouse embryonic stem cell data in Klein et al. (2015) before gene filtering analysed in Chapter 5) in which we can observe the feature of excess of zeros. This feature of zero-inflation creates some challenges for the statistical analysis of scRNA-seq data, where most methods that have been used for bulk RNA-seq data should be modified for scRNA-seq data or novel methods should be developed (Lähnemann et al., 2020).

Methods to analyse scRNA-seq data generally include the following steps:

1) **Pre-processing:** Pre-filtering, sometimes normalization of raw counts, quality control.

2) **Confounding factors:** Feature extraction, dimensionality reduction.

3) **Cell type identification:** Cell clusters.

4) **Cell type characterization:** Differentially expressed genes.

5) **Signature and driving force analysis:** Cell type specific gene signature, cell type specific driving forces.

In this thesis, the focus is on the clustering step (step 3 above). Cell type identification by clustering scRNA-seq data is challenging due to the excess of zero counts (zero-inflation). Therefore, our goal is to find a proper model-based clustering approach to deal with this type of zero-inflated data. Before describing my thesis contributions along with the proposed methodology (Sections 3.3 and 3.4 respectively), in Section 3.2 we review some of the published work on clustering scRNA-seq data.

3.2 Literature review on clustering scRNA-seq data

The process of clustering refers to partitioning data into subgroups or clusters where the observations falling into each group have more similarity to each other compared with points from the other clusters. In what follows, we introduce six different types of clustering methods that are commonly applied to scRNA-seq data.

- **Partitioning-based clustering:** This approach classifies data iteratively into K disjoint clusters based on using an optimization function that should be minimized, and the most common feasible methods to solve the optimization problem are K-means and K-medoids. In this method, we need to set the number of clusters K in advance (Ayramo and Karkkainen, 2006, Gan et al., 2007).

- **Hierarchical clustering:** For this type of clustering method, unlike the partitioning-based algorithm, there is no need to select the number of clusters initially. The results form a dendrogram, which is a tree-based representation. The most common hierarchical clustering is agglomerative clustering in which each data point is considered as a cluster in the first step, and the most similar clusters are merged iteratively (by using a proximity matrix) to reach either K clusters or one cluster (James et al., 2017).

- **Model-based clustering:** In this type of clustering, we assume that data arise from a probabilistic model, usually a finite mixture distribution. Model parameters and cluster

assignments are estimated using maximum likelihood estimation approaches such as the EM algorithm or Bayesian techniques such as the Gibbs sampler algorithm (Fraley and Raftery, 2002).

- **Graph-based clustering:** First, considering a similarity matrix of data, a graph representation of the data is constructed based on the k-nearest neighbour graph approach. Then, through applying the hierarchical clustering algorithm, the most similar subclusters are merged based on closeness and the relative interconnectivity of the clusters (Gan et al., 2007).

- **Density-based clustering:** In this kind of clustering, points concentrated in dense regions are considered cluster dense, and points are called *noise* if they do not belong to clusters (Gan et al., 2007). An example of a clustering method of this type is DBSCAN (Ester et al., 1996).

- **Deep-learning-based clustering:** This type of clustering applies a deep clustering network using first a deep autoencoder as an artificial neural network for unsupervised learning to represent high dimensional data in lower dimensions (Cheng and Ma, 2022, Lopez et al., 2018). Then, running K-means or other clustering algorithms on the representation vector learned by deep auto-encoder tends to give better results compared with the simple K-means for example.

Table 3.2 presents some of the existing tools for clustering scRNA-seq data based on each type of clustering approach described above. Each of these tools is briefly described as follows.

Partitioning-based tools include SC3 (Kiselev et al., 2017), SIMLR (Wang et al., 2017), RaceID (Grun et al., 2015), and RaceID2 (Grun et al., 2016). SC3 applies K-means to data after dimensionality reduction (conducted via principal component analysis - PCA) considering different types of dissimilarities. SIMLR is a dimensionality reduction technique and its framework can be used to perform K-means clustering after dimensionality reduction. RaceID is a tool developed to deal with rare cell type identification in a complex population of scRNA-seq data. After some preprocessing steps such as normalizing and removing counts with low gene expression levels, the K-means clustering method is used for the purpose of cell type identification. In RaceID2, K-means is replaced by K-medoids.

Several authors have considered hierarchical clustering. In CIDR (P. Lin and Ho, 2017), a PCA-based algorithm is used as an imputation method to reduce the impact of dropouts in scRNA-seq data. Then, hierarchical clustering is performed on the first few principal components, and the optimal number of clusters is determined using the Calinski-Harabasz index (Caliński and Harabasz, 1974). PcaReduce (žurauskienė and Yau, 2016) applies hierarchical

Type of clustering	Methods
Partitioning	SC3 (Kiselev et al., 2017) SIMLR (Wang et al., 2017) RaceID, RaceID2 (Grun et al., 2015, 2016)
Hierarchical	CellBIC (Kim et al., 2018) Corr (Jiang et al., 2018) CIDR (P. Lin and Ho, 2017) PcaReduce (žurauskienė and Yau, 2016) Tasic et al. (Tasic et al., 2016) MPath (Chen et al., 2016a) BACKSPIN (Zeisel et al., 2015) SINCERA (Guo et al., 2015)
Model-based	BasClu (Liu et al., 2019) DIMM-SC (Sun et al., 2018) TSCAN (Ji and Ji, 2016) BISCUIT (Prabhakaran et al., 2016)
Graph-based	Secuer (Wei et al., 2022) RGGC (Liu, 2021) Scanpy (Wolf et al., 2018) Park and Zhao (Park and Zhao, 2018) BiSNN-walk (SHI and HUANG, 2017) Seurat (Satija et al., 2015, Satija, 2015) Phenograph (Levine et al., 2015) SNN-Cliq (Xu and Su, 2015) SPARC (Li et al., 2015)
Density-based	PanoView (Hu et al., 2019) GiniClust (Lan Jiang and Yuan, 2016)
Deep-learning based	scGAC (Cheng and Ma, 2022) scGMAI (Yu et al., 2021) ScDCC (Tian et al., 2021) DCA (Eraslan et al., 2019) ScDeepCluster (Tian et al., 2019) scVI (Lopez et al., 2018)

Table 3.2: Some of the existing methods for clustering scRNA-seq data per type of clustering approach.

clustering to a reduced representation of the data obtained also via PCA. Zeisel et al. (2015) proposes BackSpin, an unsupervised biclustering method that sorts the expression matrix by cell-to-cell and gene-to-gene similarity without using dimensionality reduction. MPath (Chen et al., 2016a) performs hierarchical clustering by deriving multi-branching development using neighbourhood-based cell state transitions. Tasic et al. (2016) iteratively clusters cells in the principal component space and then splits cells into groups until they reach a termination criterion. SINCERA (Guo et al., 2015) is a computational pipeline for scRNA-seq data that includes hierarchical clustering to group cells according to their gene expression profiles. Kim et al. (2018) developed CellBIC, which implements a top-down approach of hierarchical clustering to cluster scRNA-seq data based on their modality in the gene expression distribution. Jiang et al. (2018) proposes Corr, which uses a similarity metric based on cell-to-cell differentiability correlations in the hierarchical clustering framework.

Model-based clustering methods are studied in TSCAN (Ji and Ji, 2016), BISCUIT (Prabhakaran et al., 2016), DIMM-SC (Sun et al., 2018), and BasClu (Liu et al., 2019). After dimensionality reduction via PCA, TSCAN considers a mixture of multivariate Gaussian distributions to cluster cells based on their gene expression. BISCUIT is a hierarchical Bayesian Dirichlet process mixture model for clustering scRNA-seq data assuming that gene log counts follow a Gaussian distribution. BasClu extends the Dirichlet process mixture model of BISCUIT by introducing a sequence of latent binary indicators to represent whether genes are expressed or not to address the excess of zero counts. In addition, BasClu accounts for dropout events by also modelling the probability of missing data across genes. DIMM-SC proposes a Dirichlet mixture model for clustering droplet-based scRNA-seq data assuming that gene read counts follow a multinomial distribution.

PhenoGraph (Levine et al., 2015), Seurat (Satija et al., 2015, Satija, 2015), and SCANPY (Wolf et al., 2018) are graph-based algorithms that are applied to scRNA-seq data after PCA for dimensionality reduction. In PhenoGraph, the cells are partitioned into groups by clustering a graph based on their phenotypic similarity; that is, first, for each cell, the k nearest neighbours are found, which results in N sets with K neighbourhoods, and then a weighted graph is built on these sets. Seurat is an R package that spatially maps single cells yielding a transcriptome-wide map of spatial patterning. SCANPY is a toolkit python-based package for analyzing scRNA-seq data, including pre-processing (comparable to Seurat), visualization, clustering (similar to PhenoGraph), pseudo time and trajectory inference, differential expression testing, and simulation of gene regulatory networks. SNN-Cliq (Xu and Su, 2015) is a graph-based algorithm obtained by combining the quasi-clique-based algorithm (which is a graph-theory-based algorithm introduced earlier by the same authors) with the shared nearest neighbour (SNN) similarity measures (Houle et al., 2010). SHI and HUANG (2017) proposed

an iterative biclustering approach (BiSNN-Walk) based on the SNN-cliq algorithm. BiSNN-walk returns a ranked list of clusters, which indicate the cluster's reliability, and ranks the genes in a gene cluster based on their affiliation levels to the associated cell cluster. Park and Zhao (2018) proposes a spectral clustering method based on imposing the sparse structure of scRNA-seq data on a similarity matrix and then shrinking the pairwise differences on the rows of the target matrix. Li et al. (2015) proposed SPARC as a method that uses a similarity metric based on the relationship among cells and includes an outlier detection method. Liu (2021) introduces a regularization graphical clustering method (RGGC) based on higher-order correlations and subspace learning. Wei et al. (2022) presents a spectral clustering algorithm (Secuer) for scRNA-seq data that is an anchor-based bipartite graph representation.

GiniClust (Lan Jiang and Yuan, 2016) is a density-based algorithm that borrows the idea of the Gini index from social sciences to detect rare cell types. In this algorithm, first, a bidirectional Gini index is defined to identify genes that are specifically unexpressed in a rare cell type. Then after normalizing these Gini index values, the high Gini genes are selected. Based on the gene expression of these high Gini genes, cell clusters are identified by applying DBSCAN (Ester et al., 1996). Another density-based method is PanoView (Hu et al., 2019). PanoView uses a density-based method called ordering local maximum by a convex hull to iteratively search cell types in a principal component space.

Some clustering algorithms based on deep learning are also applied to scRNA-seq data. Yu et al. (2021) proposes a Gaussian mixture method called scGMAI based on deep autoencoder networks and independent component analysis to cluster cell types from scRNA-seq data. Cheng and Ma (2022) presents a single-cell graph attentional clustering called scGAC for clustering scRNA-seq data using a four-step approach: 1) constructing a cell graph, 2) refining the cell graph by using network denoising, 3) learning the clustering representation of cells by a graph attentional autoencoder, and 4) finding the cell types clusters. Eraslan et al. (2019) introduces a deep count autoencoder network for denoising scRNA-seq data followed by K-means clustering. DCA considers the overdispersion and sparsity of the count data by using a zero-inflated negative binomial noise model and nonlinear gene-gene or gene-dispersion interactions. Lopez et al. (2018) proposes scVI as a Bayesian hierarchical model in which deep neural networks are used to specify the conditional distributions under a variational inference approach. Tian et al. (2019) developed ScDeepCluster, which integrates a zero-inflated negative binomial model-based autoencoder with clustering loss and deep embedding clustering. Tian et al. (2021) introduces a model-based deep embedding clustering method for scRNA-seq data called scDCC, which integrates prior knowledge into constrained information via a loss function.

3.3 Thesis Contribution

As mentioned in Section 3.1, the excess of zero counts (zero-inflation) in single-cell RNA-seq data causes some challenges in analyzing these data compared to bulk RNA-seq data. As seen in Section 3.2, most available clustering tools do not directly tackle the zero-inflation of scRNA-seq data. Instead, in most studies, this issue of the zero-inflation is tackled in the dimensionality reduction or feature selection step of the analysis, which is carried out before clustering. For example, Pierson and Yau (2015) developed a dimensionality reduction technique (Zero Inflated Factor Analysis, ZIFA), which takes into account the zero inflation of scRNA-seq data, and they demonstrated that in comparison with other dimensionality reduction methods, ZIFA performs better in both simulated and biological data sets. Also, Tian et al. (2019, 2021) consider a zero-inflated negative binomial (ZINB) model-based loss function autoencoder for dimensionality reduction followed by a deep-embedded clustering algorithm. Qiu (2020) tackles the zero-inflation problem by first binarizing the scRNA-seq data, turning all the non-zero observations into one, and then proposing a non-probabilistic multi-step clustering method to cluster the binarized data.

Thus, in this thesis, the goal is to cluster scRNA-seq data based on their gene expression profiles through a model-based approach which takes into account the zero-inflated distribution of the raw counts (number of reads aligned to each gene). We assume a probabilistic model in which scRNA-seq data follows a mixture of either zero-inflated Poisson or zero-inflated negative binomial distributions. We then allow the logarithm of the rate parameter of the Poisson or negative binomial component to be a linear combination of some fixed and known covariates such as batch effects and cell size factor. Estimation of cluster assignments and model parameters is conducted via the EM algorithm. We implement our proposed methodology using R, and the code is available online at `https://github.com/desouzalab/em-mzip`.

3.4 Proposed methodology

Since in our model-based clustering approach, the assumed model for the scRNA-seq data is a mixture of zero-inflated Poisson (ZIP) or zero-inflated negative binomial (ZINB) distributions, we first introduce these distributions in Section 3.4.1. Section 3.4.2 presents a literature review in Statistics on this framework. The proposed mixture model for zero-inflated Poisson counts is presented in Section 3.5. Sections 3.5.1 and 3.5.2 describe the parameter inference via the EM algorithm without and with covariates, respectively. In Section 3.6 we present the case of a mixture of ZINB distributions.

3.4.1 Poisson and negative binomial zero-inflated models

A ZIP model is in the form of a mixture distribution with two components. The first component corresponds to the zero counts and the second one to the non-zero counts. Let $y_1, \dots y_n$ be a set of independent observations arising from a ZIP model. We can write for $i = 1, \dots, n$:

$$P(Y_i = y_i) = \begin{cases} \phi + (1-\phi)e^{-\lambda_i} & \text{for } y_i = 0 \\ (1-\phi)\frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!} & \text{for } y_i = 1, 2, 3, \dots, \end{cases} \tag{3.1}$$

where λ_i is the Poisson rate and $0 < \phi < 1$ is the probability of always (perfect) zero.

Similarly, a ZINB model has the same form except for the non-zero component in which the Poisson distribution is replaced by the negative binomial distribution and, therefore, we can write:

$$P(Y_i = y_i) = \begin{cases} \phi + (1-\phi)(\frac{1}{1+\alpha\mu_i})^{\frac{1}{\alpha}} & \text{for } y_i = 0 \\ (1-\phi)\frac{\Gamma(y_i+\alpha)}{\Gamma(y_i+1)\Gamma(\alpha)}(\frac{1}{1+\alpha\mu_i})^{\frac{1}{\alpha}}(1-\frac{1}{1+\alpha\mu_i})^{y_i} & \text{for } y_i = 1, 2, 3, \dots, \end{cases} \tag{3.2}$$

where α, $\frac{1}{\alpha}$ and μ_i are the dispersion, size, and mean (rate) parameters of the negative-binomial distribution, respectively. The function $\Gamma(\cdot)$ is the gamma function.

One can consider covariates (ZIP or ZINB regression) and model the rate parameter λ_i (for ZIP) or μ_i (for ZINB) via a log link as follows:

$$\log(\lambda_i) = x_i^T\beta, \text{ or}$$

$$\log(\mu_i) = x_i^T\beta,$$

where β is the unknown vector of coefficients and x_i is the vector of known covariates for observation i, for $i = 1, \dots, n$ (Faraway, 2016, Workie and Azene, 2021). The probability of always zero can also be modeled as a linear combination of covariates via a logit link.

3.4.2 Literature review in Statistics

Some papers in Statistics have considered ZIP or ZINB regression models (no mixture) to analyze different types of data. For instance, Lyashevska et al. (2016) and Pilosof et al. (2012) used ZIP and ZINB regression models, respectively, to model abundance of species. Xue et al. (2020) introduced a zero-inflated Poisson regression model with random intercepts to analyze the amount of daily and weekly physical activity of Hispanic/Latino adults. Xue et al. (2020) conducted maximum likelihood parameter estimation via the Gaussian quadrature technique, which is implemented in the R package *GLMMadaptive*.

Other authors have studied mixtures of ZIP regression models. Lim et al. (2014) proposed a ZIP regression model where the Poisson component is assumed to be a mixture of Poisson distributions. Parameters are estimated via the EM algorithm with an embedded iteratively re-weighted least squares method. As an application, Lim et al. (2014) considers a dataset on dental caries in adolescents. Chen et al. (2016b) presented a mixture of zero-inflated Poisson regression models with random effects to analyze correlated multilevel data. For obtaining the maximum-likelihood estimates of the parameters, the authors developed a stochastic EM algorithm. The Bayesian Information Criterion (BIC) was used for comparing models with different latent classes. The proposed methodology was used to analyze data from a survey on adolescent fitness.

In the R programming language, some packages include functions for parameter estimation of zero-inflated models for counts. The function *zeroinfl* in the *pscl* package (Jackman et al., 2020) can be used to fit zero-inflated regression models for counts (Poisson or negative binomial) via the maximum likelihood method. The functions *zipoisson* and *zinegbinomial*, available in the *VGAM* package (Yee and Moler, 2022), can be applied to fit a zero-inflated Poisson or zero-inflated negative binomial distribution via maximum likelihood. The function *ZIP* in the *ZIPBayes* package (Zhang and Yi, 2021) can be used to estimate parameters of a zero-inflated Poisson model via the Markov Chain Monte Carlo (MCMC) algorithm. In the *mpath* package (Wang et al., 2022), the function *zipath* fits a zero-inflated regression model for counts by using regularization methods such as a LASSO or elastic net.

The function *bzinb* (in the package *bzinb* Cho et al. (2019)) can be used to find the maximum likelihood parameter estimates for a bivariate zero-inflated negative binomial (ZINB) model. In order to fit zero-inflated count models via MCMC, the function *zic* in the package under the same name (*zic*) can be applied (Jochmann, 2017). And, finally, in the package *poisson.glm.mix* (Papastamoulis, 2022) different functions are available for fitting a mixture of Poisson generalized linear models via the EM algorithm. However, this package does not consider zero inflation.

So far, none of these existing tools deals with a mixture of zero-inflated counts when the data are in a matrix structure as in our proposed methodology; see matrix (3.3) below.

3.5 The proposed mixture model for ZIP counts

Let Y_{ng} be a random variable for the number of read counts aligned to gene g in cell n, for $g = 1, \ldots, G$ and $n = 1, \ldots, N$, where Y_{ng} takes a value in $\{0, 1, 2, 3, \ldots\}$. So, the observed data

can be written in the following matrix format:

$$\mathbf{y} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1G} \\ y_{21} & y_{22} & \cdots & y_{2G} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NG} \end{pmatrix}. \tag{3.3}$$

Suppose that there are $K << N$ clusters of cells and let $\boldsymbol{Z} = \{Z_{11}, \ldots, Z_{NK}\}$ be the set of latent random variables indicating the true cell cluster assignments, that is:

$$Z_{nk} = \begin{cases} 1 & \text{if cell } n \text{ belongs to cluster } k, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

We can also write $\boldsymbol{Z}$ as the following matrix:

$$\boldsymbol{Z} = \begin{pmatrix} Z_{11} & Z_{12} & \cdots & Z_{1K} \\ Z_{21} & Z_{22} & \cdots & Z_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{N1} & Z_{N2} & \cdots & Z_{NK} \end{pmatrix},$$

with $\sum_{k=1}^{K} Z_{nk} = 1$, $P(Z_{nk} = 1) = \pi_k$ for $n = 1, \ldots, N$, $k = 1, \ldots, K$, and $\sum_{k=1}^{K} \pi_k = 1$. We assume that given Z_{nk}, that is, given that cell n belongs to cluster k, genes in cell n are independent and follow a ZIP distribution with parameters that depend on cluster k. Let $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k\}$ be the set of all models parameters with $\boldsymbol{\theta}_k = \{\pi_k, \phi_k, \boldsymbol{\lambda}_k\}$, and $\boldsymbol{\lambda_k} = \{\lambda_{1k}, \ldots, \lambda_{gk}\}$. Thus, we can write the probability mass function (pmf) for each cell as the following mixture of zero-inflated Poisson (ZIP) distributions:

$$p(\mathbf{y}_n \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\mathbf{y}_n \,|\, \boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \prod_{g=1}^{G} p(y_{ng} \,|\, \lambda_{gk}, \phi_k),$$

where

$$p(y_{ng} \,|\, \lambda_{gk}, \phi_k) = \begin{cases} \phi_k + (1 - \phi_k) e^{-\lambda_{gk}} & \text{if } y_{ng} = 0 \\ (1 - \phi_k) \frac{e^{-\lambda_{gk}} \lambda_{gk}^{y_{ng}}}{y_{ng}!} & \text{if } y_{ng} = 1, 2, 3, \ldots, \end{cases} \tag{3.5}$$

in which λ_{gk} is the Poisson rate parameter and ϕ_k is the probability of always zero. We can also

write (3.5) as:

$$p(y_{ng} \mid \lambda_{gk}, \phi_k) = \begin{cases} \phi_k & \text{if } y_{ng} \text{ belongs to the zero state;} \\ (1-\phi_k)\frac{e^{-\lambda_{gk}}\lambda_{gk}^{y_{ng}}}{y_{ng}!} & \text{if } y_{ng} \text{ belongs to the Poisson state.} \end{cases} \tag{3.6}$$

By assuming independence across cells, the observed-data likelihood based on all cells is given by:

$$\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{y}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \prod_{g=1}^{G} p(y_{ng} \mid \lambda_{gk}, \phi_k)$$

Thus, the observed data log-likelihood can be written as:

$$\ell(\boldsymbol{\theta} \mid \boldsymbol{y}) = \sum_{n=1}^{N} \log\Big[\sum_{k=1}^{K} \pi_k \prod_{g=1}^{G} p(y_{ng} \mid \lambda_{gk}, \phi_k)\Big]. \tag{3.7}$$

The goal is to find the parameter estimates that maximize (3.7). To tackle this problem, we develop an EM algorithm to iteratively find the parameter estimates. To obtain the parameter estimates using the EM framework, we consider the true latent cluster assignments Z_{nk} for $n = 1, \ldots, N$ and $k = 1, \ldots, K$ as in (3.4) and we also introduce another set of hidden variables $\boldsymbol{U}$ defined as follows:

$$\boldsymbol{U} = \begin{pmatrix} U_{11} & U_{12} & \cdots & U_{1G} \\ U_{21} & U_{22} & \cdots & U_{2G} \\ \vdots & \vdots & \ddots & \vdots \\ U_{N1} & U_{N2} & \cdots & U_{NG} \end{pmatrix}$$

where

$$U_{ng} = \begin{cases} 1 & \text{if } y_{ng} \text{ is from perfect zero state,} \\ 0 & \text{if } y_{ng} \text{ is from Poisson state.} \end{cases}$$

This latent indicator variable is drawn from a Bernoulli distribution, $U_{ng} \sim \text{Bernoulli}(\phi_k)$, with probability of success (i.e., probability of always zero) ϕ_k defined as $\phi_k = P(U_{ng} = 1 | Z_{nk} = 1)$, which depends on the cluster k.

In the E-step of the EM algorithm, the conditional expectation of the complete-data log-likelihood given the observed data and current parameter estimates is obtained. Then, in the M-step, the expectation from the E-step is maximized with respect to each parameter of interest. These two steps are implemented iteratively until convergence. Now, considering the observed

counts and the introduced latent variables, the complete-data likelihood can be written as:

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{u},\boldsymbol{z}) = \\
&= \prod_{n=1}^{N} p(\boldsymbol{y}_n,\boldsymbol{u}_n,\boldsymbol{z}_n\,|\,\boldsymbol{\theta}) \\
&= \prod_{n=1}^{N} \Big(p(\boldsymbol{y}_n\,|\,\boldsymbol{u}_n,\boldsymbol{z}_n,\boldsymbol{\theta}) \times p(\boldsymbol{u}_n\,|\,\boldsymbol{z}_n,\boldsymbol{\theta}) \times p(\boldsymbol{z}_n\,|\,\boldsymbol{\theta}) \Big) \\
&= \prod_{n=1}^{N} \prod_{k=1}^{K} \Big(p(\boldsymbol{y}_n\,|\boldsymbol{u}_n,\boldsymbol{\theta}_k) \times p(\boldsymbol{u}_n\,|\,\boldsymbol{\theta}_k) \times p(z_{nk}\,|\,\boldsymbol{\theta}_k) \Big)^{z_{nk}} \\
&= \prod_{n=1}^{N} \prod_{k=1}^{K} \prod_{g=1}^{G} p(y_{ng}\,|\,u_{ng},\lambda_{gk})^{z_{nk}} \times \prod_{n=1}^{N} \prod_{k=1}^{K} \prod_{g=1}^{G} p(u_{ng}\,|\,\phi_k)^{z_{nk}} \times \prod_{n=1}^{N} \prod_{k=1}^{K} p(z_{nk}\,|\,\pi_k)^{z_{nk}} \\
&= \prod_{n=1}^{N} \prod_{k=1}^{K} \prod_{g=1}^{G} \left(\frac{e^{-\lambda_{gk}} \lambda_{gk}^{y_{ng}}}{y_{ng}!} \right)^{(1-u_{ng})z_{nk}} \times \prod_{n=1}^{N} \prod_{k=1}^{K} \prod_{g=1}^{G} \Big(\phi_k^{u_{ng}} (1-\phi_k)^{(1-u_{ng})} \Big)^{z_{nk}} \times \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}}.
\end{aligned}
$$

By applying the logarithm, the complete-data log-likelihood function is:

$$
\begin{aligned}
\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{u},\boldsymbol{z}) &= \log \mathcal{L}(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{u},\boldsymbol{z}) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{g=1}^{G} z_{nk}(1-u_{ng}) \log \Big(\frac{e^{-\lambda_{gk}} \lambda_{gk}^{y_{ng}}}{y_{ng}!} \Big) \\
&+ \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{g=1}^{G} \Big(z_{nk} u_{ng} \log \phi_k + z_{nk}(1-u_{ng}) \log(1-\phi_k) \Big) \\
&+ \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log \pi_k.
\end{aligned}
$$

In what follows, we present the E and M steps of our proposed EM algorithm for the cases without covariates (Section 3.5.1) and with covariates (Section 3.5.2).

3.5.1 EM for the ZIP mixture model without covariates

E-Step: We first write the conditional expectation of the complete-data log-likelihood given the current estimates of the parameters, $\boldsymbol{\theta}^{(t)} = \{\boldsymbol{\lambda}^{(t)}, \boldsymbol{\phi}^{(t)}, \boldsymbol{\pi}^{(t)}\}$, and the observed data $\boldsymbol{y}$:

$$
\begin{aligned}
Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) &= E\Big[\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{u},\boldsymbol{z})\,|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big] \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G} E\Big[Z_{nk}(1-U_{ng})\,|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log\left(\frac{e^{-\lambda_{gk}}\lambda_{gk}^{y_{ng}}}{y_{ng!}}\right) \\
&+ \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G}\left(E\Big[Z_{nk}U_{ng}\,|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log\phi_k + E\Big[Z_{nk}(1-U_{ng})\,|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log(1-\phi_k)\right) \\
&+ \sum_{n=1}^{N}\sum_{k=1}^{K} E\Big[Z_{nk}\,|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log\pi_k. \qquad (3.8)
\end{aligned}
$$

Then, the expectations in (3.8) are calculated as follows:

$$
\begin{aligned}
\hat{Z}_{nk}^{(t)} &= E\Big[Z_{nk}\,|\,\boldsymbol{y}\,,\,\boldsymbol{\theta}^{(t)}\Big] = E\Big[Z_{nk}\,|\,\boldsymbol{y}_n\,,\,\boldsymbol{\theta}^{(t)}\Big] \\
&= p\Big(Z_{nk}=1\,|\,\boldsymbol{y}_n\,,\,\boldsymbol{\theta}^{(t)}\Big) \\
&= \frac{p\Big(\boldsymbol{y}_n, Z_{nk}=1\,|\,\boldsymbol{\theta}^{(t)}\Big)}{p\Big(\boldsymbol{y}_n\,|\,\boldsymbol{\theta}^{(t)}\Big)} \\
&= \frac{p\Big(\boldsymbol{y}_n\,|\,Z_{nk}=1,\boldsymbol{\theta}^{(t)}\Big)\,p\Big(Z_{nk}=1\,|\,\boldsymbol{\theta}^{(t)}\Big)}{\sum_{j=1}^{K} p\Big(\boldsymbol{y}_n\,|\,Z_{nj}=1,\boldsymbol{\theta}^{(t)}\Big)\,p\Big(Z_{nj}=1\,|\,\boldsymbol{\theta}^{(t)}\Big)} \\
&= \frac{p\Big(Z_{nk}=1\,|\,\boldsymbol{\theta}^{(t)}\Big)\prod_{g=1}^{G} p\Big(y_{ng}\,|\,Z_{nk}=1,\boldsymbol{\theta}^{(t)}\Big)}{\sum_{j=1}^{K} p\Big(Z_{nj}=1\,|\,\boldsymbol{\theta}^{(t)}\Big)\prod_{g=1}^{G} p\Big(y_{ng}\,|\,Z_{nj}=1,\boldsymbol{\theta}^{(t)}\Big)} \\
&= \frac{\pi_k^{(t)}\prod_{g=1}^{G} p\Big(y_{ng}\,|\,\lambda_{kg}^{(t)},\phi_k^{(t)}\Big)}{\sum_{j=1}^{K}\pi_j^{(t)}\prod_{g=1}^{G} p\Big(y_{ng}\,|\,\lambda_{jg}^{(t)},\phi_j^{(t)}\Big)}, \qquad (3.9)
\end{aligned}
$$

and

$$
\begin{aligned}
E\big[Z_{nk}U_{ng}\,\big|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\big] &= p(Z_{nk}=1,U_{ng}=1\,\big|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}) \\
&= p(U_{ng}=1\,\big|\,Z_{nk}=1,y_{ng},\boldsymbol{\theta}^{(t)})\times p(Z_{nk}=1\,\big|\,\boldsymbol{y}_n,\boldsymbol{\theta}^{(t)}) \\
&= \hat{U}_{ngk}^{(t)}\hat{Z}_{nk}^{(t)},
\end{aligned}
$$

where $\hat{Z}_{nk}^{(t)}$ is as in (3.9) and $\hat{U}_{ngk}^{(t)}$ is given by:

$$
\begin{aligned}
\hat{U}_{ngk}^{(t)} &= p(U_{ng}=1 \,|\, Z_{nk}=1, y_{ng}, \boldsymbol{\theta}^{(t)}) \\
&= \frac{p(y_{ng}, U_{ng}=1, Z_{nk}=1 \,|\, \boldsymbol{\theta}^{(t)})}{p(y_{ng}, Z_{nk}=1 \,|\, \boldsymbol{\theta}^{(t)})} \\
&= \frac{p(y_{ng} \,|\, U_{ng}=1, Z_{nk}=1, \boldsymbol{\theta}^{(t)}) \times p(U_{ng}=1 \,|\, Z_{nk}=1, \boldsymbol{\theta}^{(t)}) \times p(Z_{nk}=1 \,|\, \boldsymbol{\theta}^{(t)})}{p(y_{ng} \,|\, Z_{nk}=1, \boldsymbol{\theta}^{(t)}) \times p(Z_{nk}=1 \,|\, \boldsymbol{\theta}^{(t)})} \\
&= \frac{\cancel{\pi_k^{(t)}} \phi_k^{(t)} p(y_{ng} \,|\, U_{ng}=1, Z_{nk}=1, \boldsymbol{\theta}^{(t)})}{\cancel{\pi_k^{(t)}} p(y_{ng} \,|\, \lambda_{gk}^{(t)}, \phi_k^{(t)})} \\
&= \frac{\phi_k^{(t)} p(y_{ng} \,|\, U_{ng}=1, Z_{nk}=1, \boldsymbol{\theta}^{(t)})}{p(y_{ng} \,|\, \lambda_{gk}^{(t)}, \phi_k^{(t)})}.
\end{aligned} \tag{3.10}
$$

So, from (3.10), we can write:

$$
\hat{U}_{ngk}^{(t)} = \begin{cases} \dfrac{\phi_k^{(t)}}{\left(\phi_k^{(t)} + (1-\phi_k^{(t)}) e^{-\lambda_{gk}^{(t)}}\right)} & \text{if } y_{ng} = 0, \\ 0 & \text{if } y_{ng} = 1, 2, \ldots. \end{cases} \tag{3.11}
$$

Note that $\hat{U}_{ngk}^{(t)}$ in (3.11) is separated into two cases because if $U_{ng} = 1$, y_{ng} can only be equal to zero, otherwise, if y_{ng} takes a non-zero count value, it definitely arises from the Poisson state.

Using the calculated values $\hat{Z}_{nk}^{(t)}$ and $\hat{U}_{ngk}^{(t)}$ from Equations (3.9) and (3.11), we can rewrite $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ in (3.8) as:

$$
Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) = Q_1(\boldsymbol{\pi}; \boldsymbol{\pi}^{(t)}) + Q_2(\boldsymbol{\phi}; \boldsymbol{\phi}^{(t)}) + Q_3(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{(t)}), \tag{3.12}
$$

where

$$
\begin{aligned}
Q_1(\boldsymbol{\pi}; \boldsymbol{\pi}^{(t)}) &= \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{Z}_{nk}^{(t)} \log(\pi_k), \\
Q_2(\boldsymbol{\phi}; \boldsymbol{\phi}^{(t)}) &= \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{g=1}^{G} \left[\hat{Z}_{nk}^{(t)} \hat{U}_{ngk}^{(t)} \log(\phi_k) + \hat{Z}_{nk}^{(t)} (1 - \hat{U}_{ngk}^{(t)}) \log(1-\phi_k) \right], \text{ and} \\
Q_3(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{(t)}) &= \sum_{n=1}^{N} \sum_{k=1}^{K} \sum_{g=1}^{G} \left\{ \hat{Z}_{nk}^{(t)} (1 - \hat{U}_{ngk}^{(t)}) \Big[-\lambda_{gk} + y_{ng} \log(\lambda_{gk}) - \log(y_{ng}!) \Big] \right\}.
\end{aligned}
$$

M-step: Using $\hat{Z}_{nk}^{(t)}$ and $\hat{U}_{ngk}^{(t)}$ calculated in the E-step, in this step we find the updated parameters $\boldsymbol{\theta}^{(t+1)} = (\boldsymbol{\lambda}^{(t+1)}, \boldsymbol{\phi}^{(t+1)}, \boldsymbol{\pi}^{(t+1)})$ that maximize $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ given in Equation (3.12).

First, we find each $\pi_k^{(t+1)}$, through considering the restriction that $\sum_{k=1}^{K} \pi_k = 1$, and using the

augmented function and Lagrange multiplier as follows.

$$g(\boldsymbol{\pi}, \gamma) = Q_1(\boldsymbol{\pi}; \boldsymbol{\pi}^{(t)}) - \gamma \times \Big(\sum_{k=1}^{K} \pi_k - 1 \Big) = \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{Z}_{nk}^{(t)} \log(\pi_k) - \gamma \times \Big(\sum_{k=1}^{K} \pi_k - 1 \Big). \tag{3.13}$$

Taking the derivative of the (3.13) w.r.t. π_k and setting it to zero, leads to:

$$\begin{aligned} &\frac{\partial g(\boldsymbol{\pi}, \gamma)}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\hat{Z}_{nk}^{(t)}}{\pi_k} - \gamma = 0 \\ &\Rightarrow \frac{\sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}}{\pi_k} = \gamma \Rightarrow \pi_k = \frac{1}{\gamma} \sum_{n=1}^{N} \hat{Z}_{nk}^{(t)} \end{aligned} \tag{3.14}$$

Moreover, differentiating (3.13) w.r.t. γ leads to:

$$\frac{\partial g(\boldsymbol{\pi}, \gamma)}{\partial \gamma} = -\Big(\sum_{k} \pi_k - 1 \Big) = 0 \Rightarrow \sum_{k=1}^{K} \pi_k = 1. \tag{3.15}$$

Now summing both sides of (3.13) over k we obtain:

$$\begin{aligned} \sum_{k=1}^{K} \pi_k &= \frac{1}{\gamma} \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{Z}_{nk}^{(t)} \\ \Rightarrow 1 &= \frac{1}{\gamma} N \Rightarrow \gamma = N \end{aligned} \tag{3.16}$$

From (3.14) and (3.16), the updated estimate for π_k is:

$$\pi_k^{(t+1)} = \frac{\sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}}{N}. \tag{3.17}$$

The updated $\phi_k^{(t+1)}$ is obtained as follows:

$$\begin{aligned}
\frac{\partial Q_2(\boldsymbol{\phi};\boldsymbol{\phi}^{(t)})}{\partial \phi_k} &= \sum_{n=1}^{N}\sum_{g=1}^{G}\left(\hat{Z}_{nk}\hat{U}_{ngk}\frac{1}{\phi_k} - \hat{Z}_{nk}^{(t)}\left(1-\hat{U}_{ngk}^{(t)}\right)\frac{1}{1-\phi_k}\right) = 0 \\
&\Rightarrow \sum_{n=1}^{N}\sum_{g=1}^{G}\left(\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}\frac{1}{\phi_k} - \hat{Z}_{nk}\frac{1}{1-\phi_k} + \hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}\frac{1}{1-\phi_k}\right) = 0 \\
&\Rightarrow \sum_{n=1}^{N}\sum_{g=1}^{G}\left(\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}\left(\frac{1}{\phi_k}+\frac{1}{1-\phi_k}\right) - \hat{Z}_{nk}^{(t)}\frac{1}{1-\phi_k}\right) = 0 \\
&\Rightarrow \sum_{n=1}^{N}\sum_{g=1}^{G}\left(\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}\left(\frac{1-\phi_k}{\phi_k}+1\right) - \hat{Z}_{nk}^{(t)}\right) = 0 \\
&\Rightarrow \sum_{n=1}^{N}\sum_{g=1}^{G}\left(\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}\frac{1}{\phi_k} - \hat{Z}_{nk}^{(t)}\right) = 0 \\
&\Rightarrow \frac{1}{\phi_k}\sum_{n=1}^{N}\sum_{g=1}^{G}\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)} = \sum_{n=1}^{N}\sum_{g=1}^{G}\hat{Z}_{nk}^{(t)}
\end{aligned}$$

Therefore,

$$\phi_k^{(t+1)} = \frac{\sum_{n=1}^{N}\sum_{g=1}^{G}\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}}{G\sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}}. \tag{3.18}$$

And, finally, we obtain the updated $\lambda_{gk}^{(t+1)}$ as follows:

$$\frac{\partial Q_3(\boldsymbol{\lambda};\boldsymbol{\lambda}^{(t)})}{\partial \lambda_{gk}} = \sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\left[\frac{y_{ng}}{\lambda_{gk}} - 1\right] = 0,$$

so that

$$\lambda_{gk}^{(t+1)} = \frac{\sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})y_{ng}}{\sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})}. \tag{3.19}$$

Note that the conditional expected value of each Z_{nk} in Equation (3.9), obtained at the last iteration t^*, is used to infer the cluster assignment of each cell. Thus, we obtain the decision that cell n belongs to cluster k if that cluster is the one with the highest expected value (highest probability); that is:

$$\hat{Z}_{nk} = \begin{cases} 1 & \text{if } \hat{Z}_{nk}^{(t^*)} = \max\limits_{j\in\{1,\ldots,K\}}\hat{Z}_{nj}^{(t^*)}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

Algorithm 1 summarizes the EM algorithm for the ZIP mixture model without covariates.

Algorithm 1 EM algorithm for the ZIP mixture model without covariates

Input: $\boldsymbol{y}$: matrix of data; $\boldsymbol{\theta}^{(0)} = (\boldsymbol{\pi}^{(0)}, \boldsymbol{\phi}^{(0)}, \boldsymbol{\lambda}^{(0)})$: initial parameters; *tol*: tolerance; m: maximum number of iterations.

Output: optimal set of parameters $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\lambda}})$ and $\hat{Z}_{nk}$ and $\hat{U}_{ngk}$ for all n, g and k.

initial $t = 0$ (iteration number);
repeat
    Start E-step:
    Calculate $\hat{Z}_{nk}^{(t)}$, for all n and k, as in (3.9);
    Calculate $\hat{U}_{ngk}^{(t)}$, for all n, g, and k, as in (3.11).
    Start M-Step using the $\hat{Z}_{nk}^{(t)}$'s and $\hat{U}_{ngk}^{(t)}$'s:
    Compute $\pi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.17);
    Compute $\phi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.18);
    Compute $\lambda_{gk}^{(t+1)}$, for $k = 1, \ldots, K, g = 1, \ldots, G$, as in (3.19).
until $[\ell(\boldsymbol{\theta}^{(t+1)} \,|\, \boldsymbol{y}) - \ell(\boldsymbol{\theta}^{(t)} \,|\, \boldsymbol{y})] \leq tol$ or maximum number of iterations is achieved.

3.5.2 EM for the ZIP mixture model with covariates

In this case, similarly to Zhang et al. (2019), we assume that the Poisson rate parameters depend on a linear combination of covariates via a log link function as follows:

$$\log(\lambda_{ngk}) = \log(T_n) + \rho_{gk} + \beta_{0g} + \sum_{p=1}^{P} \beta_{pg} x_{np}, \tag{3.21}$$

for $n = 1, \ldots, N$, $g = 1, \ldots, G$, $k = 1, \ldots, K$, and $p = 1, \ldots, P$, where T_n is a fixed size factor (also known as a Poisson offset variable) for cell n (e.g., sequencing library size), β_{0g} is a baseline expression for gene g, ρ_{gk} is the fixed effect of cluster k on gene g, $x_{n1}, \ldots, x_{np}$ are P known covariates for cell n (e.g., batch and treatment effects), and $\beta_{1g}, \ldots, \ldots, \beta_{Pg}$ their corresponding unknown coefficients. We note that the authors of Zhang et al. (2019) do not consider zero inflation in their proposed cell classification tool.

This model with covariates can also be called a mixture of generalized ZIP regression models. We use the EM algorithm to find the estimated parameters and inferred cluster assignments. Therefore, considering the complete-data log-likelihood as:

$$\ell(\boldsymbol{\theta} \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) = \sum_{n=1}^{N} \sum_{g=1}^{G} \sum_{k=1}^{K} \Big[z_{nk} \log(\pi_k) + z_{nk} u_{ng} \log(\phi_k) + z_{nk}(1 - u_{ng}) \log(1 - \phi_k) + z_{nk}(1 - u_{ng}) \log(p(y_{ng} \,|\, \rho_{gk}, \beta_{0g}, \beta_{pg})) \Big],$$

where $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\rho}, \boldsymbol{\beta}_0, \boldsymbol{\beta})$, with

$$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)^T,$$

$$\boldsymbol{\phi} = (\phi_1, \ldots, \phi_k),^T$$

$$\boldsymbol{\rho} = (\rho_{11}, \ldots, \rho_{G1}, \ldots, \rho_{1K}, \ldots, \rho_{GK})^T,$$

$$\boldsymbol{\beta}_0 = (\beta_{01}, \ldots, \beta_{0G})^T, \text{ and}$$

$$\boldsymbol{\beta} = (\beta_{11}, \ldots, \beta_{p1}, \ldots, \beta_{1G}, \ldots, \beta_{PG})^T.$$

E-Step: Similarly to Section 3.5.1, first in the E-step, we compute the conditional expectation of the complete-data log-likelihood given the observed data and the current parameter estimates as follows:

$$\begin{aligned}
Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)}) &= E\Big[\ell(\boldsymbol{\theta} \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}^{(t)}\Big] \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} E\Big[Z_{nk} \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}^{(t)}\Big] \log(\pi_k) \\
&+ \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K} \Big(E\Big[Z_{nk} U_{ng} \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}^{(t)}\Big] \log(\phi_k) + E\Big[Z_{nk}(1 - U_{ng}) \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}^{(t)}\Big] \log(1 - \phi_k) \Big) \\
&+ \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K} E\Big[Z_{nk}(1 - U_{ng}) \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}^{(t)}\Big] \times \\
&\Big[- \exp\{\log(T_n) + \beta_{0g} + \rho_{gk} + \sum_{p=1}^{P} x_{np}\beta_{pg}\} + y_{ng}\{\log(T_n) + \beta_{0g} + \rho_{gk} + \sum_{p=1}^{P} \beta_{pg} x_{np}\} - \log\, y_{ng}! \Big]
\end{aligned} \tag{3.22}$$

Similarly to Equations (3.9) and (3.11) in Section 3.5.1, we calculate $\hat{Z}_{nk}^{(t)}$ and $\hat{U}_{ngk}^{(t)}$ as follows:

$$\hat{Z}_{nk}^{(t)} = E\Big[Z_{nk} \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}^{(t)}\Big] = \frac{\pi_k^{(t)} \prod_{g=1}^{G} p(y_{ng} \,|\, \phi_k^{(t)}, \rho_{gk}^{(t)}, \beta_{0g}^{(t)}, \beta_{pg}^{(t)})}{\sum_{k=1}^{K} \pi_k^{(t)} \prod_{g=1}^{G} p(y_{ng} \,|\, \phi_k^{(t)}, \rho_{gk}^{(t)}, \beta_{0g}^{(t)}, \beta_{pg}^{(t)})} \tag{3.23}$$

$$\hat{U}_{ngk}^{(t)} = p\Big(U_{ng} = 1 | Z_{nk} = 1, \boldsymbol{x}, y_{ng}, \boldsymbol{\theta}^{(t)}\Big) = \begin{cases} \frac{\phi_k^{(t)}}{\left(\phi_k^{(t)} + (1 - \phi_k^{(t)}) e^{-\lambda_{ngk}^{(t)}}\right)} & \text{if } y_{ng} = 0, \\ 0 & \text{if } y_{ng} = 1, 2, \ldots, \end{cases} \tag{3.24}$$

where $\lambda_{ngk}^{(t)} = \exp\Big(\log(T_n) + \rho_{gk}^{(t)} + \beta_{0g}^{(t)} + \sum_{p=1}^{P} \beta_{pg}^{(t)} x_{np}\Big)$.

By using the expected values $\hat{Z}_{nk}^{(t)}$ and $\hat{U}_{ngk}^{(t)}$ from Equations (3.23) and (3.24), we can rewrite $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ as follows:

$$Q(\boldsymbol{\theta}\,;\, \boldsymbol{\theta}^{(t)}) = Q_1(\boldsymbol{\pi}\,;\, \boldsymbol{\pi}^{(t)}) + Q_2(\boldsymbol{\phi}\,;\, \boldsymbol{\phi}^{(t)}) + Q_3\Big((\boldsymbol{\beta}_0, \boldsymbol{\rho}, \boldsymbol{\beta})\,;\, (\boldsymbol{\beta}_0^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)})\Big)$$

where

$$Q_1(\boldsymbol{\pi}\,;\boldsymbol{\pi}^{(t)}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}\log(\pi_k), \tag{3.25}$$

$$Q_2(\boldsymbol{\phi}\,;\boldsymbol{\phi}^{(t)}) = \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K}\left[\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}\log(\phi_k) + \hat{Z}_{nk}(1-\hat{U}_{ngk}^{(t)})\log(1-\phi_k)\right], \text{ and} \tag{3.26}$$

$$Q_3\Big((\boldsymbol{\beta}_0,\boldsymbol{\rho},\boldsymbol{\beta})\,;(\boldsymbol{\beta}_0^{(t)},\boldsymbol{\rho}^{(t)},\boldsymbol{\beta}^{(t)})\Big) = \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\times \tag{3.27}$$

$$\Big[-\exp\{\log(T_n)+\beta_{0g}+\rho_{gk}+\sum_{p=1}^{P}x_{np}\beta_{pg}\}+y_{ng}\{\log(T_n)+\beta_{0g}+\rho_{gk}+\sum_{p=1}^{P}\beta_{pg}x_{np}\}-\log\,y_{ng}!\Big]$$

M-Step: In the M-Step, through differentiating $Q(\boldsymbol{\theta}\,;\boldsymbol{\theta}^{(t)})$ with respect to each parameter, we can find the updated estimates. For the updated parameters $\pi_k^{(t+1)}$ and $\phi_k^{(t+1)}$, we can easily calculate them in a closed form similar to the scenario presented in Section 3.5.1. However, for $\beta_{0g}^{(t+1)}$, $\rho_{gk}^{(t+1)}$, and $\beta_{pg}^{(t+1)}$ there is no closed-form solution and a numerical optimization method within the M-step has to be used. Hence, the updated estimates for $\pi_k^{(t+1)}$ and $\phi_k^{(t+1)}$ are as follows:

$$\pi_k^{(t+1)} = \frac{\sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}}{N}, \text{ and} \tag{3.28}$$

$$\phi_k^{(t+1)} = \frac{\sum_{n=1}^{N}\sum_{g=1}^{G}\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}}{G\sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}}. \tag{3.29}$$

Next, we find the updated estimates $\beta_{0g}^{(t+1)}$, $\rho_{gk}^{(t+1)}$, and $\beta_{pg}^{(t+1)}$ as the values that maximize Q_3 in (3.27). As mentioned earlier, we cannot calculate a closed-form solution for these parameters. Thus, to find their new (updated) estimates we use the Fisher scoring algorithm (a form of Newton-Raphson method) as described in Section 1.4 of Chapter 1. Therefore, in what follows, we present the first derivatives of Q_3, and the negative of the second derivative expected values.

First Derivatives:

$$\frac{\partial Q_3}{\partial\beta_{pg}} = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})(y_{ng}-\lambda_{ngk})x_{np}$$

$$\frac{\partial Q_3}{\partial\rho_{gk}} = \sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})(y_{ng}-\lambda_{ngk})$$

$$\frac{\partial Q_3}{\partial \beta_{0g}} = \sum_{n=1}^{N}\sum_{k=1}^{K} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})(y_{ng}-\lambda_{ngk})$$

where $\lambda_{ngk} = \exp\Big(\log(T_n) + \rho_{gk} + \beta_{0g} + \sum_{p=1}^{P}\beta_{pg}x_{np}\Big)$.

Negative of the second derivative expected values:

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{0g}^2}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\lambda_{ngk}$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}^2}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})x_{np}^2\lambda_{ngk}$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \rho_{gk}^2}\Big] = \sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\lambda_{ngk}$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{0g}\partial \rho_{gk}}\Big] = \sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\lambda_{ngk}$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}\partial \beta_{rg}}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})x_{np}x_{nr}\lambda_{ngk}$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}\partial \beta_{0g}}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\lambda_{ngk}.x_{np}$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}\partial \rho_{gk}}\Big] = \sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\lambda_{ngk}x_{np}$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{0g}\partial \beta_{0g'}}\Big] = 0$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \rho_{gk}\partial \rho_{g'k}}\Big] = 0$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \rho_{gk}\partial \rho_{gk'}}\Big] = 0$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{0g}\partial \rho_{g'k}}\Big] = 0$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial \beta_{0g}\partial \beta_{pg'}}\Big] = 0$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial\beta_{pg}\partial\beta_{pg'}}\Big] = 0$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial\beta_{pg}\partial\rho_{g'k}}\Big] = 0$$

$$-E\Big[\frac{\partial^2 Q_3}{\partial\rho_{gk}\partial\beta_{pg'}}\Big] = 0$$

Algorithm 2 presents a summary of the EM algorithm steps for the ZIP mixture model with covariates.

Algorithm 2 EM algorithm for the ZIP mixture model with covariates

Input: $\boldsymbol{y}$: matrix of data; $\boldsymbol{\theta}^{(0)} = (\boldsymbol{\pi}^{(0)}, \boldsymbol{\phi}^{(0)}, \boldsymbol{\rho}^{(0)}, \boldsymbol{\beta}_0^{(0)}, \boldsymbol{\beta}^{(0)})$: initial parameters; tol: tolerance; m: maximum number of iterations; $\boldsymbol{x}$: matrix of covariates.
Output: optimal set of parameters $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\beta}}_0, \hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\beta}})$, and $\hat{Z}_{nk}$, $\hat{U}_{ngk}$ for all n, g, and k.

initial $t = 0$ (iteration number);
repeat
    Start E-step:
    Calculate $\hat{Z}_{nk}^{(t)}$, for all n and k, as in (3.23);
    Calculate $\hat{U}_{ngk}^{(t)}$, for all n, g, and k, as in (3.24).
    Start M-Step using the $\hat{Z}_{nk}^{(t)}$'s and $\hat{U}_{ngk}^{(t)}$'s:
    Compute $\pi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.28);
    Compute $\phi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.29);
    Compute $\rho_{gk}^{(t+1)}, \beta_{0g}^{(t+1)}, \beta_{pg}^{(t+1)}$, for all g, k, and p, using the Fisher scoring algorithm.
until $[\ell(\boldsymbol{\theta}^{(t+1)} \,|\, \boldsymbol{y}) - \ell(\boldsymbol{\theta}^{(t)} \,|\, \boldsymbol{y})] \leq tol$ or maximum number of iterations is achieved.

A simpler model than in (3.21) can be considered when there are no covariates but one wants to include a size factor T_n. Thus, we can model λ_{ngk} as

$$\log(\lambda_{ngk}) = \log(T_n) + \rho_{gk} + \beta_{0g}. \tag{3.30}$$

In this case, only a few modifications are required in the EM algorithm. In the E-step, $Q_1(\boldsymbol{\pi}; \boldsymbol{\pi}^{(t)})$ and $Q_2(\boldsymbol{\phi}; \boldsymbol{\phi}^{(t)})$ in equations (3.25) and (3.26) remain the same, but Q_3 now becomes:

$$Q_3\big((\boldsymbol{\beta}_0, \boldsymbol{\rho}); (\boldsymbol{\beta}_0^{(t)}, \boldsymbol{\rho}^{(t)})\big) = \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K} \hat{Z}_{nk}^{(t)}(1 - \hat{U}_{ngk}^{(t)}) \times$$
$$\Big[-\exp\{\log(T_n) + \beta_{0g} + \rho_{gk}\} + y_{ng}\{\log(T_n) + \beta_{0g} + \rho_{gk}\} - \log y_{ng}! \Big].$$

Therefore, the updates of π_k and ϕ_k are as in (3.28) and (3.29), respectively, and the updates of ρ_{gk} and β_{0g} can also be obtained via the Fisher scoring algorithm.

In our code implementation of the EM algorithm, to avoid identifiability issues when estimating the parameters, we assume that $\beta_{gk} = \beta_{0g} + \rho_{gk}$ with the restriction that $\sum_{k=1}^{K} \rho_{gk} = 0$. This assumption and restriction imply that $\beta_{0g} = \sum_{k=1}^{K} \beta_{gk}/K$. So, we fit our model considering $\log \lambda_{ngk} = \log(T_n) + \beta_{gk}$ (or $\log \lambda_{ngk} = \log(T_n) + \beta_{gk} + \sum_{p=1}^{P} \beta_{pg} x_{np}$), and after we obtain $\beta_{gk}^{(t+1)}$ at each EM iteration, we find the updates for β_{0g} and ρ_{gk} as follows:

$$\beta_{0g}^{(t+1)} = \frac{\sum_{k=1}^{K} \beta_{gk}^{(t+1)}}{K},$$

and

$$\rho_{gk}^{(t+1)} = \beta_{gk}^{(t+1)} - \beta_{0g}^{(t+1)}.$$

3.6 The proposed mixture model for ZINB counts

In this section, our proposed clustering approach will be based on a mixture model of zero-inflated negative binomial distributions, which pools information from observed data across all cells and neighboring genes to infer cell-specific cluster assignments and their corresponding gene expression profiles.

Similar to the cases of ZIP mixture models presented in Section 3.5, we again let Y_{ng} be a random variable for the number of read counts aligned to gene g in cell n, for $g = 1, \ldots, G$ and $n = 1, \ldots, N$, where Y_{ng} takes a value in $0, 1, 2, 3, \ldots$. Suppose that there are $K << N$ clusters of cells and let Z_{nk} be the latent Bernoulli random variable indicating the true cluster assignment of cell n as in (3.4). Therefore, given Z_{nk}, that is, given that cell n belongs to cluster k, we assume that genes are independent and follow a zero-inflated negative binomial (ZINB) distribution with parameters that depend on cluster k. Let $\boldsymbol{\mu}_k = (\mu_{11}, \ldots, \mu_{GK})^T$ and $\boldsymbol{\theta}_k = \{\pi_k, \phi_k, \alpha_k, \boldsymbol{\mu}_k\}$, where π_k is the cluster assignment probability (i.e., $P(Z_{nk} = 1) = \pi_k$), ϕ_k is the zero-inflation proportion (or probability of always zero), μ_{gk} is the rate parameter and α_k is the dispersion parameter which is the inverse of size (ν_k) parameter ($\alpha_k = \frac{1}{\nu_k}$). Thus, we define $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ as the set containing all model parameters and write the pmf for each cell as a mixture of ZINB distributions as follows:

$$p(\boldsymbol{y}_n \,|\, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{y}_n \,|\, \boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \prod_{g=1}^{G} p(y_{ng} \,|\, \phi_k, \alpha_k, \mu_{gk}),$$

where

$$p(y_{ng}\,|\,\phi_k,\alpha_k,\mu_{gk}) = \begin{cases} \phi_k + (1-\phi_k)\left(\frac{1}{1+\alpha_k\mu_{gk}}\right)^{\left(\frac{1}{\alpha_k}\right)} & \text{if } y_{ng}=0 \\ (1-\phi_k)\frac{\Gamma(y_{ng}+\frac{1}{\alpha_k})}{\Gamma(y_{ng}+1)\Gamma(\frac{1}{\alpha_k})}\left(\frac{1}{1+\alpha_k\mu_{gk}}\right)^{\left(\frac{1}{\alpha_k}\right)}\left(1-\frac{1}{1+\alpha_k\mu_{gk}}\right)^{(y_{ng})} & \text{if } y_{ng}=1,2,3,\ldots, \end{cases} \tag{3.31}$$

which can also be written as

$$p(y_{ng}\,|\,\phi_k,\alpha_k,\mu_{gk}) = \begin{cases} \phi_k & \text{if } y_{ng} \text{ belongs to always zero state} \\ (1-\phi_k)\frac{\Gamma(y_{ng}+\frac{1}{\alpha_k})}{\Gamma(y_{ng}+1)\Gamma(\frac{1}{\alpha_k})}\left(\frac{1}{1+\alpha_k\mu_{gk}}\right)^{\left(\frac{1}{\alpha_k}\right)}\left(1-\frac{1}{1+\alpha_k\mu_{gk}}\right)^{(y_{ng})} & \text{if } y_{ng} \text{ belongs to the NB state,} \end{cases} \tag{3.32}$$

for $n = 1,\ldots,N$, $g = 1,\ldots,G$, and $k = 1,\ldots,K$.

The observed-data likelihood based on all cells is given by:

$$\mathcal{L}(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n\,|\,\boldsymbol{\theta}) = \prod_{n=1}^{N}\sum_{k=1}^{K}\pi_k\prod_{g=1}^{G} p(y_{ng}\,|\,\phi_k,\alpha_k,\mu_{gk})$$

and therefore, the observed-data log-likelihood is:

$$\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = \sum_{n=1}^{N}\log\Big[\sum_{k=1}^{K}\pi_k\prod_{g=1}^{G} p(y_{ng}\,|\,\phi_k,\alpha_k,\mu_{gk})\Big]. \tag{3.33}$$

Similarly to Section 3.5, the goal is to find the parameter estimates that maximize (3.33) iteratively via the EM algorithm. To find the parameter estimates using the EM framework, we consider the latent cluster assignments $\boldsymbol{Z} = (Z_{11},\ldots,Z_{NK})^T$ and the hidden Bernoulli variable U_{ng} defined as follows:

$$U_{ng} = \begin{cases} 1 & \text{if } y_{ng} \text{ is from the perfect zero state,} \\ 0 & \text{if } y_{ng} \text{ is from the Negative Binomial (NB) state,} \end{cases}$$

with $P(U_{ng} = 1|Z_{nk} = 1) = \phi_k$, for $n = 1,\ldots,N$ and $g = 1,\ldots,G$. In the E-step, we calculate the conditional expectation of the complete-data log-likelihood given the observed data and the current parameter estimates. In the M-step, we maximize the expectation from the E-step with respect to each parameter to obtain the updated parameter estimates. Considering the observed counts and the latent random variables, we can write the completed-data log-likelihood as

follows:

$$\begin{aligned}\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{u},\boldsymbol{z}) &= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G} z_{nk}(1-u_{ng})\log\left(\frac{\Gamma(y_{ng}+\frac{1}{\alpha_k})}{\Gamma(y_{ng}+1)\Gamma(\frac{1}{\alpha_k})}(\frac{1}{1+\alpha_k\mu_{gk}})^{(\frac{1}{\alpha_k})}(1-\frac{1}{1+\alpha_k\mu_{gk}})^{(y_{ng})}\right)\\ &+\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G}\Big(z_{nk}u_{ng}\log(\phi_k)+z_{nk}(1-u_{ng})\log(1-\phi_k)\Big)\\ &+\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\log\pi_k.\end{aligned}$$

In what follows, the E and M steps of our proposed EM algorithm for the ZINB mixture model without covariates (Section 3.6.1) and with covariates (Section 3.6.2) are presented.

3.6.1 EM for the ZINB mixture model without covariates

E-Step: Given the current estimates of the parameters $\boldsymbol{\theta}^{(t)} = (\boldsymbol{\alpha}^{(t)},\boldsymbol{\mu}^{(t)},\boldsymbol{\phi}^{(t)},\boldsymbol{\pi}^{(t)})$ and the observed data $\boldsymbol{y}$, we compute the conditional expectation of the complete-data log-likelihood as:

$$\begin{aligned}Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) &= E\Big[\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{u},\boldsymbol{z})\,\Big|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G} E\Big[Z_{nk}(1-U_{ng})\,\Big|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log\left(\frac{\Gamma(y_{ng}+\frac{1}{\alpha_k})}{\Gamma(y_{ng}+1)\Gamma(\frac{1}{\alpha_k})}(\frac{1}{1+\alpha_k\mu_{gk}})^{(\frac{1}{\alpha_k})}(1-\frac{1}{1+\alpha_k\mu_{gk}})^{(y_{ng})}\right)\\ &+\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G}\left(E\Big[Z_{nk}U_{ng}\,\Big|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log(\phi_k)+E\Big[Z_{nk}(1-U_{ng})\,\Big|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log(1-\phi_k)\right)\\ &+\sum_{n=1}^{N}\sum_{k=1}^{K} E\Big[Z_{nk}\,\Big|\,\boldsymbol{y},\boldsymbol{\theta}^{(t)}\Big]\log(\pi_k).\end{aligned}\tag{3.34}$$

Using the approach described in Section 3.5.1, the expected values in (3.34) can be computed via $\hat{Z}_{nk}^{(t)}$ and $\hat{U}_{ng}^{(t)}$ given by:

$$\begin{aligned}\hat{Z}_{nk}^{(t)} &= E\Big[Z_{nk}\,|\,\boldsymbol{y}\,,\,\boldsymbol{\theta}^{(t)}\Big]\\ &= E\Big[Z_{nk}\,|\,\boldsymbol{y}_n\,,\,\boldsymbol{\theta}^{(t)}\Big]\\ &= p\Big(Z_{nk}=1\,|\,\boldsymbol{y}_n\,,\,\boldsymbol{\theta}^{(t)}\Big)\\ &= \frac{\pi_k^{(t)}\prod_{g=1}^{G} p\Big(y_{ng}\,|\,\alpha_k^{(t)},\mu_{gk}^{(t)},\phi_k^{(t)}\Big)}{\sum_{j=1}^{K}\pi_j^{(t)}\prod_{g=1}^{G} p\Big(y_{ng}\,|\,\alpha_j^{(t)},\mu_{gj}^{(t)},\phi_j^{(t)}\Big)},\end{aligned}\tag{3.35}$$

and

$$\begin{aligned}\hat{U}_{ngk}^{(t)} &= p(U_{ng}=1 \,|\, Z_{nk}=1, y_{ng}, \boldsymbol{\theta}^{(t)}) \\ &= \begin{cases} \dfrac{\phi_k^{(t)}}{\left(\phi_k^{(t)} + (1-\phi_k^{(t)})\left(\frac{1}{1+\alpha_k^{(t)}\mu_{gk}^{(t)}}\right)^{\frac{1}{\alpha_k^{(t)}}}\right)} & \text{if } y_{ng}=0 \\ 0 & \text{if } y_{ng}=1,2,\ldots. \end{cases}\end{aligned} \tag{3.36}$$

From (3.36) we note again that $\hat{U}_{ngk}^{(t)}$ is separated into two cases, if $U_{ng}=1$, y_{ng} can only be equal to zero, and $U_{ng}=0$, if y_{ng} takes a non-zero count value, it definitely arises from the negative binomial state. Thus, using $\hat{Z}_{nk}^{(t)}$ and $\hat{U}_{ngk}^{(t)}$ from Equations (3.35) and (3.36), we can rewrite (3.34) as:

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) = Q_1(\boldsymbol{\pi};\boldsymbol{\pi}^{(t)}) + Q_2(\boldsymbol{\phi};\boldsymbol{\phi}^{(t)}) + Q_3((\boldsymbol{\mu},\boldsymbol{\alpha});(\boldsymbol{\mu}^{(t)},\boldsymbol{\alpha}^{(t)})), \tag{3.37}$$

where

$$Q_1(\boldsymbol{\pi}\,;\boldsymbol{\pi}^{(t)}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}\log(\pi_k),$$

$$Q_2(\boldsymbol{\phi}\,;\boldsymbol{\phi}^{(t)}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G}\left[\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}\log(\phi_k) + \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\log(1-\phi_k)\right], \text{ and}$$

$$Q_3\Big((\boldsymbol{\mu}\,,\boldsymbol{\alpha});(\boldsymbol{\mu}^{(t)},\boldsymbol{\alpha}^{(t)})\Big) = \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G}\left[\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\left\{y_{ng}\log\left(\frac{\alpha_k\mu_{gk}}{1+\alpha_k\mu_{gk}}\right) - \frac{1}{\alpha_k}\log(1+\alpha_k\mu_{gk})\right.\right.$$

$$\left.\left. + \log\Gamma(y_{ng}+\frac{1}{\alpha_k}) - \log\Gamma(y_{ng}+1) - \log\Gamma(\frac{1}{\alpha_k})\right\}\right].$$

M-step: In this step, we find the updated parameters $\boldsymbol{\theta}^{(t+1)} = (\boldsymbol{\alpha}^{(t+1)},\boldsymbol{\mu}^{(t+1)},\boldsymbol{\phi}^{(t+1)},\boldsymbol{\pi}^{(t+1)})$ that maximize $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)})$ given in Equation (3.37).

For calculating the updated estimate of π_k, similar to Section (3.5.1), we use the Lagrange multiplier (see Equations 3.13 to 3.16) and obtain:

$$\pi_k^{(t+1)} = \frac{\sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}}{N}. \tag{3.38}$$

The updated $\phi_k^{(t+1)}$ is as in (3.18); that is:

$$\phi_k^{(t+1)} = \frac{\sum_{n=1}^{N}\sum_{g=1}^{G}\hat{Z}_{nk}^{(t)}\hat{U}_{ngk}^{(t)}}{G\sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}}. \tag{3.39}$$

We obtain the updated $\mu_{gk}^{(t+1)}$ as follows:

$$\frac{\partial Q_3\Big((\boldsymbol{\mu},\boldsymbol{\alpha})\,;\,(\boldsymbol{\mu}^{(t)},\boldsymbol{\alpha}^{(t)})\Big)}{\partial \mu_{gk}} = \sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{y_{ng}-\mu_{gk}}{\mu_{gk}(1+\alpha_k\mu_{gk})} = 0,$$

$$\Longrightarrow \mu_{gk}^{(t+1)} = \frac{\sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})y_{ng}}{\sum_{n=1}^{N} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})}. \tag{3.40}$$

Finally, to reach the updated $\alpha_k^{(t+1)}$, we consider the expectation-conditional maximization (ECM) algorithm by fixing μ_{gk} at $\mu_{gk}^{(t+1)}$ and obtaining $\alpha_k^{(t+1)}$ as the solution of the following equation:

$$\frac{\partial\, Q_3\Big((\boldsymbol{\mu},\boldsymbol{\alpha});\,(\boldsymbol{\mu^{(t)}},\boldsymbol{\alpha}^{(t)})\Big)}{\partial\, \alpha_k} = \sum_{n=1}^{N}\sum_{g=1}^{G} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\left[\frac{1}{\alpha_k^2}\left(\ln(1+\alpha_k\mu_{gk}^{(t+1)}) + \frac{\alpha_k(y_{ng}-\mu_{gk}^{(t+1)})}{(1+\alpha_k\mu_{gk}^{(t+1)})} + \psi(y_{ng}+\frac{1}{\alpha_k}) - \psi(\frac{1}{\alpha_k})\right)\right] = 0, \tag{3.41}$$

where $\psi(\cdot)$ is the so-called digamma function, which is defined as the derivative of the natural logarithm of Γ; that is, $\psi(x) = \dfrac{d}{dx}\log\Gamma(x)$ (Hilbe, 2011). However, there is no closed-form solution for (3.41); therefore, a numerical optimization algorithm, such as Newton-Raphson or Fisher scoring, must be applied. To facilitate computation, as the dispersion parameter (α) is the inverse of the size parameter (ν) in the negative binomial distribution, we consider this alternate form of the negative binomial using ν and rewrite Q_3 as follows:

$$Q_3\Big((\boldsymbol{\mu}\,,\,\boldsymbol{\alpha});(\boldsymbol{\mu}^{(t)},\boldsymbol{\alpha}^{(t)})\Big) = \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{g=1}^{G}\left[\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\left\{y_{ng}\,\log\left(\frac{\frac{\mu_{gk}}{\nu_k}}{1+\frac{\mu_{gk}}{\nu_k}}\right) - \nu_k\log(1+\tfrac{\mu_{ng}}{\nu_k}) + \log\Gamma(y_{ng}+\nu_k) - \log\Gamma(y_{ng}+1) - \log\Gamma(\nu_k)\right\}\right]. \tag{3.42}$$

Now, fixing μ_{gk} at $\mu_{gk}^{(t+1)}$, the first derivative and the negative of the second derivative of Q_3 w.r.t. ν_k are:

$$\frac{\partial\, Q_3}{\partial\, \nu_k} = \sum_{n=1}^{N}\sum_{g=1}^{G} \hat{Z}_{nk}^{(t)}(\hat{U}_{ngk}^{(t)})\times \left[\psi(y_{ng}+\nu_k) - \psi(\nu_k) + \log\nu_k + 1 - \log(\nu_k+\mu_{gk}^{(t+1)}) - \frac{y_{ng}+\nu_k}{\nu_k+\mu_{gk}^{(t+1)}}\right], \tag{3.43}$$

and

$$-\frac{\partial^2 Q_3}{\partial \nu_k^2} = \sum_{n=1}^{N}\sum_{g=1}^{G} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\times \left[-\psi'(y_{ng}+\nu_k) + \psi'(\nu_k) - \frac{1}{\nu_k} + \frac{2}{\nu_k+\mu_{gk}^{(t+1)}} - \frac{y_{ng}+\nu_k}{(\nu_k+\mu_{gk}^{(t+1)})^2} \right], \tag{3.44}$$

where $\psi'(\cdot)$ is the trigamma function defined as the second derivative of the natural logarithm of the gamma function (Hilbe, 2011). Thus, we find the updated $\nu_k^{(t+1)}$ (and, subsequently, $\alpha_k^{(t+1)} = 1/\nu_k^{(t+1)}$) using the first and second derivatives of Q_3 as in (3.43) and (3.44) and the Newton-Raphson algorithm implemented in the function *theta.ml* from the R package *MASS*. Note that, when using *theta.ml* in R, we enter the data in a vector format rather than a matrix, and we use the $\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})$'s as weights also in a vector format and the $\mu_{gk}^{(t+1)}$'s as the rate parameters.

Algorithm 3 shows the ECM algorithm to obtain the updated parameter estimates for the ZINB mixture model without covariates.

Algorithm 3 ECM algorithm for the ZINB mixture model without covariates

Input: y: Matrix of Data; $\boldsymbol{\theta}^{(0)} = (\boldsymbol{\pi}^{(0)}, \boldsymbol{\phi}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\nu}^{(0)})$: initial parameters; *tol*: tolerance; m: maximum number of iterations;

Output: optimal set of parameters $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\nu}})$, and $\hat{Z}_{nk}$, $\hat{U}_{ngk}$, for all n, g and k.

1: initial $t = 0$ (iteration number);
2: **repeat**
3: Start E-step;
4: Calculate $\hat{Z}_{nk}^{(t)}$, for all n and k, as in (3.35);
5: Calculate $\hat{U}_{ngk}^{(t)}$, for all n, g, and k, as in (3.36);
6: Start M-Step using the $\hat{Z}_{nk}^{(t)}$'s and $\hat{U}_{ngk}^{(t)}$'s;
7: Compute $\pi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.38);
8: Compute $\phi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.39);
9: Compute $\mu_{gk}^{(t+1)}$, for $k = 1, \ldots, K$ and $g = 1, \ldots, G$ as in (3.40);
10: Fix μ_{gk} at $\mu_{gk}^{(t+1)}$, compute $\nu_k^{(t+1)}$, for $k = 1, \ldots, K$, using the Newton-Raphson algorithm via the *theta.ml* function in R;
11: **until** $[\ell(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{y}) - \ell(\boldsymbol{\theta}^{(t)}|\boldsymbol{y})] \leq tol$ or maximum number of iterations is achieved.

3.6.2 EM for the ZINB mixture model with covariates

In the case of a ZINB mixture model with covariates, we assume that the log-link function for the rate parameters is as in Eq. (3.21) in Section 3.5.2; that is:

$$\log(\mu_{ngk}) = \log(T_n) + \rho_{gk} + \beta_{0g} + \sum_{p=1}^{P} \beta_{pg} x_{np}, \tag{3.45}$$

for $n = 1,\ldots,N$, $g = 1,\ldots,G$, $k = 1,\ldots,K$, and $p = 1,\ldots,P$, where T_n is a fixed size factor for cell n, β_{0g} is a baseline expression for gene g, ρ_{gk} is the fixed effect of cluster k on gene g, $x_{n1},\ldots,x_{np}$ are P known covariates for cell n, and $\beta_{1g},\ldots,\beta_{Pg}$ their corresponding unknown coefficients.

Let $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\alpha}, \boldsymbol{\rho}, \boldsymbol{\beta}_0, \boldsymbol{\beta})$, where $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_k)^T$, $\boldsymbol{\phi} = (\phi_1,\ldots,\phi_k)^T$, $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_k)^T$, $\boldsymbol{\rho} = (\rho_{11},\ldots,\rho_{G1},\ldots,\rho_{1K},\ldots,\rho_{GK})^T$, $\boldsymbol{\beta}_0 = (\beta_{01},\ldots,\beta_{0G})^T$, and $\boldsymbol{\beta} = (\beta_{11},\ldots,\beta_{p1},\ldots,\beta_{1G},\ldots,\beta_{PG})^T$. In what follows, we describe how we can find the estimates of the parameters in $\boldsymbol{\theta}$ using the EM algorithm. In this case, the complete-data log-likelihood can be written as:

$$\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{x},z,\boldsymbol{u}) = \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K}\Bigg[z_{nk}\log(\pi_k) + z_{nk}u_{ng}\log(\phi_k) + z_{nk}(1-u_{ng})\,\log(1-\phi_k) + z_{nk}(1-u_{ng})\,\log(\,p(y_{ng}\,|\,\rho_{gk},\beta_{0g},\beta_{pg},\alpha_k))\Bigg].$$

E-Step: The conditional expectation of the complete-data log-likelihood given the observed data and current parameter estimates is as follows:

$$\begin{aligned}
Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(t)}) &= E\Big[\ell(\boldsymbol{\theta}\,|\,\boldsymbol{y},\boldsymbol{x},z,\boldsymbol{u})\,\Big|\,\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}^{(t)}\Big] = \\
&\sum_{n=1}^{N}\sum_{k=1}^{K} E\Big[Z_{nk}\,|\,\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}^{(t)}\Big]\log(\pi_k) \\
&+ \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K}\Big(E\Big[Z_{nk}U_{ng}\,|\,\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}^{(t)}\Big]\log(\phi_k) + E\Big[Z_{nk}(1-U_{ng})\,|\,\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}^{(t)}\Big]\log(1-\phi_k)\Big) \\
&+ \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K} E\Big[Z_{nk}(1-U_{ng})\,|\,\boldsymbol{y},\boldsymbol{x},\boldsymbol{\theta}^{(t)}\Big]\times \\
&\log\Bigg\{\frac{\Gamma(y_{ng}+\frac{1}{\alpha_k})}{\Gamma(y_{ng}+1)\Gamma(\frac{1}{\alpha_k})}\times\Big(\frac{1}{1+\alpha_k\exp(\log(T_n)+\rho_{gk}+\beta_{0g}+\sum_{p=1}^{P}\beta_{pg}x_{np})}\Big)^{\frac{1}{\alpha_k}} \\
&\times\Big(1-\frac{1}{1+\alpha_k\exp(\log(T_n)+\rho_{gk}+\beta_{0g}+\sum_{p=1}^{P}\beta_{pg}x_{np})}\Big)^{y_{ng}}\Bigg\}
\end{aligned} \tag{3.46}$$

As described in previous sections, we can calculate the expectations in (3.46) using $\hat{Z}_{nk}$ and $\hat{U}_{ngk}$ given by:

$$\hat{Z}_{nk}^{(t)} = E\Big[Z_{nk} \,|\, \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\theta}^{(t)}\Big] = \frac{\pi_k^{(t)} \prod_{g=1}^{G} p(y_{ng} \,|\, \phi_k^{(t)}, \alpha_k^{(t)}, \rho_{gk}^{(t)}, \beta_{0g}^{(t)}, \beta_{pg}^{(t)})}{\sum_{k=1}^{K} \pi_k^{(t)} \prod_{g=1}^{G} p(y_{ng} \,|\, \phi_k^{(t)}, \alpha_k^{(t)}, \rho_{gk}^{(t)}, \beta_{0g}^{(t)}, \beta_{pg}^{(t)})}, \tag{3.47}$$

and

$$\begin{aligned} \hat{U}_{ngk}^{(t)} &= p(U_{ng} = 1 | Z_{nk} = 1, \boldsymbol{x}, y_{ng}, \boldsymbol{\theta}^{(t)}) \\ &= \begin{cases} \dfrac{\phi_k^{(t)}}{\left(\phi_k^{(t)} + (1-\phi_k^{(t)})\left(\frac{1}{1+\alpha_k^{(t)}\mu_{ngk}^{(t)}}\right)^{\frac{1}{\alpha_k^{(t)}}}\right)} & \text{if } y_{ng} = 0, \\ 0 & \text{if } y_{ng} = 1, 2, \ldots, \end{cases} \end{aligned} \tag{3.48}$$

where $\mu_{ngk}^{(t)} = \exp\Big(\log(T_n) + \rho_{gk}^{(t)} + \beta_{0g}^{(t)} + \sum_{p=1}^{P} \beta_{pg}^{(t)} x_{np}\Big)$.

Using $\hat{Z}_{nk}$ and $\hat{U}_{ngk}$ as in (3.47) and (3.48), respectively, we can rewrite $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$ as follows:

$$Q(\boldsymbol{\theta}\,;\, \boldsymbol{\theta}^{(t)}) = Q_1(\boldsymbol{\pi}\,;\, \boldsymbol{\pi}^{(t)}) + Q_2(\boldsymbol{\phi}\,;\, \boldsymbol{\phi}^{(t)}) + Q_3((\boldsymbol{\beta}_0, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\alpha})\,;\, (\boldsymbol{\beta}_0^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\alpha}^{(t)})), \tag{3.49}$$

where

$$Q_1(\boldsymbol{\pi}\,;\, \boldsymbol{\pi}^{(t)}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{Z}_{nk}^{(t)} \log(\pi_k), \tag{3.50}$$

$$Q_2(\boldsymbol{\phi}\,;\, \boldsymbol{\phi}^{(t)}) = \sum_{n=1}^{N} \sum_{g=1}^{G} \sum_{k=1}^{K} \Big[\hat{Z}_{nk}^{(t)} \hat{U}_{ngk}^{(t)} \log(\phi_k) + \hat{Z}_{nk}(1 - \hat{U}_{ngk}^{(t)}) \log(1 - \phi_k)\Big], \tag{3.51}$$

and

$$\begin{aligned} &Q_3\Big((\boldsymbol{\alpha}, \boldsymbol{\rho}, \boldsymbol{\beta}_0, \boldsymbol{\beta}); (\boldsymbol{\alpha}^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}_0^{(t)}, \boldsymbol{\beta}^{(t)}\Big) = \\ &\sum_{n=1}^{N} \sum_{g=1}^{G} \sum_{k=1}^{K} \hat{Z}_{nk}^{(t)} (1 - \hat{U}_{ngk}^{(t)}) \times \\ &\log\Bigg\{ \frac{\Gamma(y_{ng} + \frac{1}{\alpha_k})}{\Gamma(y_{ng}+1)\Gamma(\frac{1}{\alpha_k})} \times \Big(\frac{1}{1 + \alpha_k \exp(\log(T_n) + \rho_{gk} + \beta_{0g} + \sum_{p=1}^{P} \beta_{pg} x_{np})}\Big)^{\frac{1}{\alpha_k}} \\ &\times \Big(1 - \frac{1}{1 + \alpha_k \exp(\log(T_n) + \rho_{gk} + \beta_{0g} + \sum_{p=1}^{P} \beta_{pg} x_{np})}\Big)^{y_{ng}} \Bigg\}. \end{aligned} \tag{3.52}$$

M-Step: In this step, we maximize $Q(\boldsymbol{\theta}\,;\, \boldsymbol{\theta}^{(t)})$ in (3.49) with respect to each parameter in $\boldsymbol{\theta}$ to find the updated parameter estimates. The updated estimates of π_k and ϕ_k can be obtained

in closed form, as shown in Equations (3.38) and (3.39). However, no closed-form solution exists for α_k, β_{0g}, ρ_{gk}, and β_{gp}; therefore, we apply the ECM algorithm along with the Newton-Raphson optimization method to find $\alpha_k^{(t+1)}$, $\beta_{0g}^{(t+1)}$, $\rho_{gk}^{(t+1)}$, and $\beta_{gp}^{(t+1)}$. Details are presented as follows.

To find the updated $\beta_{0g}^{(t+1)}$, $\rho_{gk}^{(t+1)}$, and $\beta_{gp}^{(t+1)}$ we apply the ECM algorithm by fixing α_k at its current value $\alpha_k^{(t)}$ and maximize Q_3 in (3.52) w.r.t. β_{0g}, ρ_{gk}, and β_{gp} using the Newton-Raphson algorithm. Thus, with $\mu_{ngk} = \exp\big(\log(T_n) + \rho_{gk} + \beta_{0g} + \sum_{p=1}^{P}\beta_{pg}x_{np}\big)$, the first derivatives and negative second derivatives of Q_3 w.r.t. these parameters are as the following:

First derivatives:

$$\frac{\partial Q_3}{\partial \beta_{0g}} = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\Big(\frac{y_{ng}-\mu_{ngk}}{1+\alpha_k^{(t)}\mu_{ngk}}\Big)$$

$$\frac{\partial Q_3}{\partial \beta_{pg}} = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\Big(\frac{x_{np}(y_{ng}-\mu_{ngk})}{1+\alpha_k^{(t)}\mu_{ngk}}\Big)$$

$$\frac{\partial Q_3}{\partial \rho_{gk}} = \sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\Big(\frac{y_{ng}-\mu_{ngk}}{1+\alpha_k^{(t)}\mu_{ngk}}\Big)$$

Negative of the second derivatives:

$$-\Big[\frac{\partial^2 Q_3}{\partial \beta_{0g}^2}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{\mu_{ngk}(1+\alpha_k^{(t)}y_{ng})}{(1+\alpha_k^{(t)}\mu_{ngk})^2}$$

$$-\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}^2}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{x_{np}^2\mu_{ngk}(1+\alpha_k^{(t)}y_{ng})}{(1+\alpha_k^{(t)}\mu_{ngk})^2}$$

$$-\Big[\frac{\partial^2 Q_3}{\partial \rho_{gk}^2}\Big] = \sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{\mu_{ngk}(1+\alpha_k^{(t)}y_{ng})}{(1+\alpha_k^{(t)}\mu_{ngk})^2}$$

$$-\Big[\frac{\partial^2 Q_3}{\partial \beta_{0g}\partial \rho_{gk}}\Big] = \sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{\mu_{ngk}(1+\alpha_k^{(t)}y_{ng})}{(1+\alpha_k^{(t)}\mu_{ngk})^2}$$

$$-\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}\partial \beta_{rg}}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{x_{np}x_{nr}\mu_{ngk}(1+\alpha_k^{(t)}y_{ng})}{(1+\alpha_k^{(t)}\mu_{ngk})^2}$$

$$-\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}\partial \beta_{0g}}\Big] = \sum_{n=1}^{N}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{x_{np}\mu_{ngk}(1+\alpha_k^{(t)}y_{ng})}{(1+\alpha_k^{(t)}\mu_{ngk})^2}$$

$$-\Big[\frac{\partial^2 Q_3}{\partial \beta_{pg}\partial \rho_{gk}}\Big] = \sum_{n=1}^{N}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\frac{x_{np}\mu_{ngk}(1+\alpha_k^{(t)}y_{ng})}{(1+\alpha_k^{(t)}\mu_{ngk})^2}$$

$$-\left[\frac{\partial^2 Q_3}{\partial\beta_{0g}\partial\beta_{0g'}}\right] = 0$$

$$-\left[\frac{\partial^2 Q_3}{\partial\rho_{gk}\partial\rho_{g'k}}\right] = 0$$

$$-\left[\frac{\partial^2 Q_3}{\partial\rho_{gk}\partial\rho_{gk'}}\right] = 0$$

$$-\left[\frac{\partial^2 Q_3}{\partial\beta_{0g}\partial\rho_{g'k}}\right] = 0$$

$$-\left[\frac{\partial^2 Q_3}{\partial\beta_{0g}\partial\beta_{pg'}}\right] = 0$$

$$-\left[\frac{\partial^2 Q_3}{\partial\beta_{pg}\partial\beta_{pg'}}\right] = 0$$

$$-\left[\frac{\partial^2 Q_3}{\partial\beta_{pg}\partial\rho_{g'k}}\right] = 0$$

$$-\left[\frac{\partial^2 Q_3}{\partial\rho_{gk}\partial\beta_{pg'}}\right] = 0$$

Now, similarly to Section 3.6.1, to find the updated estimate of α_k, we consider Q_3 based on the alternate form of the negative binomial with size parameter ν_k (as $\alpha_k = 1/\nu_k$). Moreover, we use the ECM algorithm by fixing β_{0g}, ρ_{gk}, and β_{gp} at their updated values $\beta_{0g}^{(t+1)}$, $\rho_{gk}^{(t+1)}$, and $\beta_{gp}^{(t+1)}$ and obtain $\nu_k^{(t+1)}$ (and, subsequently, $\alpha_k^{(t+1)}$) using the Newton-Raphson algorithm implemented by the function *theta.ml* from the library *MASS* in R. Analogously to Equations (3.43) and (3.44), the first derivative and negative second derivative of Q_3 are as follows:

$$\frac{\partial Q_3}{\partial \nu_k} = \sum_{n=1}^{N}\sum_{g=1}^{G}\hat{Z}_{nk}^{(t)}(\hat{U}_{ngk}^{(t)})$$
$$\left[\psi(y_{ng}+\nu_k) - \psi(\nu_k) + \log\nu_k + 1 - \log(\nu_k+\mu_{ngk}) - \frac{y_{ng}+\nu_k}{\nu_k+\mu_{ngk}^{(t+1)}}\right], \tag{3.53}$$

and

$$-\frac{\partial^2 Q_3}{\partial \nu_k^2} = \sum_{n=1}^{N}\sum_{g=1}^{G} \hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})\times \left[-\psi'(y_{ng}+\nu_k) + \psi'(\nu_k) - \frac{1}{\nu_k} + \frac{2}{\mu_{ngk}^{(t+1)}+\nu_k} - \frac{y_{ng}+\nu_k}{(\nu_k+\mu_{ngk}^{(t+1)})^2}\right], \tag{3.54}$$

where $\mu_{ngk}^{(t+1)} = \exp\Big(\log(T_n) + \rho_{gk}^{(t+1)} + \beta_{0g}^{(t+1)} + \sum_{p=1}^{P}\beta_{pg}^{(t+1)}x_{np}\Big)$, and $\psi(\cdot)$ and $\psi'(\cdot)$ are the digamma and trigamma functions, respectively, defined previously in Section 3.6.1.

We summarize the ECM algorithm to obtain the updated parameter estimates for the ZINB mixture model with covariates in Algorithm 4.

Algorithm 4 ECM algorithm for the ZINB mixture model with covariates

Input: $\boldsymbol{y}$: matrix of data; $\boldsymbol{\theta}^{(0)} = (\boldsymbol{\pi}^{(0)}, \boldsymbol{\phi}^{(0)}, \boldsymbol{\nu}^{(0)}, \boldsymbol{\beta}_0^{(0)}, \boldsymbol{\rho}^{(0)}, \boldsymbol{\beta}^{(0)})$: initial parameters; x_{np}: matrix of covariates; *tol*: tolerance; m: maximum number of iterations;

Output: optimal set of parameters $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\beta}}_0, \hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\nu}})$, and $\hat{Z}_{nk}$, $\hat{U}_{ngk}$, for all n, g and k.

1: initial $t = 0$ (iteration number);
2: **repeat**
3: Start E-step;
4: Calculate $\hat{Z}_{nk}^{(t)}$, for all n and k, as in (3.47);
5: Calculate $\hat{U}_{ngk}^{(t)}$, for all n, g, and k, as in (3.48);
6: Start M-Step using the $\hat{Z}_{nk}^{(t)}$'s and $\hat{U}_{ngk}^{(t)}$'s;
7: Compute $\pi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.38);
8: Compute $\phi_k^{(t+1)}$, for $k = 1, \ldots, K$, as in (3.39);
9: Fix α_k at $\alpha_k^{(t)}$, compute $\beta_{0g}^{(t+1)}$, $\rho_{gk}^{(t+1)}$, and $\beta_{pg}^{(t+1)}$, for all g, k and p using the Newton-Raphson algorithm;
10: Fix β_{0g}, ρ_{gk}, and β_{pg} at $\beta_{0g}^{(t+1)}$, $\rho_{gk}^{(t+1)}$, and $\beta_{pg}^{(t+1)}$, compute $\nu^{(t+1)}$ using the Newton-Raphson algorithm via the *theta.ml* function in R. Let $\alpha_k^{(t+1)} = 1/\nu_k^{(t+1)}$.
11: **until** $[\ell(\boldsymbol{\theta}^{(t+1)}\,|\,\boldsymbol{y}) - \ell(\boldsymbol{\theta}^{(t)}\,|\,\boldsymbol{y})] \leq tol$ or maximum number of iterations is achieved.

A simpler model than in (3.45) can be considered when there are no covariates, that is, we can model μ_{ngk} as:

$$\log(\mu_{ngk}) = \log(T_n) + \rho_{gk} + \beta_{0g}. \tag{3.55}$$

In this case, in the E-step, Q_1 and Q_2 in (3.50) and (3.51) remain the same, but Q_3 now becomes:

$$Q_3\Big((\boldsymbol{\alpha},\boldsymbol{\beta}_0,\boldsymbol{\rho})\,;\,(\boldsymbol{\alpha}^{(t)},\boldsymbol{\beta}_0^{(t)},\boldsymbol{\rho}^{(t)})\Big) = \sum_{n=1}^{N}\sum_{g=1}^{G}\sum_{k=1}^{K}\hat{Z}_{nk}^{(t)}(1-\hat{U}_{ngk}^{(t)})$$
$$\log\left\{\frac{\Gamma(y_{ng}+\frac{1}{\alpha_k})}{\Gamma(y_{ng}+1)\Gamma(\frac{1}{\alpha_k})}\times\left(\frac{1}{1+\alpha_k\times\exp(\log(T_n)+\rho_{gk}+\beta_{0g})}\right)^{\frac{1}{\alpha_k}}\right.$$
$$\left.\times\left(1-\frac{1}{1+\alpha_k\times\exp(\log(T_n)+\rho_{gk}+\beta_{0g})}\right)^{y_{ng}}\right\}. \tag{3.56}$$

The updated estimates of π_k and ϕ_k are as in (3.38) and (3.39), respectively, and the updated estimates of β_{0g}, ρ_{gk}, and α_k (or ν_k) can also be obtained via Newton-Raphson within the ECM algorithm.

Similarly to the ZIP case, in the code implementation, to also avoid identifiability issues when estimating the parameters of the ZINB mixture model with covariates, one can consider $\beta_{gk} = \beta_{0g} + \rho_{gk}$ with the restriction that $\sum_{k=1}^{K}\rho_{gk} = 0$. For more details, see the end of Section 3.5.2.

Chapter 4

Simulation Results for the Mixture of Zero-Inflated Poisson and Negative-Binomial Models

In this chapter, for each model introduced in Chapter 3, we conduct simulation studies to assess the performance of our proposed EM algorithm under various scenarios by varying different parameters and hyperparameters. For the hyperparameters, we vary the number of cells (N) (the number of rows in the matrix in Eq. 3.3), the number of genes (G) (the number of columns in the matrix), and the number of clusters (K). We note that the number of clusters is fixed in the EM algorithm. However, the optimal K can be found using a criterion such as the Aikaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), or the Integrated Complete Likelihood (ICL) (McLachlan and Krishnan, 2008). For the parameters, we consider different values of $\pi_1, \ldots, \pi_K$, the cluster assignment probabilities, $\phi_1, \ldots, \phi_K$, the probabilities of always zero, and $\nu_1, \ldots, \nu_K$, the size parameters for the case of a mixture of ZINB distributions. We also examine different values for the rate parameters for the cases with and without covariates for the mixture of ZIP and ZINB distributions.

The simulations for the ZIP mixture model without covariates and the ZIP mixture model with a size factor (i.e., with only β_{0g} and ρ_{gk}) were performed using Sharcnet's Graham computer cluster via the Digital Research Alliance of Canada, with a single node consisting of two Intel E5-2683 V4 (Broadwell) with 2.1 GHZ, for an overall of 32 computing cores. The number of available cores informs the decision to choose $S = 256$ simulated datasets, a multiple of the number of cores. The computations were performed on redCent OS 7, with R version 4.1.2 (R core Team, 2021), using the packages *Parallel* to simulate and to compute the EM algorithm of independent datasets simultaneously, the *FDRSEG* (Hi et al. (2017)) package to calculate the V-measure. Simulations for the ZINB case were performed on ASUSTEK Zen-

Book UX535LI, Intel Core i7-10750H @2.60 GHZ with R version 4.2.2, for the case without covariates. We simulated $S = 100$ datasets for each scenario and similar to the mixture of ZIP cases, the library *FDRSEG* was used for calculating V-measures.

4.1 Performance Metrics

In what follows, the different metrics and plots used to assess the performance of the proposed EM algorithm are introduced. For evaluating the performance regarding the parameters $\pi_1, \ldots, \pi_k$ and $\phi_1, \ldots, \phi_k$, the means and standard deviations of the obtained EM estimates along with boxplots are computed across the different simulation scenarios. To evaluate the performance regarding the estimation of the rate parameters λ_{gk}'s or μ_{gk}'s (case without covariates) and β_{0g}'s, β_{pg}'s, and ρ_{gk}'s (case with covariates), the mean squared error (MSE) or the median absolute deviation (MAD) are applied. For the size parameters in the ZINB case, we present boxplots, means, and standard deviations of the EM estimates. The V-measure is used to evaluate the clustering performance, i.e., how well the clustering performs compared to the true assigned clusters of each data set.

Mean Squared error (MSE): The number of rate parameters (λ_{gk}'s or μ_{gk}'s in the case without covariates, and β_{0g}'s, β_{pg}'s, and ρ_{gk}'s for the cases with covariates) vary and can increase to a high number according to some settings such as the number of genes (G) in a simulated dataset. Therefore, we calculate the overall or cluster-specific mean squared error as follows:

MSE for the rate parameters for a mixture of ZIP without covariates:

$$\text{MSE}_k = \frac{1}{SG}\sum_{s=1}^{S}\sum_{g=1}^{G}(\lambda_{kg} - \hat{\lambda}_{kg}^{(s)})^2$$

MSE for the rate parameters for a mixture of ZINB without covariates:

$$\text{MSE}_k = \frac{1}{SG}\sum_{s=1}^{S}\sum_{g=1}^{G}(\mu_{kg} - \hat{\mu}_{kg}^{(s)})^2$$

MSE for the rate parameters for a mixture of ZIP or ZINB with covariates:

$$\text{MSE}_k = \frac{1}{SG}\sum_{s=1}^{S}\sum_{g=1}^{G}(\rho_{kg} - \hat{\rho}_{kg}^{(s)})^2$$

$$\text{MSE} = \frac{1}{SG}\sum_{s=1}^{S}\sum_{g=1}^{G}(\beta_{0g} - \hat{\beta}_{0g}^{(s)})^2$$

$$\text{MSE}_p = \frac{1}{SG}\sum_{s=1}^{S}\sum_{g=1}^{G}(\beta_{pg} - \hat{\beta}_{pg}^{(s)})^2$$

Median Absolute Deviation (MAD): This is another metric to measure the estimation error related to the rate parameters that we use in the presence of outliers. We first calculate the median absolute error (deviation) over the genes (G) in each simulated dataset s. Then, we obtain the median of those errors over all the simulated sets (S).

V-measure: The V-measure (Rosenberg and Hirschberg, 2007) is a metric to assess how well a set of clusters $K = k_1, \ldots, k_m$ partition a set of N observations knowing the fact that they belong to a set of classes $C = c_1, \ldots, c_n$. This is an entropy-based metric to evaluate the performance of the clustering using the criteria of homogeneity and completeness. The homogeneity represents how similar the elements of the clusters are to the other elements of the same cluster, as measured by:

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{otherwise,} \end{cases}$$

where

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N}\log\Big(\frac{a_{ck}}{\sum_{c=1}^{|C|}a_{ck}}\Big), \text{ and}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}a_{ck}}{n}\log\Big(\frac{\sum_{k=1}^{|K|}a_{ck}}{n}\Big),$$

with a_{ck} as the number of observations from class c assigned to cluster k. The completeness represents how close the elements of the same class are clustered together, and is measured by:

$$c = \begin{cases} 1 & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{otherwise,} \end{cases}$$

where

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{N}\log\Big(\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}}\Big), \text{ and}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}a_{ck}}{n}\log\Big(\frac{\sum_{c=1}^{|C|}a_{ck}}{n}\Big).$$

With these two measures, h and c, the overall V-measure is obtained as follows:

$$V_\beta = \frac{(1+\beta)\times h \times c}{(\beta \times h) + c},$$

where we use the default $\beta = 1$ in our calculations.

4.2 Simulation results for the ZIP mixture model without covariates

We study six different simulation scenarios for the case of ZIP without covariates presented in Section 3.5.1. In each scenario, we vary one of the model parameters or hyperparameters while holding the others fixed. In Scenarios 1 and 2, we vary the number of cells (N) and the number of genes (G), respectively. In Scenario 3, we vary the number of clusters (K). In Scenario 4, we study the effect of changing the cluster assignment probabilities (π_k's) from balanced to unbalanced cases. In Scenario 5, we vary the similarities among clusters by changing the λ_{gk}'s. Finally, the effect of changes in the probabilities of always zero, ϕ_k's, is studied in Scenario 6.

Table 4.1 summarizes each of the proposed simulation scenarios for the case of a zero-inflated Poisson mixture model without covariates, and it shows which parameters and hyperparameters vary in each scenario (see the $\star$ symbol) along with the ones that we keep fixed. The number of simulated datasets in each scenario is $S = 256$. We apply the proposed EM algorithm for a ZIP mixture model without covariates (Algorithm 1 in Section 3.5.1) to each simulated dataset in each scenario and present the results in Sections 4.2.1 to 4.2.6. For all scenarios we set the initial parameter values in the EM algorithm to the true parameter values to speed up computation. In Section 4.3.4 (Scenario 4 of Section 4.3) we considered initial values that differed from the true parameter values based on the K-means clustering method and the EM algorithm also converged; however, it took longer than when starting from the truth, as expected. As shown in the steps of the algorithms on Chapter 3, we repeat the E-step and M-step of the EM algorithm until $[\ell(\boldsymbol{\theta}^{(t+1)}\,|\,\boldsymbol{y}) - \ell(\boldsymbol{\theta}^{(t)}\,|\,\boldsymbol{y})] \leq tolerance$ or the maximum number of iteration reached. For almost all of our simulation scenarios, the first stopping rule of the algorithm ($[\ell(\boldsymbol{\theta}^{(t+1)}\,|\,\boldsymbol{y}) - \ell(\boldsymbol{\theta}^{(t)}\,|\,\boldsymbol{y})] \leq tol$) reached before continuation until the maximum number of iterations.

4.2.1 Scenario 1

In this scenario, we vary N according to six different values (cases), while all other parameters are kept fixed as shown in Table 4.2 below. Three distinct values are chosen for the rate parameters (λ_{kg}'s) and we repeat the same value for a third of the number of genes in each cluster (i.e., $\frac{G}{3} = 40$ times), in a way that the rate parameters are distinct for each gene and across clusters. For this scenario, we choose $\lambda_1 = 5$, $\lambda_2 = 10$, and $\lambda_3 = 15$ and we use the following

Table 4.1: Settings used for each simulation study scenario. The $\star$ indicates the parameter or hyperparameter that varies in each scenario.

Scenario	N	G	K	ϕ_k	π_k
1	$\star$	120	3	0.1	$\frac{1}{K}$
2	1200	$\star$	3	0.1	$\frac{1}{K}$
3	1200	120	$\star$	0.1	$\frac{1}{K}$
4	1200	120	2	0.1	$\star$
5‡	1200	120	2	0.1	$\frac{1}{K}$
6	1200	120	3	$\star$	$\frac{1}{K}$

‡ The settings are kept the same but we vary the similarities among clusters by changing the λ_{gk}'s.

matrix for generating the simulated data sets:

$$\lambda = \begin{pmatrix} \lambda_1, & \ldots, & \lambda_1, & \lambda_2, & \ldots, & \lambda_2, & \lambda_3, & \ldots, & \lambda_3 \\ \lambda_2, & \ldots, & \lambda_2, & \lambda_3, & \ldots, & \lambda_3, & \lambda_1, & \ldots, & \lambda_1 \\ \lambda_3, & \ldots, & \lambda_3, & \lambda_1, & \ldots, & \lambda_1, & \lambda_2, & \ldots, & \lambda_2 \end{pmatrix}.$$

Figure 4.1 shows an example of simulated data for Case 3 ($N = 120$) in Table 4.2.

Table 4.2: ZIP mixture model without covariates. Scenario 1: Values chosen for the number of observations N in each of five cases along with the fixed parameters used to simulate the datasets under a ZIP mixture model without covariates.

Case	N	G	K	ϕ_k	π_k
1	12				
2	60				
3	120	120	3	0.1	$1/K$
4	600				
5	1200				

For this scenario, Figures 4.2 and 4.3, and Tables 4.3 and 4.4 show that the EM estimates for π_k and ϕ_k, for $k = 1, 2$, and 3, are centered around the true values across all the different choices of N, except for ϕ_1, ϕ_2 and ϕ_3 when $N = 12$. Furthermore, according to Tables 4.3 and 4.4, as N increases, the standard deviations of these estimates decrease as desired. Table 4.5 demonstrates that the MSE for estimating the λ_{gk}'s decreases while N increases. Moreover, according to Figure 4.4, the clustering performance measured by the V-measures is deemed satisfactory except for the lowest value of $N = 12$, which results in some misclassifications. Finally, we can see from Figures B.1 and B.2 and Tables B.1 and B.2 that although the computation time increases, as N increases, the number of iterations until convergence decreases for

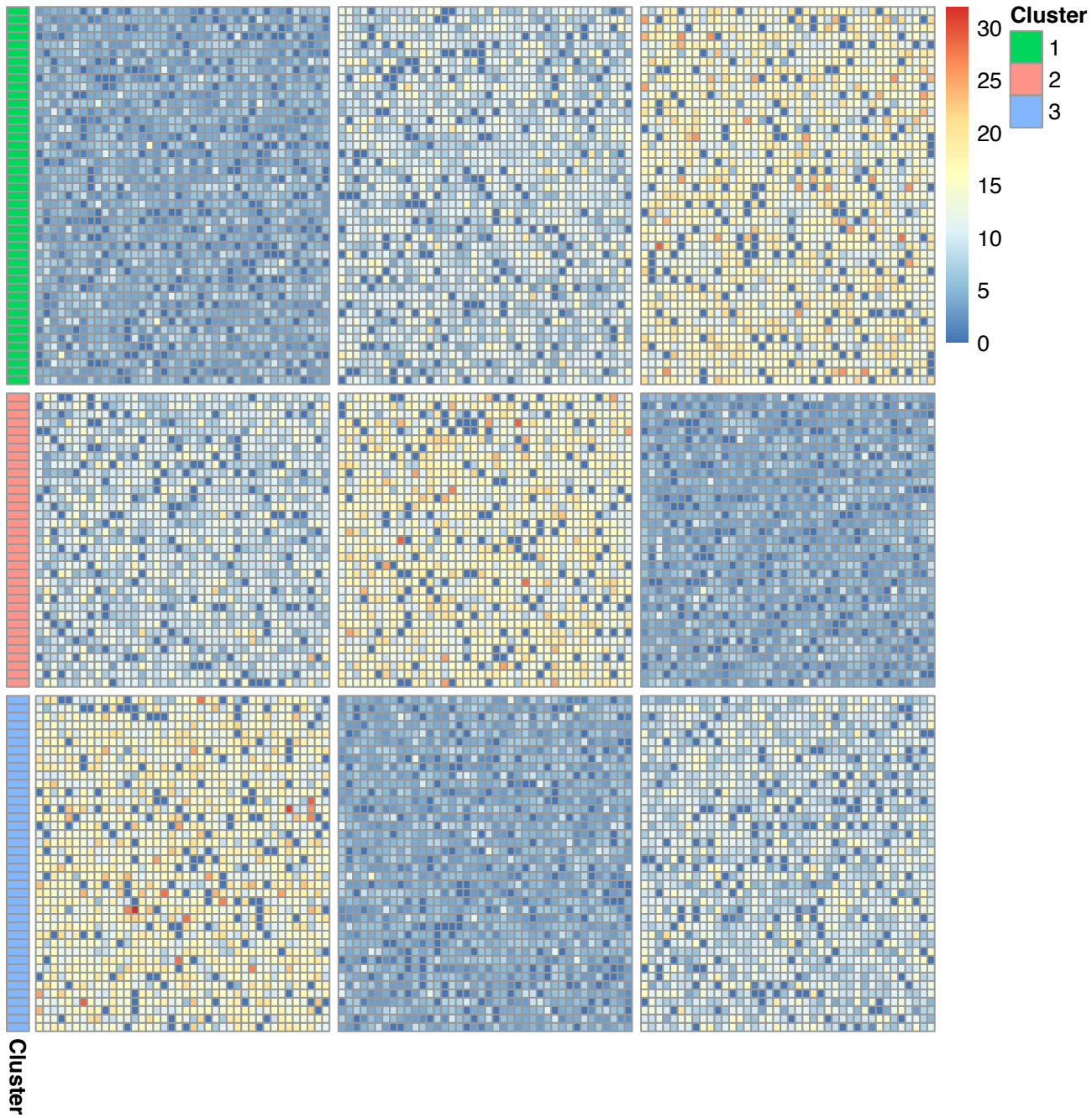

Figure 4.1: Scenario 1: Heatmap of a simulated data set generated according to the settings in Case 3 of Table 4.2. Darker colors represent higher counts. The assigned true clusters at the simulation stage are represented by the colored column on the left side of the plot.

the first three cases and then stabilizes for the larger N cases.

Table 4.3: Scenario 1: Mean and standard deviation (SD) for the estimates of ϕ_k obtained for each k and each N using the EM algorithm across the datasets simulated from the settings described in Table 4.2.

k	N	Mean	SD
	12	0.09052	0.02572
	60	0.10002	0.00581
1	120	0.09998	0.00414
	600	0.09990	0.00190
	1200	0.09994	0.00136
	12	0.08971	0.02448
	60	0.09949	0.00608
2	120	0.09966	0.00432
	600	0.10004	0.00188
	1200	0.10003	0.00132
	12	0.09242	0.02372
	60	0.10024	0.00616
3	120	0.09989	0.00464
	600	0.10015	0.00194
	1200	0.10012	0.00155

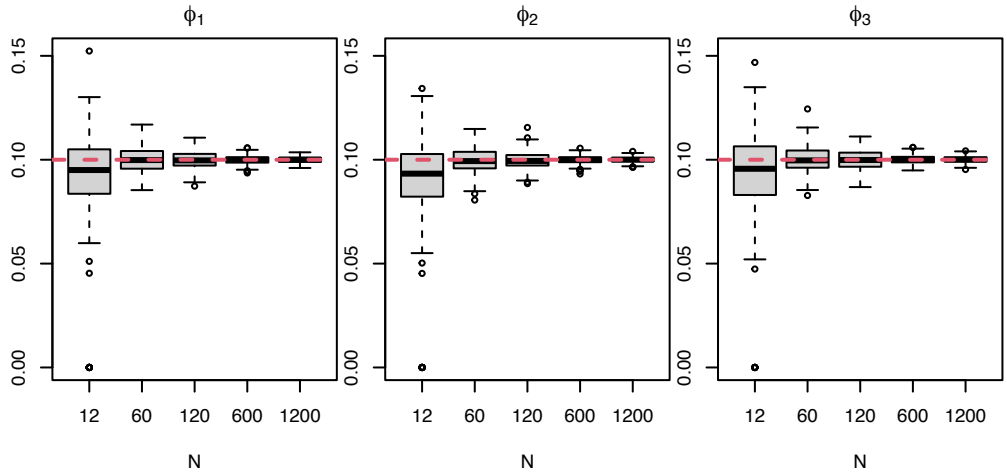

Figure 4.2: Scenario 1: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.2. Red lines correspond to true values. See also Table 4.3.

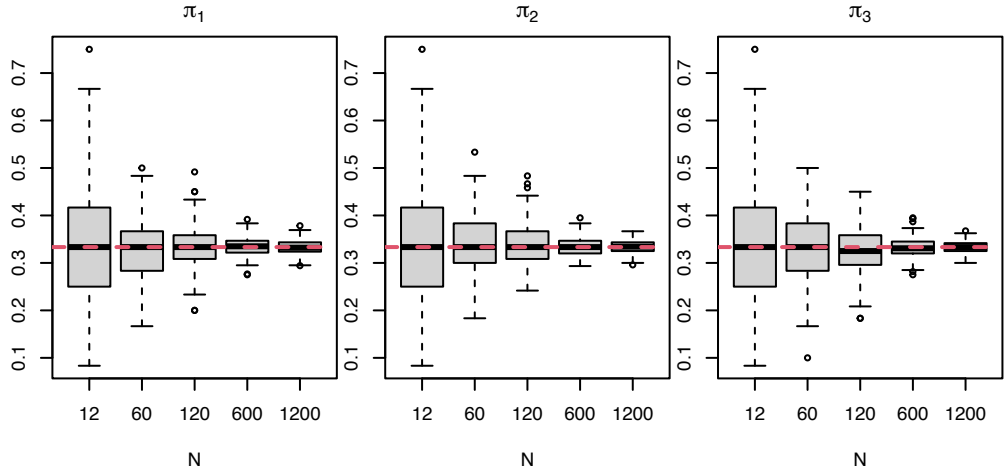

Figure 4.3: Scenario 1: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.2. Red lines correspond to true values. See also Table 4.4.

Table 4.4: Scenario 1: Mean and standard deviation (SD) for the estimates of π_k obtained for each k and each N using the EM algorithm across the datasets simulated from the settings described in Table 4.2.

k	N	Mean	SD
1	12	0.33431	0.13777
	60	0.32689	0.05793
	120	0.33385	0.04303
	600	0.33447	0.01805
	1200	0.33274	0.01361
2	12	0.34017	0.13359
	60	0.33906	0.06123
	120	0.33815	0.04353
	600	0.33371	0.01862
	1200	0.33420	0.01334
3	12	0.32552	0.13445
	60	0.33405	0.06115
	120	0.32799	0.04730
	600	0.33182	0.01967
	1200	0.33306	0.01287

Table 4.5: Scenario 1: Mean squared error across genes and simulated datasets for the EM estimates of the λ_{gk}'s for each cluster k and each N according to the settings described in Table 4.2.

	N				
k	12	60	120	600	1200
1	4.66910	0.58775	0.28343	0.05595	0.02819
2	4.18363	0.57776	0.28229	0.05559	0.02740
3	4.48772	0.57781	0.29115	0.05579	0.02800

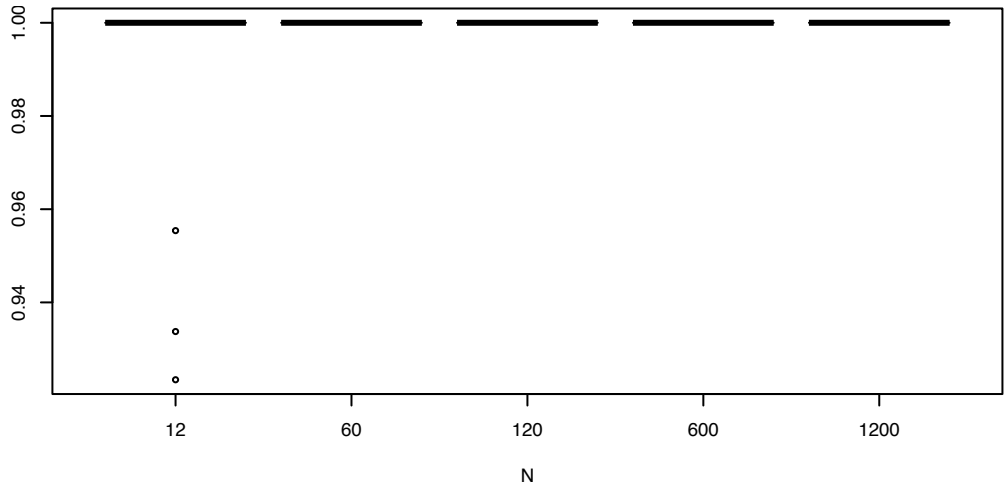

Figure 4.4: Scenario 1: Boxplots of the V-measures comparing the EM clustering assignments with true cluster labels, across the datasets simulated from the settings described in Table 4.2.

4.2.2 Scenario 2

In this scenario, the number of genes G varies while the other parameters are kept fixed, as shown in Table 4.6. The number of clusters, the cluster assignment probabilities, the probabilities of always zero, and the λ_{gk}'s are fixed to values similar to those in Scenario 1. We fix $N = 1200$ for this scenario, as seen in Table 4.6.

Table 4.6: ZIP mixture model without covariates. Scenario 2: Values chosen for the number of genes G in each of five cases along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1		12			
2		60			
3	1200	120	3	0.1	$1/K$
4		600			
5		1500			

Figure 4.5 and Table 4.7 show that as G increases, the standard deviation of the estimates of each ϕ_k decreases, while the estimates remain centred around the true value of 0.1. For the estimates of π_k, according to Figure (4.6) and Table (4.8), their standard deviations remain somewhat the same when varying G across clusters. In addition, estimates of π_k are unbiased, as in Scenario 1. Table 4.9 shows that the MSE of the estimates of the λ_{gk}'s for each cluster remains almost the same while varying G. Furthermore, the resulting V-measure values (Figure 4.7) are equal to one for $G = 60, 120, 600, 1500$. However, for $G = 12$, a few misclassifications lead to V-measure values slightly less than one. The number of iterations remains constant (Table B.3 and Figure B.3), and the total computing time increases (Table B.4 and Figure B.4) as G increases.

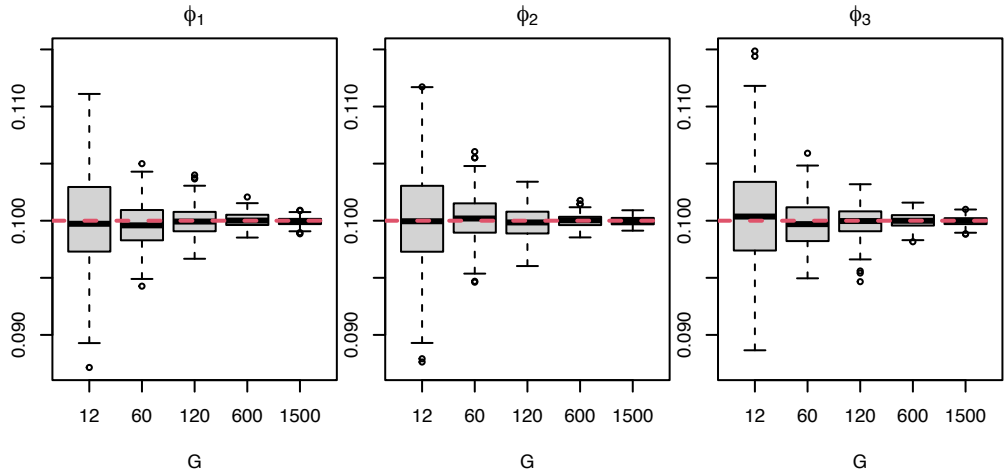

Figure 4.5: Scenario 2: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.6. Red lines correspond to true values. See also Table 4.7.

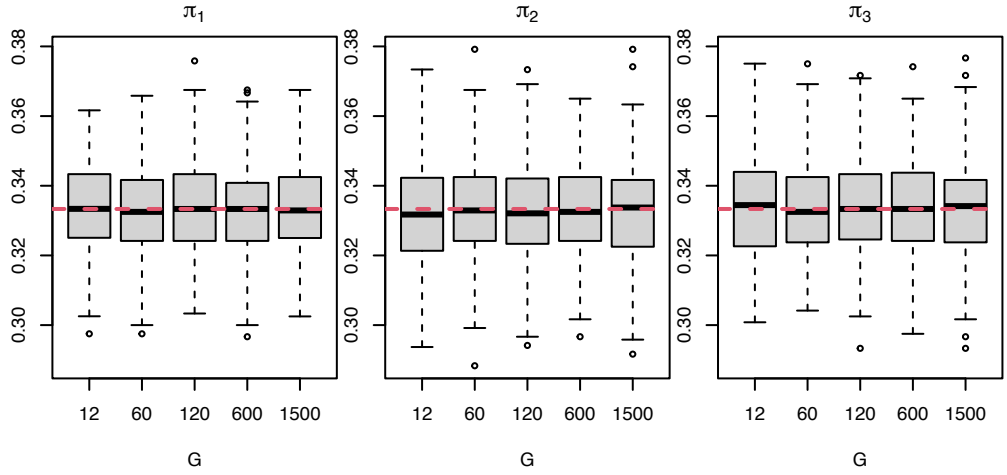

Figure 4.6: Scenario 2: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.6. Red lines correspond to true values. See also Table 4.8.

Table 4.7: Scenario 2: Mean and standard deviation (SD) for the estimates of ϕ_k obtained for each k and each G using the EM algorithm across the datasets simulated from the settings described in Table 4.6.

k	G	Mean	SD
	12	0.09992	0.00450
	60	0.09966	0.00195
1	120	0.09998	0.00133
	600	0.10007	0.00063
	1500	0.09993	0.00036
	12	0.09997	0.00437
	60	0.10018	0.00206
2	120	0.09985	0.00142
	600	0.10000	0.00056
	1500	0.09999	0.00038
	12	0.10050	0.00459
	60	0.09977	0.00198
3	120	0.09995	0.00133
	600	0.09998	0.00066
	1500	0.09998	0.00042

Table 4.8: Scenario 2: Mean and standard deviation (SD) for the estimates of π_k obtained for each k and each G using the EM algorithm across the datasets simulated from the settings described in Table 4.6.

k	G	Mean	SD
	12	0.33396	0.01266
	60	0.33258	0.01346
1	120	0.33403	0.01329
	600	0.33247	0.01318
	1500	0.33383	0.01323
	12	0.33200	0.01446
	60	0.33375	0.01365
2	120	0.33238	0.01350
	600	0.33339	0.01319
	1500	0.33291	0.01353
	12	0.33404	0.01374
	60	0.33367	0.01395
3	120	0.33359	0.01362
	600	0.33414	0.01320
	1500	0.33325	0.01389

Table 4.9: Scenario 2: Mean squared error across genes and simulated datasets for the EM estimates of the λ_{gk}'s for each cluster k and each G according to the settings described in Table 4.6.

	G				
k	12	60	120	600	1500
1	0.02826	0.02804	0.02796	0.02823	0.02792
2	0.02823	0.02838	0.02794	0.02799	0.02810
3	0.02804	0.02879	0.02777	0.02791	0.02792

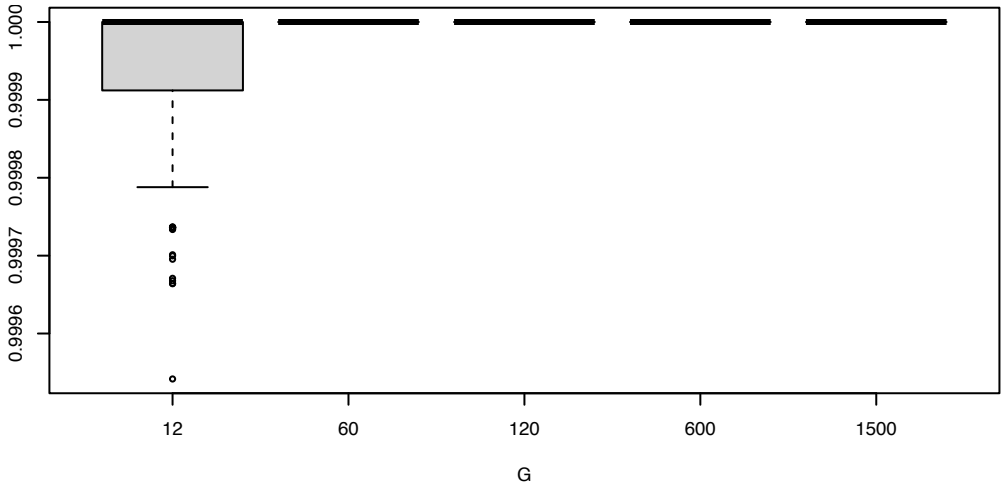

Figure 4.7: Scenario 2: Boxplots for the V-measures comparing the EM clustering assignments with true cluster labels, across the datasets simulated from the settings described in Table 4.6.

4.2.3 Scenario 3

For this scenario, the number of clusters K changes, and, therefore, the corresponding probabilities of cluster assignments also change to maintain the balanced format of the clusters. We keep the values of the other parameters fixed, and their choice is shown in Table 4.10. Note that when the number of clusters varies, the number of ϕ_k parameters changes, but their values are the same. For example, we have ϕ_1 and ϕ_2 for the case of $K = 2$ and ϕ_1, ϕ_2, ϕ_3, ϕ_4, ϕ_5 for $K = 5$, but their values are all equal to 0.1. For Scenario 3, the true values of the λ_{gk}'s used to simulate the data are as follows:

$$K = 1 \rightarrow \lambda_{gk}\text{'s} = 10$$

$$K = 2 \rightarrow \lambda_{gk}\text{'s} = (10, 15)$$

$$K = 3 \rightarrow \lambda_{gk}\text{'s} = (5, 10, 15)$$

$$K = 5 \rightarrow \lambda_{gk}\text{'s} = (5, 10, 15, 20, 25)$$

So that, for example, for $K = 5$ in the above setting, the first $\frac{120}{5} = 24$ genes in cluster $k = 1$ are assigned λ_{gk} values equal to 5, then the following 24 genes are assigned values of 10, etc. We repeat this process for the remaining clusters so that each gene's rate parameters are distinct across clusters.

Table 4.10: ZIP mixture model without covariates. Scenario 3: Values chosen for the number of clusters K in each of four different cases along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1			1		
2	1200	120	2	0.1	$1/K$
3			3		
4			5		

We can observe from Table (4.11) and Figure (4.8) that as the number of clusters increases, the standard deviation of the estimates of ϕ_k increases due to the reduction in the number of observations per cluster since N is fixed. In addition, the estimates of ϕ_k's are centered around the true values used to simulate the data. Figure 4.9 and Table 4.12 show that the estimates of π_k are also centered around the true values. Note that in Figure 4.9, we merge the vectors of π_k estimates for different K into one vector in order to show their behaviour better. The standard deviations of the estimates of π_k remain somewhat the same across the different K choices (Table 4.12). From Table 4.13, we can see that for the λ_{gk}'s, the MSE increases as the number

of clusters increases. In this simulation scenario, all V-measure values remain constant and equal to one, demonstrating no misclassification. The number of iterations until convergence remains the same in all cases (Table B.5 and Figure B.5); however, we observe an increase in computing time when the number of clusters increases (Table B.6 and Figure B.6).

Table 4.11: Scenario 3: Mean and standard deviation (SD) for the estimates of ϕ_k obtained for each k in each choice of K using the EM algorithm across the datasets simulated from the settings described in Table 4.10. Regarding the table structure, note that ϕ_1 exists for $K = 1, 2, 3$, and 5 clusters (all cases), while ϕ_2 only exists for $K = 2, 3, 5$, ϕ_3 exists for only $K = 3, 5$, and finally, ϕ_4 and ϕ_5 exist only for $K = 5$.

k	K	Mean	SD
	1	0.10001	0.00077
1	2	0.10008	0.00111
	3	0.10004	0.00133
	5	0.10001	0.00174
	2	0.10000	0.00109
2	3	0.09991	0.00140
	5	0.10012	0.00174
3	3	0.10008	0.00135
	5	0.10007	0.00169
4	5	0.10011	0.00178
5	5	0.09977	0.00186

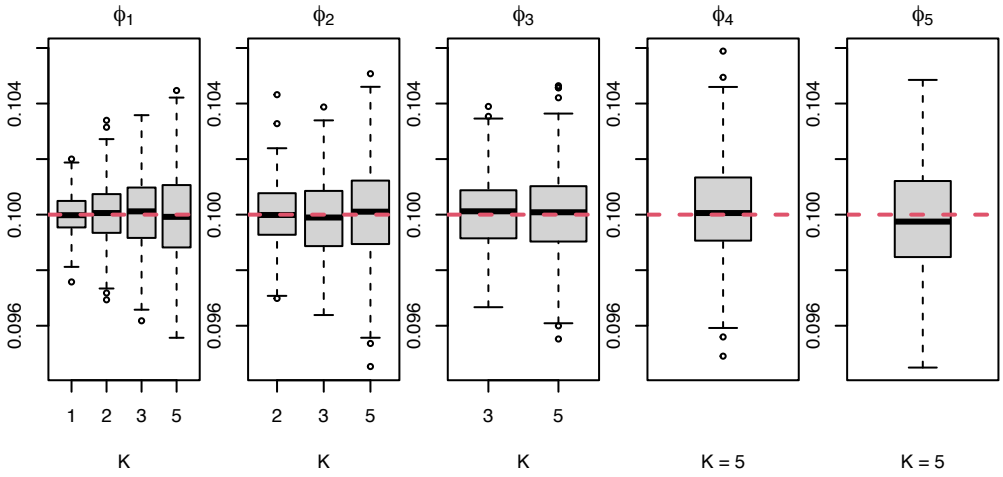

Figure 4.8: Scenario 3: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.10. Red lines correspond to true values. See also Table 4.11. Note that ϕ_1 exists for $K = 1, 2, 3$, and 5 clusters (all cases), while ϕ_2 only exists for $K = 2, 3, 5$, ϕ_3 exists for only $K = 3, 5$, and finally, ϕ_4 and ϕ_5 exist only for $K = 5$.

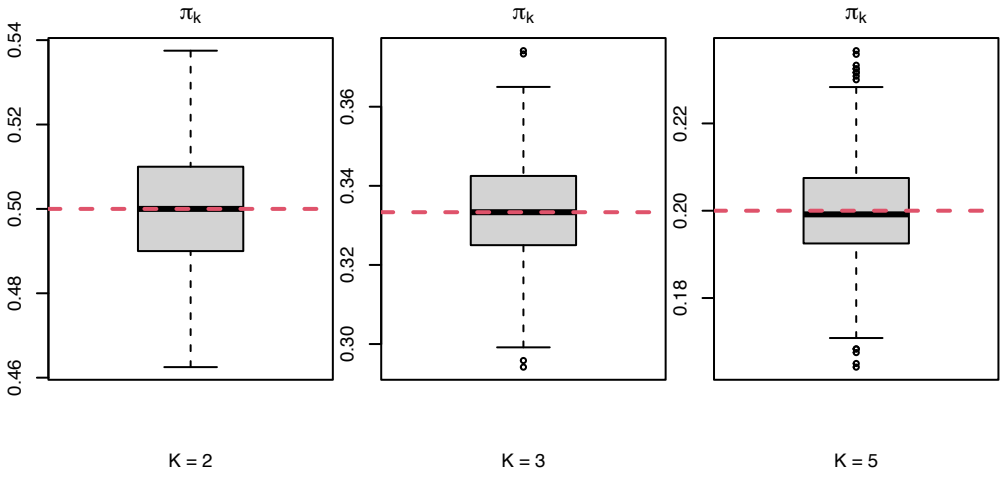

Figure 4.9: Scenario 3: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.10. Red lines correspond to true values. See also Table 4.12. Note that the estimates of π_k over different K are all merged into one vector.

Table 4.12: Scenario 3: Mean and standard deviation (SD) for the estimates of π_k obtained for each k in each choice of K using the EM algorithm across the datasets simulated from the settings described in Table 4.10. Regarding the table structure, note that π_1 exists for $K = 1, 2, 3$, and 5 clusters (all cases), while π_2 only exists for $K = 2, 3, 5$, π_3 exists for only $K = 3, 5$, and finally, π_4 and π_5 exist only for $K = 5$. In addition, π_1 is always equal to 1 when $K = 1$.

k	K	Mean	SD
	1	-	-
1	2	0.49974	0.01406
	3	0.33266	0.01289
	5	0.20015	0.01160
	2	0.50026	0.01406
2	3	0.33458	0.01281
	5	0.19951	0.01182
3	3	0.33276	0.01288
	5	0.20049	0.01176
4	5	0.19993	0.01149
5	5	0.19992	0.01179

Table 4.13: Scenario 3: Mean squared error for the EM estimates of the λ_{gk}'s for each k and each K across genes and simulated datasets according to the settings described in Table 4.10.

	K			
k	1	2	3	5
1	0.00921	0.02316	0.02767	0.06997
2		0.02309	0.02766	0.06998
3			0.02817	0.06918
4				0.07115
5				0.06950

4.2.4 Scenario 4

In this scenario, we consider two clusters and study the effect of changing the probabilities of the clusters from a balanced case to a very unbalanced case. Changes in cluster probabilities can be seen in the columns π_1 and π_2 of Table 4.14. All other parameters and hyperparameters are kept fixed according to the settings shown in Table 4.14.

Table 4.14: ZIP mixture model without covariates. Scenario 4: Values chosen for the proportion π_k assigned to each cluster k in each of four different cases along with the fixed parameters used to simulate the datasets. Note that $\pi_2 = 1 - \pi_1$.

Case	N	G	K	ϕ_k	π_1	π_2
1					0.50	0.50
2	1200	120	2	0.1	0.25	0.75
3					0.10	0.90
4					0.05	0.95

We can see in Figure 4.10 and Table 4.15 that as the cluster proportions change from balanced to unbalanced, the standard deviations of estimated values of ϕ_1 increase. At the same time, no significant changes are observed in the standard deviations of the estimates of ϕ_2 as $\pi_2 \geq \pi_1$ across all cases. Moreover, the estimates of ϕ_1 and ϕ_2 are centered around their true values. Figure 4.11 and Table 4.16 show that the estimated values of π_k are centered around their true values, and the standard deviations are small for all cases. For the λ_{gk}'s, from Table 4.17, we can observe that the MSEs increase for the first cluster while slightly decreasing for the second cluster as π_1 and π_2 change from balanced to unbalanced. The V-measures for this scenario are equal to one for all cases. For the number of iterations and computing times, no considerable differences are observed in the balanced and unbalanced scenarios (Tables B.7, B.8 and Figures B.7, B.8).

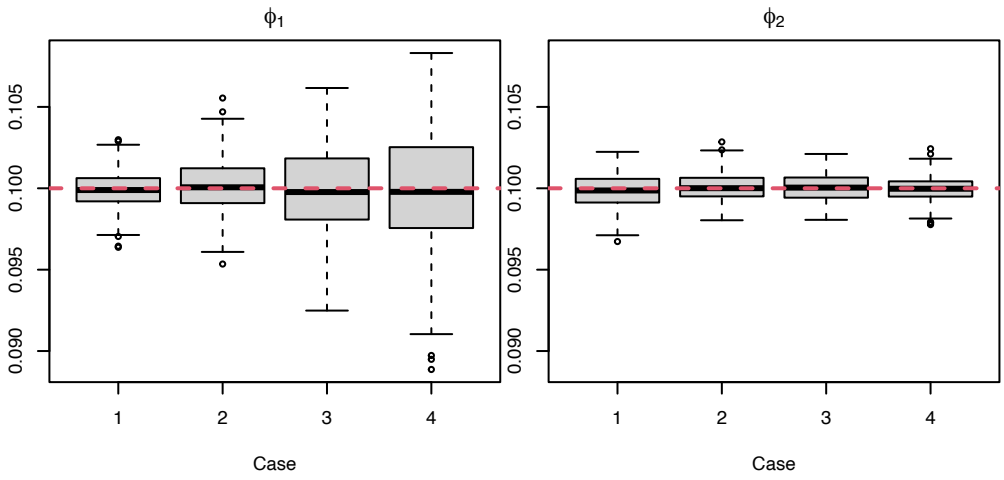

Figure 4.10: Scenario 4: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.14. Red lines correspond to true values. See also Table 4.15.

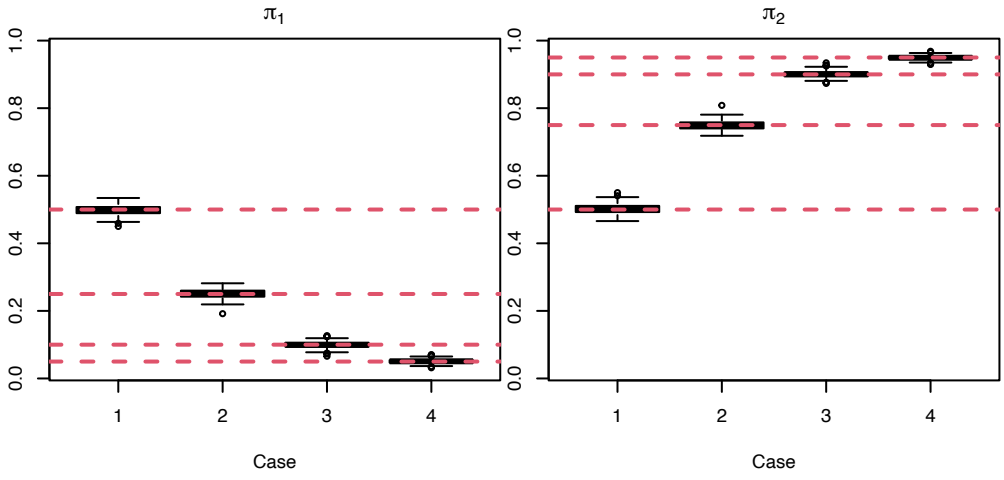

Figure 4.11: Scenario 4: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.14. Red lines correspond to true values. See also Table 4.16.

Table 4.15: Scenario 4: Mean and standard deviation (SD) for the estimates of ϕ_k obtained for each k and each case using the EM algorithm across the datasets simulated from the settings described in Table 4.14.

k	Case	Mean	SD
1	1	0.09993	0.00113
	2	0.10008	0.00167
	3	0.09995	0.00251
	4	0.09984	0.00362
2	1	0.09989	0.00106
	2	0.10005	0.00090
	3	0.10003	0.00084
	4	0.09998	0.00076

Table 4.16: Scenario 4: Mean and standard Deviation for the estimates of π_k obtained for each k and each case using the EM algorithm across the datasets simulated from the settings described in Table 4.14.

k	Case	Mean	SD
1	1	0.49846	0.01470
	2	0.25039	0.01319
	3	0.09937	0.00912
	4	0.05069	0.00653
2	1	0.50154	0.01470
	2	0.74961	0.01319
	3	0.90063	0.00912
	4	0.94931	0.00653

Table 4.17: Scenario 4: Mean squared error for the EM estimates of the λ_{gk}'s for each k and each case across genes and simulated datasets according to the settings described in Table 4.14.

	Case			
k	1	2	3	4
1	0.02306	0.04653	0.11769	0.23423
2	0.02327	0.01544	0.01303	0.01208

4.2.5 Scenario 5

In this scenario, we vary the choice of the rate parameters, the λ_{gk}'s. In all other previous scenarios, their values are selected to avoid any overlap between the clusters. Now, in Scenario 5, considering p_λ as the proportion of overlap of the λ_{gk} values across clusters, we start from no overlap (Case 1) and go until the case of having two-thirds of the genes with the same λ_{gk} values across clusters (Case 5). We consider the true values of 4 and 5 for the λ_{gk}'s. All other parameters are kept fixed in this scenario. The settings used to simulate data under this scenario are shown in Table 4.18.

Table 4.18: ZIP mixture model without covariates. Scenario 5: Values chosen for the proportion of λ_{gk} parameters in common between the two clusters, p_λ, in each of six possible cases along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k	p_λ
1						0
2						$1/6$
3	1200	120	2	0.1	$1/K$	$1/3$
4						$1/2$
5						$2/3$

Figure 4.12 and Table 4.19 show that the EM estimates of ϕ_1 and ϕ_2 remain somewhat unbiased and with small standard deviations across all cases in Scenario 5. For the estimates of π_1 and π_2, we can see from Figure 4.13 and Table 4.20 that they are all unbiased, while their standard deviations slightly increase in cases 4 and 5. The MSEs of the estimated values of λ_{gk} are slightly similar across the cases according to Table 4.21. Moreover, as expected, according to the V-measures, clustering performance decreases as the overlap between the clusters increases (see Figure 4.14). Also, computing time increases for the most complicated case with more cluster overlap (Table B.9 and Figure B.9). Finally, the number of iterations also increases for the more complicated cases, as expected (see Table B.10 and Figure B.10).

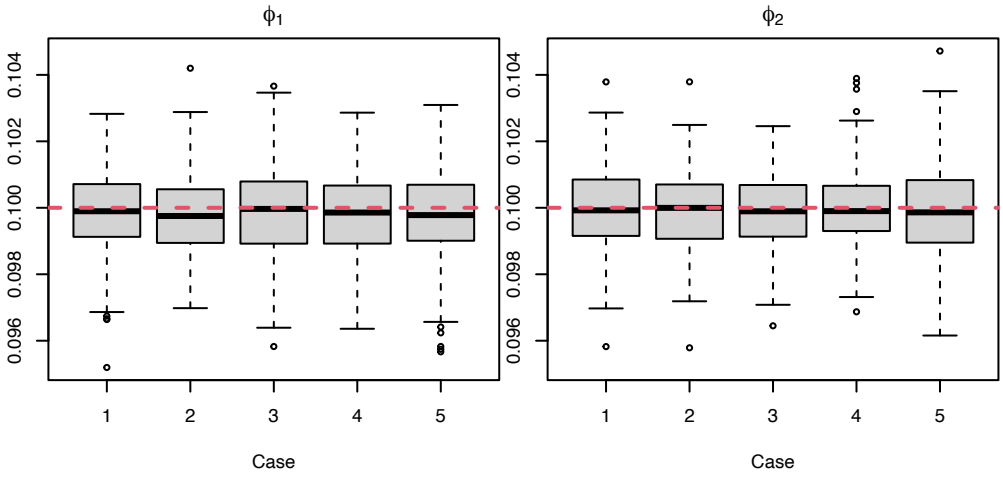

Figure 4.12: Scenario 5: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.18. Red lines correspond to true values. See also Table 4.19.

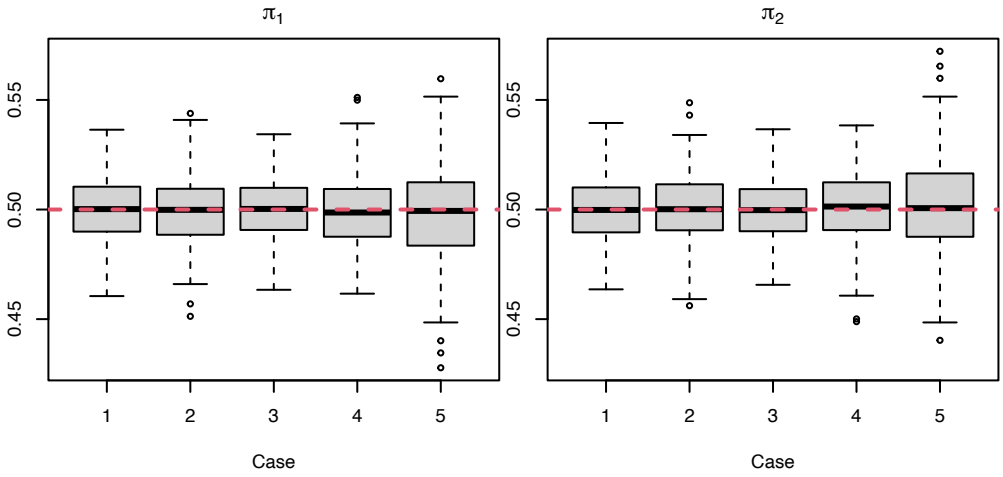

Figure 4.13: Scenario 5: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.18. Red lines correspond to true values. See also Table 4.20.

Table 4.19: Scenario 5: Mean and standard deviation (SD) for the estimates of ϕ_k obtained for each k and each case using the EM algorithm across the datasets simulated from the settings described in Table 4.18.

k	Case	Mean	SD
1	1	0.09989	0.00123
	2	0.09978	0.00122
	3	0.09991	0.00132
	4	0.09985	0.00130
	5	0.09975	0.00134
2	1	0.09998	0.00116
	2	0.09991	0.00115
	3	0.09994	0.00111
	4	0.10000	0.00113
	5	0.09989	0.00145

Table 4.20: Scenario 5: Mean and standard deviation (SD) for the estimates of π_k obtained for each k and each case using the EM algorithm across the datasets simulated from the settings described in Table 4.18.

k	Case	Mean	SD
1	1	0.50024	0.01459
	2	0.49967	0.01485
	3	0.50013	0.01443
	4	0.49850	0.01666
	5	0.49791	0.02190
2	1	0.49976	0.01459
	2	0.50033	0.01485
	3	0.49987	0.01443
	4	0.50150	0.01666
	5	0.50209	0.02190

Table 4.21: Scenario 5: Mean squared error for the EM estimates of the λ_{gk}'s for each k and each case across genes and simulated datasets according to the settings described in Table 4.18.

	Case				
k	1	2	3	4	5
1	0.00804	0.00813	0.00827	0.00874	0.00945
2	0.00965	0.00967	0.00947	0.00952	0.00996

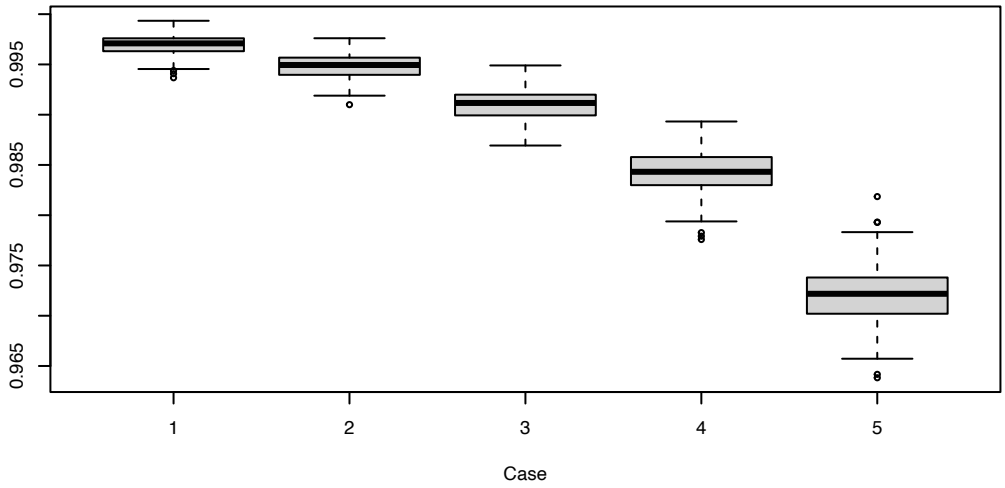

Figure 4.14: Scenario 5: Boxplots of the V-measures of the clustering obtained by the EM algorithm across the datasets simulated from the settings described in Table 4.18.

4.2.6 Scenario 6

The effect of changes in the probability of always zero (ϕ_k) is studied in this scenario. Considering three clusters, we first consider the same probability of always zero across all clusters, while these probabilities change from small to large (cases 1 to 5). For the last case, case 6, we consider different probabilities across clusters (see Table 4.22).

Table 4.22: ZIP mixture model without covariates. Scenario 6: Values chosen for the probability of always zero ϕ_k in each of six different cases along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1				0.05	
2				0.10	
3	1200	120	3	0.25	$1/K$
4				0.50	
5				0.90	
6				(0.05, 0.25, 0.50)	

Both Figure 4.15 and Table 4.23 demonstrate small values of the standard deviation of the estimates of ϕ_k's and no remarkable changes across the different cases. Also, the estimates are all centered around their true values. It should be mentioned that in Figure 4.15, for cases 1 to 5, we merge the estimates of ϕ_1, ϕ_2, and ϕ_3 into one vector to show their behaviour better. No significant differences are also observed in the means and standard deviations of the estimate of π_k's (Table 4.24 and Figure 4.16). Table 4.25 shows that, for any cluster k, the MSE of the estimates of the λ_{gk}'s increases as the probability of always zero increases. We can also observe in Table 4.25 that for each cluster k, the MSEs are similar, except for the last case (case 6), where there are different probabilities of always zero in each cluster. The V-measures show a slightly worse clustering performance in case 5, where there are more zeros ($\phi_1 = \phi_2 = \phi_3 = 0.9$) than in the other cases (Figure 4.17). Moreover, for case 5, standard deviations of the number of iterations are slightly larger than for the other cases (Table B.11 and Figure B.11). Also, computing times increase for cases 4 and 5, where the probabilities of always zero are large, and for case 6, where they vary across the clusters (Table B.12 and Figure B.12).

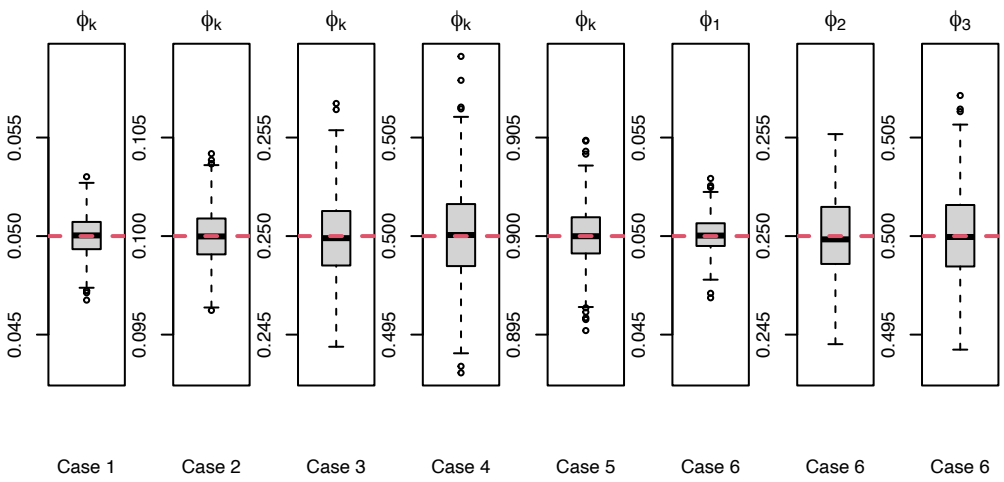

Figure 4.15: Scenario 6: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.22. Red lines correspond to true values. See also Table 4.23. Note that the estimates of ϕ_1, ϕ_2, and ϕ_3 are all merged into one vector for cases 1-5.

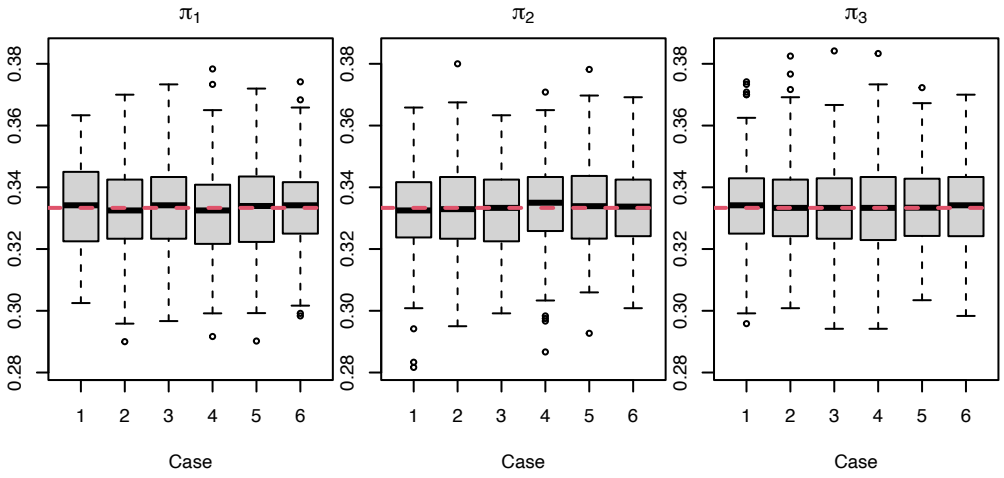

Figure 4.16: Scenario 6: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.22. Red lines correspond to true values. See also Table 4.24.

Table 4.23: Scenario 6: Mean and standard deviation for the estimates of ϕ_k obtained for each k and each case using the EM algorithm across the datasets simulated from the settings described in Table 4.22.

ϕ_k	Case	Mean	SD
ϕ_1	1	0.05006	0.00093
	2	0.10003	0.00140
	3	0.24992	0.00188
	4	0.50034	0.00234
	5	0.90006	0.00144
	6	0.05007	0.00099
ϕ_2	1	0.04989	0.00106
	2	0.09994	0.00134
	3	0.25001	0.00195
	4	0.49992	0.00214
	5	0.89987	0.00139
	6	0.24996	0.00201
ϕ_3	1	0.05006	0.00104
	2	0.10010	0.00145
	3	0.24981	0.00195
	4	0.49990	0.00239
	5	0.90006	0.00137
	6	0.50007	0.00240

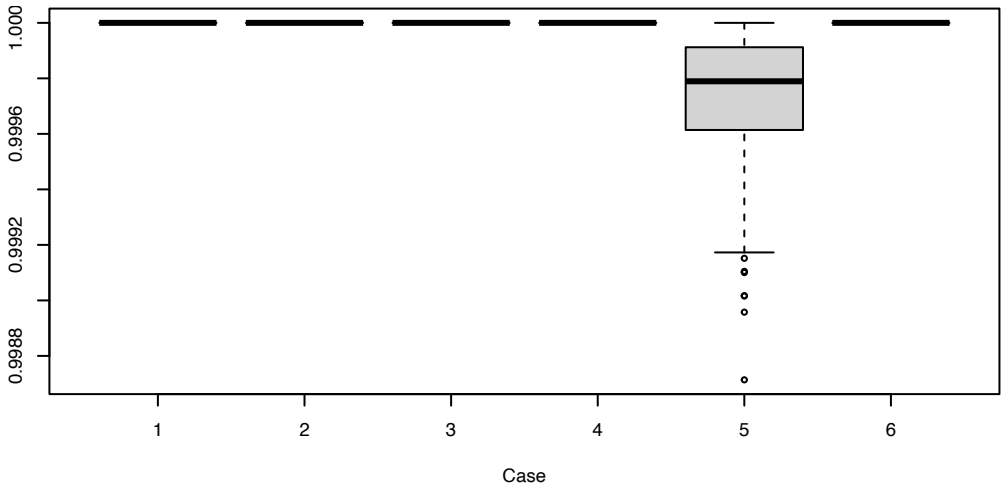

Figure 4.17: Scenario 6: Boxplots for the V-measures of the clustering obtained by the EM algorithm across the datasets simulated from the settings described in Table 4.22.

Table 4.24: Scenario 6: Mean and standard deviation for the estimates of π_k obtained for each k and each case using the EM algorithm across the datasets simulated from the settings described in Table 4.22.

π_k	Case	Mean	SD
π_1	1	0.33378	0.01395
	2	0.33344	0.01388
	3	0.33390	0.01424
	4	0.33157	0.01438
	5	0.33285	0.01484
	6	0.33345	0.01359
π_2	1	0.33214	0.01418
	2	0.33278	0.01467
	3	0.33251	0.01354
	4	0.33445	0.01347
	5	0.33357	0.01382
	6	0.33315	0.01287
π_3	1	0.33408	0.01424
	2	0.33379	0.01367
	3	0.33359	0.01403
	4	0.33398	0.01428
	5	0.33358	0.01323
	6	0.33340	0.01387

Table 4.25: Scenario 6: Mean squared error for the EM estimates of the λ_{gk}'s for each k and each N across genes and simulated datasets according to the settings described in Table 4.22.

	Case					
k	1	2	3	4	5	6
1	0.02645	0.02775	0.03323	0.05075	0.25699	0.02649
2	0.02629	0.02795	0.03420	0.05042	0.25946	0.03409
3	0.02626	0.02808	0.03321	0.05096	0.25699	0.05017

4.3 Simulation scenarios for the ZIP mixture model with a size factor

In the simulation scenarios of this section, we model the rate parameter (expected read count) via a log link function as $\log(\lambda_{ngk}) = \log(T_n) + \beta_{0g} + \rho_{gk}$, which was introduced earlier in Section 3.5.2 of Chapter 3. So, to simulate data for the scenarios described in the following subsections, we construct the rate parameter by simulating the size factors, T_n, for $n = 1 \ldots, N$, from a normal distribution with parameters $\mu = 1000$, $\sigma = 100$, and we fix β_{0g} at a value of one for all genes. For most of the cases, we choose three distinct values (−0.6, 0 and 0.6) for the cluster effects (ρ_{gk}'s), and we repeat the same value for a third of the number of genes in each cluster (i.e., $\frac{G}{3}$ times), in a way that the rate parameters for each gene are distinct across clusters. Note that, to avoid identifiability issues, we consider the restriction of $\sum_{k=1}^{K} \rho_{gk} = 0$ to select the values of ρ_{gk}.

We study a total of six scenarios in this section. The number of simulated datasets in each scenario is $S = 256$. In scenarios 1, 2, and 3, we vary N, G, and K, respectively. In scenario 4, we consider two cases. In case 1, the initial parameter values for the EM algorithm are equal to the true values used to generate the data. For case 2, we obtain the initial parameter values based on the results of K-means clustering. In scenario 5, we consider different values for the probabilities of cluster assignments. Finally, in scenario 6, we vary the probabilities of always zero and compare the results for this scenario. Note that, as in Section 4.2, for all cases, except case 2 of Scenario 4, we set the initial parameter values in the EM algorithm to the true parameter values to speed up computation.

4.3.1 Scenario 1

In this scenario, similar to the case without covariates in Section 4.2.1, we vary N while all other parameters and hyperparameters are kept fixed. See Table 4.26 for the parameter setting used to generate data under this scenario.

As can be seen from the boxplots and tables (Tables 4.27, 4.28 and Figures 4.18, 4.19), the estimated values of probability of always zero ($\hat{\phi}_k$) and the estimated values of the cluster assignment's probabilities ($\hat{\pi}_k$), are both approximately around their true values and as N increases, their standard deviations decrease. For the estimated values of the parameters ρ_{gk} and β_{0g}, we consider the median absolute deviation (MAD) and we can observe from Tables 4.29 and 4.30 that as N increases, the MADs for both ρ_{gk} and β_{0g} decrease. For most cases, the V-measures are one (see Figure 4.20), except for the case with $N = 12$ where there is some misclassification. As expected, the computing times increases while N increases (Table B.13

Table 4.26: ZIP mixture model with a size factor. Scenario 1: Values chosen for the number of observations N in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1	12				
2	60				
3	120	120	3	0.1	$1/K$
4	600				
5	1200				

and Figure B.13). Finally, the required number of iterations to achieve convergence decreases as N grows (Table B.14 and Figure B.14).

Table 4.27: Scenario 1: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each N, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.26.

ϕ_k	N	Mean	SD
	12	0.09738	0.01496
	60	0.10003	0.00655
ϕ_1	120	0.09983	0.00416
	600	0.09999	0.00195
	1200	0.10027	0.00125
	12	0.09748	0.01473
	60	0.10020	0.00628
ϕ_2	120	0.09978	0.00448
	600	0.10001	0.00199
	1200	0.10020	0.00139
	12	0.09714	0.01625
	60	0.09961	0.00642
ϕ_3	120	0.09987	0.00427
	600	0.10003	0.00205
	1200	0.09994	0.00139

Table 4.28: Scenario 1: Mean and standard deviation for the estimates of π_k for each k and each N, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.26.

π_k	N	Mean	SD
	12	0.35156	0.12500
	60	0.33841	0.06177
π_1	120	0.33490	0.04423
	600	0.33444	0.01958
	1200	0.33292	0.01328
	12	0.33952	0.13218
	60	0.33333	0.06444
π_2	120	0.33402	0.04170
	600	0.33236	0.01870
	1200	0.33380	0.01333
	12	0.30892	0.12448
	60	0.32826	0.05925
π_3	120	0.33109	0.04288
	600	0.33320	0.01748
	1200	0.33328	0.01330

Table 4.29: Scenario 1: Median absolute deviation for the estimates of ρ_{gk} for each k and each N, using the EM algorithm across the datasets simulated from the settings described in Table 4.26.

	N				
k	12	60	120	600	1200
1	0.112408550	0.045393355	0.031839327	0.014172495	0.009948854
2	0.112812275	0.044446965	0.031538945	0.014278217	0.010051250
3	0.115637525	0.045467278	0.031932817	0.014086308	0.010112260

Table 4.30: Scenario 1: Median absolute deviation for the estimates of β_{0g} for each N, using the EM algorithm across the datasets simulated from the settings described in Table 4.26.

N				
12	60	120	600	1200
0.08286	0.03281	0.02255	0.01017	0.00729

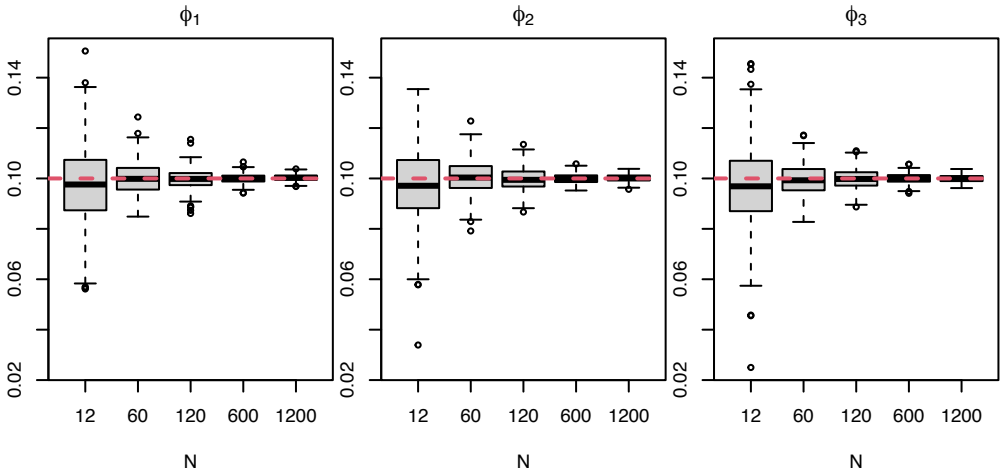

Figure 4.18: Scenario 1: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.26. Red lines correspond to true values. See also Table 4.27.

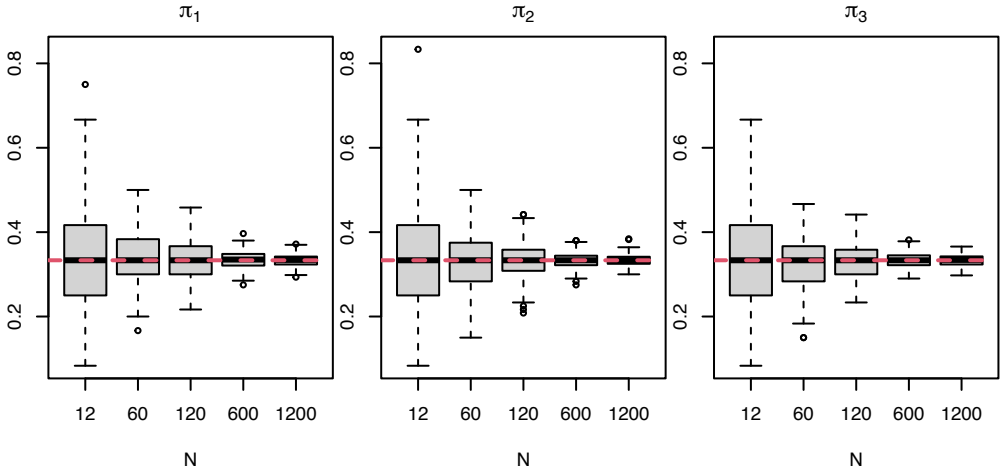

Figure 4.19: Scenario 1: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.26. Red lines correspond to true values. See also Table 4.28.

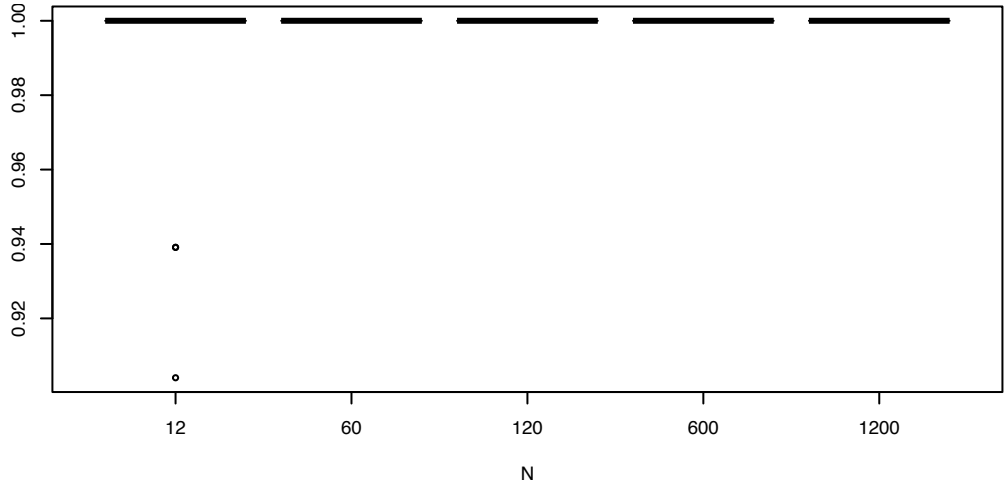

Figure 4.20: Scenario 1: Boxplots for the V-measures of the clustering obtained by the EM algorithm across the datasets simulated from the settings described in Table 4.26.

4.3.2 Scenario 2

For this scenario, the number of genes G varies as G = 12, 60, 120, 600, 1200, 6000 while all other parameters and hyperparameters are kept fixed according to the setting in Table 4.31.

In this scenario, as can be seen in Tables 4.32 and 4.33 and Figures 4.21 and 4.22, all estimates of π_k and ϕ_k are approximately around their true values. Standard deviations decrease as G increases for the probability of always zero (ϕ_k). However, the standard deviations remain almost the same for the cluster assignment probabilities (π_k) as G increases. Again, as expected, the MSEs for ρ_{gk} and β_{0g} remain somewhat the same in all cases (Tables 4.34 and 4.35). According to Table B.15 and Figure B.15, for all cases, the number of iterations is between 7 to 9. The computation times for cases 5 and 6 which have more genes, increase (Table B.16 and Figure B.16). Except for the first case with G = 12, which has some misclassification leading to V-measures slightly less than one, for all other cases, the V-measures are equal to one (Figure 4.23).

Table 4.31: ZIP mixture model with a size factor. Scenario 2: Values chosen for the number of genes G in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1		12			
2		60			
3	1200	120	3	0.1	$1/K$
4		600			
5		1200			
6		6000			

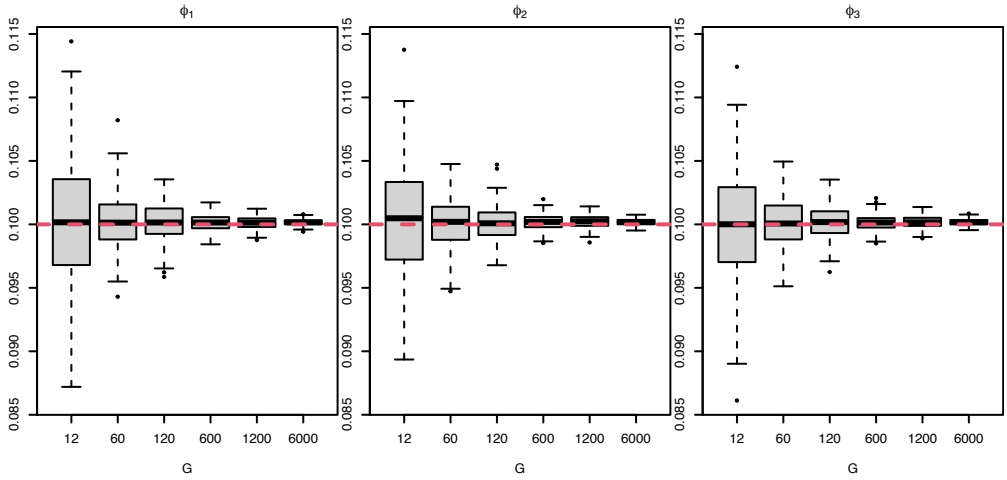

Figure 4.21: Scenario 2: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.31. Red lines correspond to true values. See also Table 4.32.

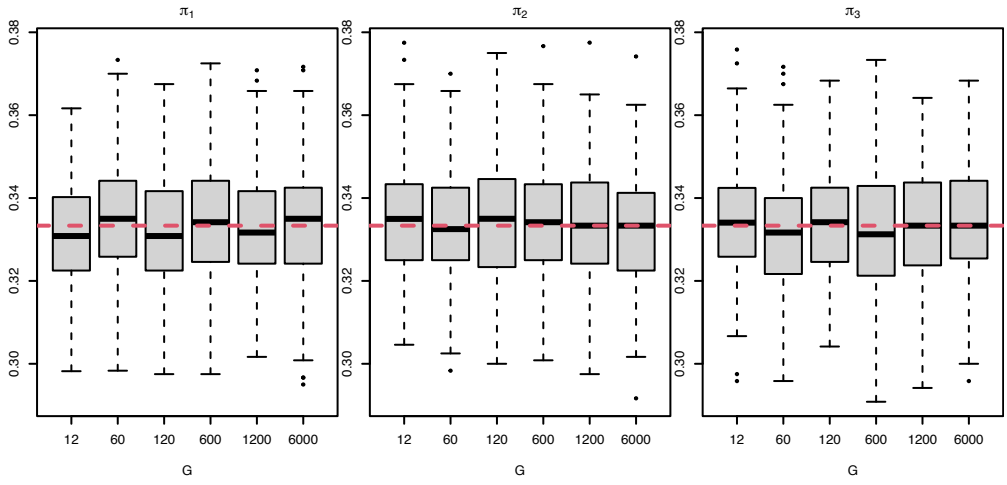

Figure 4.22: Scenario 2: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.31. Red lines correspond to true values. See also Table 4.33.

Table 4.32: Scenario 2: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each G, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.31.

ϕ_k	G	Mean	SD
ϕ_1	12	0.10021	0.00457
	60	0.10027	0.00214
	120	0.10021	0.00141
	600	0.10015	0.00060
	1200	0.10013	0.00047
	6000	0.10016	0.00023
ϕ_2	12	0.10021	0.00422
	60	0.10015	0.00199
	120	0.10005	0.00133
	600	0.10017	0.00061
	1200	0.10022	0.00049
	6000	0.10017	0.00024
ϕ_3	12	0.09993	0.00419
	60	0.10008	0.00187
	120	0.10016	0.00126
	600	0.10014	0.00061
	1200	0.10019	0.00047
	6000	0.10017	0.00024

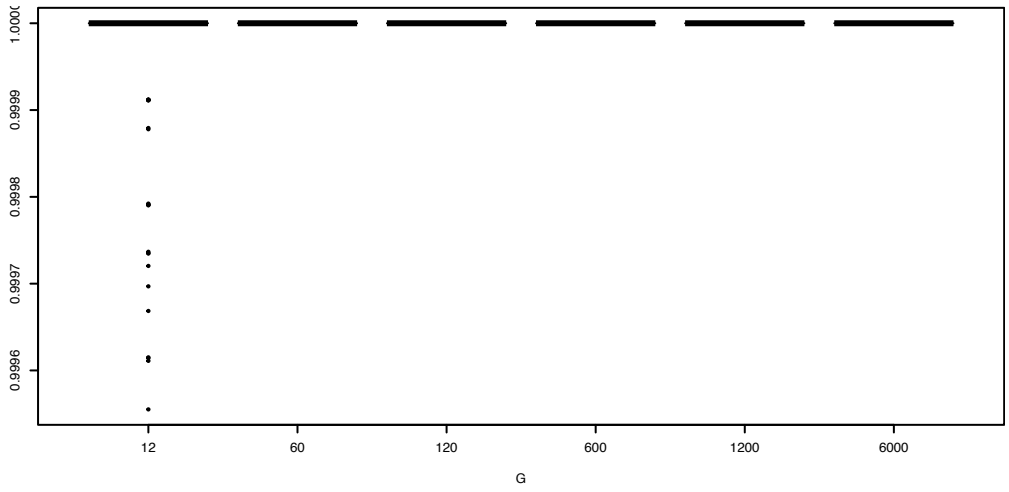

Figure 4.23: Scenario 2: Boxplots for the V-measures of the clustering obtained by the EM algorithm across the datasets simulated from the settings described in Table 4.31.

Table 4.33: Scenario 2: Mean and standard deviation (SD) for the estimates of π_k for each k and each G, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.31.

π_k	G	Mean	SD
π_1	12	0.33115	0.01282
	60	0.33486	0.01351
	120	0.33135	0.01334
	600	0.33437	0.01373
	1200	0.33301	0.01280
	6000	0.33365	0.01426
π_2	12	0.33442	0.01320
	60	0.33349	0.01266
	120	0.33472	0.01439
	600	0.33385	0.01349
	1200	0.33363	0.01346
	6000	0.33223	0.01306
π_3	12	0.33443	0.01311
	60	0.33165	0.01382
	120	0.33393	0.01312
	600	0.33178	0.01462
	1200	0.33336	0.01371
	6000	0.33412	0.01397

Table 4.34: Scenario 2: Mean squared error for the estimates of ρ_{gk} for each k and each G, using the EM algorithm across the datasets simulated from the settings described in Table 4.31.

	G					
k	12	60	120	600	1200	6000
1	0.000242605	0.000238823	0.000236353	0.000234922	0.000235966	0.000236160
2	0.000233509	0.000239676	0.000238435	0.000236828	0.000234413	0.000235708
3	0.000238199	0.000237094	0.000236564	0.000235157	0.000235041	0.000235893

Table 4.35: Scenario 2: Mean squared error for the estimates of β_{0g} for each G, using the EM algorithm across the datasets simulated from the settings described in Table 4.31.

G					
12	60	120	600	1200	6000
0.000115021	0.000114548	0.000112108	0.000113132	0.000113452	0.000114123

4.3.3 Scenario 3

For this scenario, after fixing N and G, we vary K as the number of clusters to evaluate how that affect the EM parameter estimation. For this scenario, the true values of the ρ_{gk}'s used to simulate the data are as follows:

$$K = 1 \rightarrow \rho_{gk}\text{'s} = 0$$

$$K = 2 \rightarrow \rho_{gk}\text{'s} = (-0.6, 0.6)$$

$$K = 3 \rightarrow \rho_{gk}\text{'s} = (-0.6, 0, 0.6)$$

$$K = 5 \rightarrow \rho_{gk}\text{'s} = (-0.8, -0.6, 0, 0.6, 0.8)$$

So that, for example, for $K = 5$ in the above setting, the first $\frac{120}{5} = 24$ genes in cluster $k = 1$ are assigned ρ_{gk} values equal to -0.8, then the following 24 genes are assigned values of -0.6, etc. We repeat this process for the remaining clusters so that each gene's rate parameters are distinct across clusters. Table 4.36 shows the parameter setting for this scenario.

As can be seen in both Tables 4.37, 4.38 and Figures 4.24, 4.25 the estimates of ϕ_k and π_k are very close to their true values. For the estimates of ϕ_k, the standard deviations are lower for small K and become larger when K increases. The standard deviations of the estimates of π_k are almost the same and small as we change K. Note that, in Figure 4.25, the resulted estimates of π_k for different K are all merged into one vector to show their behavior more exactly, which shows the closeness of the estimates to their true values and their stable standard deviations in all cases. Tables 4.39 and 4.40 show that for most cases, the MSEs for the estimates of ρ_{gk} and β_{0g} are small, showing their closeness to their true corresponding values. The average number of iterations for most cases is around 9, with fewer iterations needed when there is no cluster (i.e., $K = 1$, see Table B.17 and Figure B.17). Furthermore, computation time increases as the number of clusters increases (Table B.18 and Figure B.18), which is expected. And finally, all V-measures demonstrate true clustering assignments and are equal to one.

Table 4.36: ZIP mixture model with a size factor. Scenario 3: Values chosen for the number of clusters K in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1			1		
2	1200	120	2	0.1	$1/K$
3			3		
4			5		

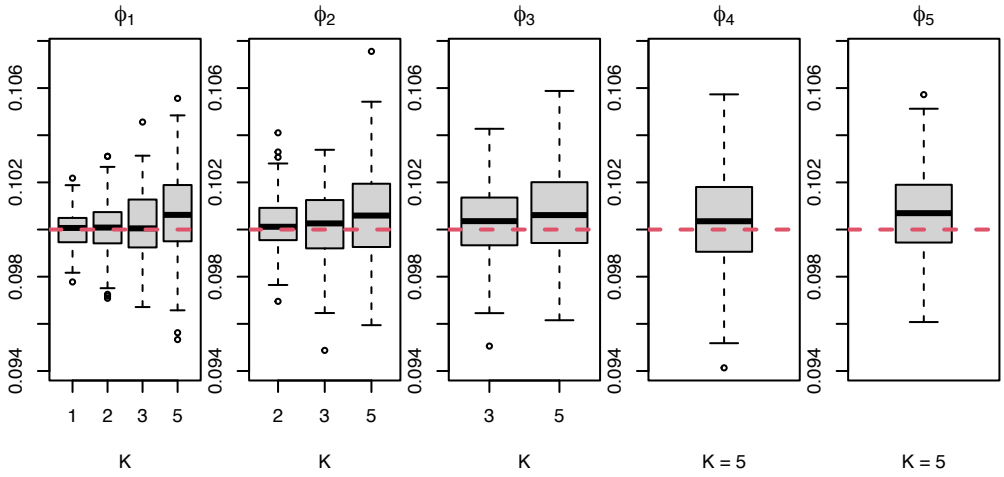

Figure 4.24: Scenario 3: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.36. Red lines correspond to true values. See also Table 4.37.

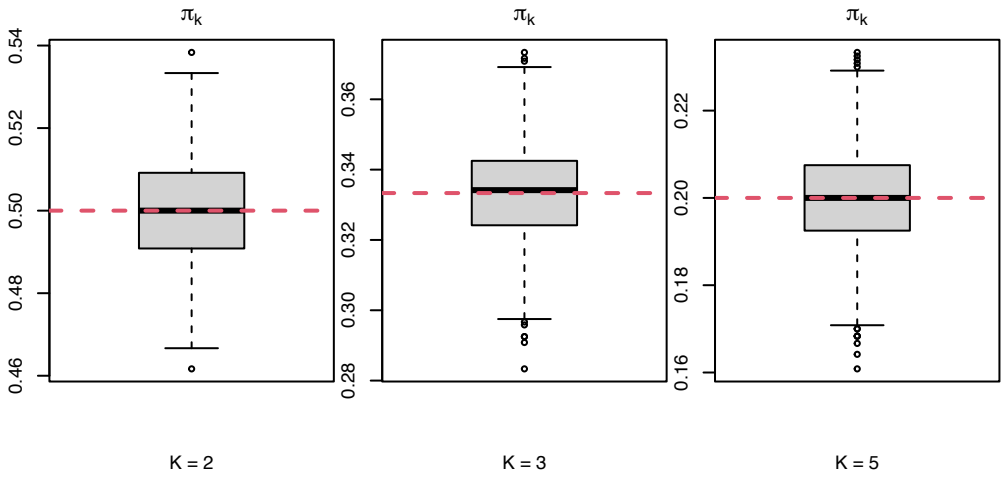

Figure 4.25: Scenario 3: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.36. Red lines correspond to true values. See also Table 4.38. Note that the estimates of π_k over different K are all merged into one vector.

Table 4.37: Scenario 3: Mean and standard deviations for the estimates of ϕ_k for each k and each K, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.36.

ϕ_k	K	Mean	SD
ϕ_1	1	0.09998	0.00074
	2	0.10008	0.00114
	3	0.10016	0.00139
	5	0.10065	0.00178
ϕ_2	2	0.10017	0.00109
	3	0.10024	0.00142
	5	0.10068	0.00191
ϕ_3	3	0.10030	0.00148
	5	0.10072	0.00182
ϕ_4	5	0.10040	0.00192
ϕ_5	5	0.10073	0.00177

Table 4.38: Scenario 3: Mean and standard deviation for the estimates of π_k for each k and each K, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.36.

π_k	K	Mean	SD
π_1	1	1.00000	0.00000
	2	0.50001	0.01348
	3	0.33347	0.01482
	5	0.20009	0.01170
π_2	2	0.49999	0.01348
	3	0.33316	0.01414
	5	0.19911	0.01142
π_3	3	0.33337	0.01443
	5	0.19873	0.01164
π_4	5	0.20121	0.01079
π_5	5	0.20086	0.01070

Table 4.39: Scenario 3: Mean squared error for the estimates of ρ_{gk} for each k and each K, using the EM algorithm across the datasets simulated from the settings described in Table 4.36.

		K		
k	1	2	3	5
1	0	0.000129537	0.000235247	0.000525599
2		0.000129537	0.000238153	0.000529169
3			0.000236647	0.000515542
4				0.000515218
5				0.000519752

Table 4.40: Scenario 3: Mean squared error for the estimates of β_{0g} for each K, using the EM algorithm across the datasets simulated from the settings described in Table 4.36.

K			
1	2	3	5
0.000101369	0.000117110	0.000112684	0.000121417

4.3.4 Scenario 4

In this scenario, we investigate the sensitivity of our proposed EM algorithm to the choice of initial parameter values. We consider two cases and for both of them, we simulate data from the setting in Table 4.41, and the method described in detail in Section 4.3 for the cluster effect parameters. For case 1, we consider the true values as the initial parameter values as in all other simulation scenarios. In case 2, we use the K-means initialization approach to obtain initial clusters. Then, we find the initial parameter values for the obtained clusters as described in the following paragraph.

We find the initial values of the cluster assignment probabilities ($\pi_k^{(0)}$) by calculating the number of cells in each obtained cluster divided by the total number of cells ($N = 1200$). Then, we calculate the proportions of zero in each cluster as initial values for $\phi_k^{(0)}$. Finally, we compute the means over the genes for each inferred cluster as the initial values of the rate parameters ($\lambda_{gk}^{(0)}$).

As can be seen in Tables 4.42 and 4.43 and Figures 4.26 and 4.27 the estimates of ϕ_k's and π_k's are centered at their true values in both cases. However, there are some outliers in case 2 with larger standard deviations. According to Tables 4.44 and 4.45, the estimates for the parameters β_{0g} and ρ_{gk} all have reasonably small MADs in both cases, although higher values for case 2. Furthermore, most of the time the V-measures are approximately one (all greater than 0.95), except for some misclassification that occurred in case 2 (Figure 4.28). In most simulations, the number of iterations is almost the same, and not too many iterations are needed for convergence, but there were some cases with more iterations for case 2 (Table B.19 and Figure B.19). Finally, the computation time was considerably longer for case 2 (Table B.20 and Figure B.20)

Table 4.41: ZIP mixture model with a size factor. Scenario 4: Values chosen for the fixed parameters used to simulate the datasets.

N	G	K	ϕ_k	π_k
1200	120	3	0.1	$1/K$

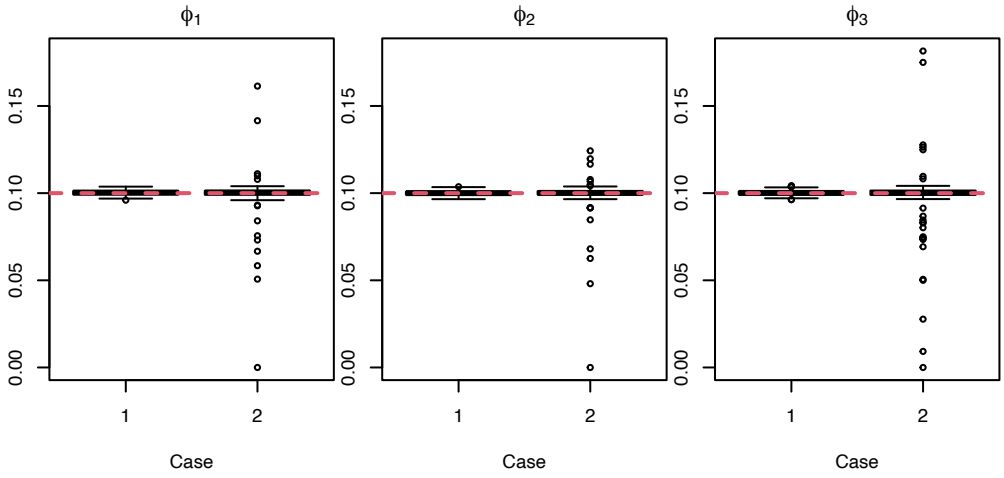

Figure 4.26: Scenario 4: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.41. Red lines correspond to true values. See also Table 4.42.

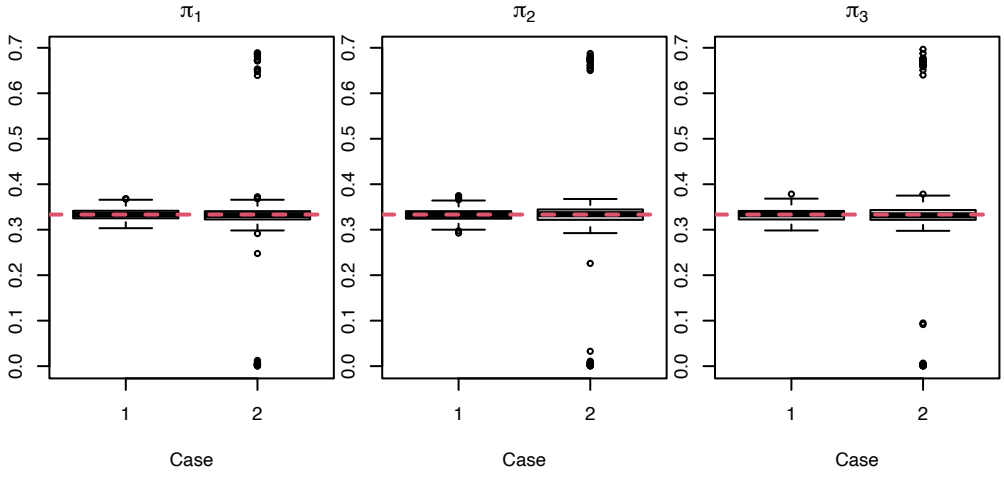

Figure 4.27: Scenario 4: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.41. Red lines correspond to true values. See also Table 4.43.

Table 4.42: Scenario 4: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.41.

ϕ_k	Case	Mean	SD
ϕ_1	1	0.10023	0.00142
	2	0.09962	0.00956
ϕ_2	1	0.10003	0.00140
	2	0.09946	0.00825
ϕ_3	1	0.10017	0.00140
	2	0.09899	0.01376

Table 4.43: Scenario 4: Mean and standard deviation (SD) for the estimates of π_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.41.

π_k	Case	Mean	SD
π_1	1	0.33444	0.01317
	2	0.33124	0.11683
π_2	1	0.33255	0.01399
	2	0.33474	0.12418
π_3	1	0.33301	0.01327
	2	0.33403	0.14037

Table 4.44: Scenario 4: Median absolute deviation for the estimates of ρ_{gk} for each k and each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.41.

	Case	
k	1	2
1	0.01026	0.59934
2	0.01005	0.59589
3	0.01009	0.60201

Table 4.45: Scenario 4: Median absolute deviation for the estimates of β_{0g} for each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.41.

Case	
1	2
0.00707	0.00949

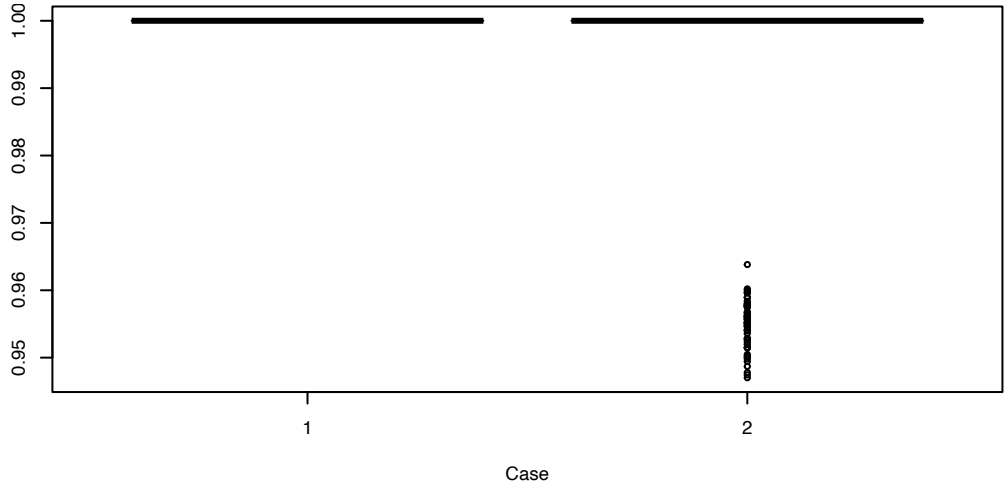

Figure 4.28: Scenario 4: Boxplots for the V-measures of the clustering obtained by the EM algorithm across the datasets simulated from the settings described in Table 4.41.

4.3.5 Scenario 5

For this scenario, we use $K = 2$ clusters, and rather than using the equal cluster assignment probabilities, we use different values for these probabilities. The setting for this scenario is shown in Table 4.46.

As demonstrated in Tables 4.47 and 4.48 and Figures 4.29 and 4.30 the estimated probabilities of ϕ_k and π_k are close to their true values. From Table 4.47 and Figure 4.29, for the probabilities of always zero estimates (ϕ_k), we can see that the standard deviations for the first cluster increase in each case, but remain almost the same for the second cluster. For the estimates of π_k, the standard deviations are almost the same and small enough in all cases (Table 4.48 and Figure 4.30). According to the MSE results (see Tables 4.49 and 4.50), the estimates of ρ_{gk} and β_{0g} are close to their true values, and when the clusters are more unbalanced, the MSEs become a bit larger for both estimates. Also, the V-measures in all the cases are one or very close to one. The number of iterations is equal to 9 for all cases (Table B.21 and Figure B.21), and the computation time is the same for all cases (Table B.22 and Figure B.22)

Table 4.46: ZIP mixture model with a size factor. Scenario 5: Values chosen for the proportion assigned to each cluster π_k in each case along with the fixed parameters used to simulate the datasets.
Note that $\pi_2 = 1 - \pi_1$.

Case	N	G	K	ϕ_k	π_1	π_2
1					0.50	0.50
2	1200	120	2	0.1	0.25	0.75
3					0.10	0.90
4					0.05	0.95

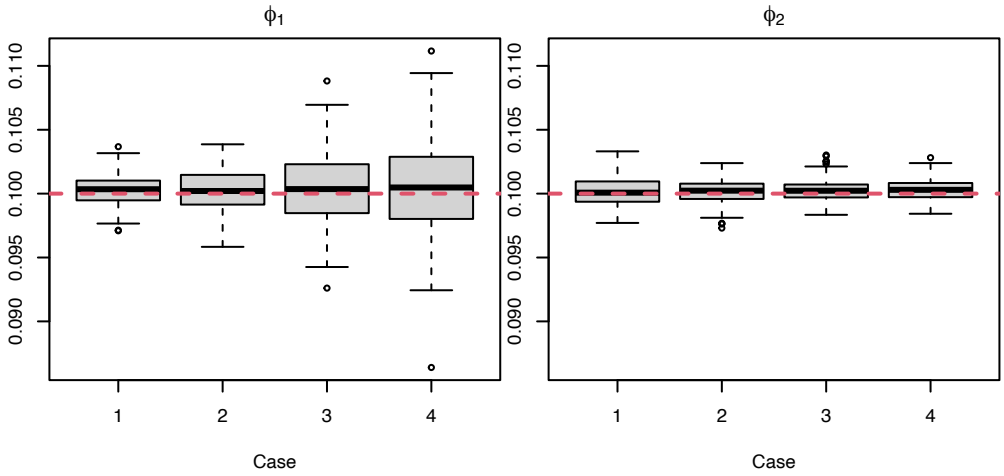

Figure 4.29: Scenario 5: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.46. Red lines correspond to true values. See also Table 4.47.

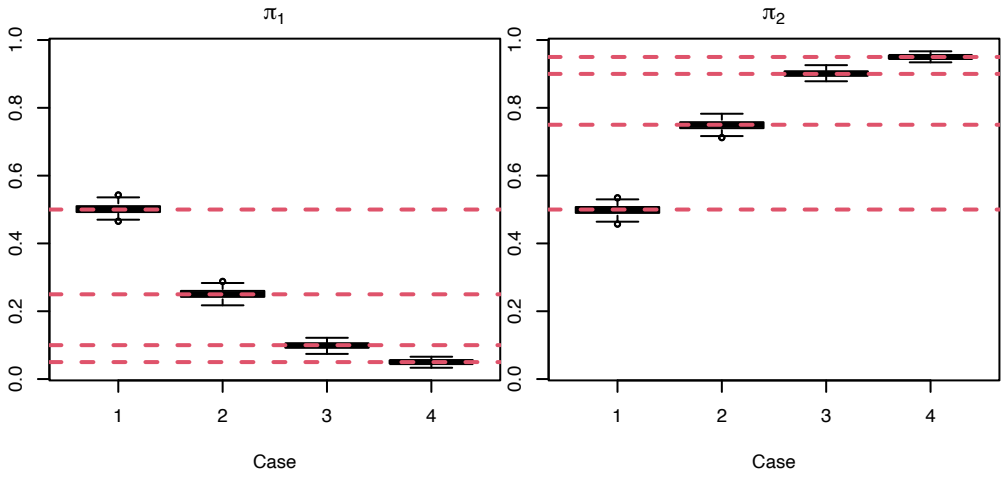

Figure 4.30: Scenario 5: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.46. Red lines correspond to true values. See also Table 4.48.

Table 4.47: Scenario 5: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.46.

ϕ_k	Case	Mean	SD
ϕ_1	1	0.10027	0.00116
	2	0.10024	0.00158
	3	0.10029	0.00271
	4	0.10041	0.00342
ϕ_2	1	0.10015	0.00110
	2	0.10019	0.00094
	3	0.10023	0.00085
	4	0.10031	0.00082

Table 4.48: Scenario 5: Mean and standard deviation (SD) for the estimates of π_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.46.

π_k	Case	Mean	SD
π_1	1	0.50042	0.01393
	2	0.25139	0.01278
	3	0.09914	0.00900
	4	0.05029	0.00618
π_2	1	0.49958	0.01393
	2	0.74861	0.01278
	3	0.90086	0.00900
	4	0.94971	0.00618

Table 4.49: Scenario 5: Mean squared error for the estimates of ρ_{gk} for each k and each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.46.

	Case			
k	1	2	3	4
1	0.000128601	0.000168903	0.000335283	0.000650272
2	0.000128601	0.000168903	0.000335283	0.000650272

Table 4.50: Scenario 5: Mean squared error for the estimates of β_{0g} for each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.46.

Case			
1	2	3	4
0.00012	0.00016	0.00032	0.00063

4.3.6 Scenario 6

For this scenario, the probability of always zero varies between different cases to see their effects on parameter estimation. We start from a very small probability of always zero (5%) to a large one (90%) across $K = 3$ clusters. All other parameters and hyperparameters are kept fixed and the setting for this scenario can be seen in Table 4.51.

According to Tables 4.52 and 4.53 and Figures 4.31 and 4.32, the estimates of π_k and ϕ_k are all close to their true values and have almost the same standard deviations for all cases. The only thing is that the standard deviations have a little increase in some cases when there is a larger probability of always zero. We should note that in Figure 4.31 for cases 1 to 5, the estimates of ϕ_1, ϕ_2, and ϕ_3 are all merged into one vector to demonstrate their behaviour better. Again, the V-measures are mostly close to one except for some misclassifications in case 5 (Figure 4.33). Also, the estimates of the parameters β_{0g} and ρ_{gk} have small MSEs that show their proximity to the true values (Tables 4.54 and 4.55). The number of iterations is almost the same, except for case 5, which has always zero proportions of 90% (Table B.23 and Figure B.23). The computational time is almost the same across the cases except for case 4, which is faster compared to the other cases (Table B.24 and Figure B.24).

Table 4.51: ZIP mixture model with a size factor. Scenario 6: Values chosen for the probability of always-zero in the ZIP distribution ϕ_k in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1				0.05	
2				0.10	
3	1200	120	3	0.25	$1/K$
4				0.50	
5				0.90	
6				(0.05, 0.25, 0.50)	

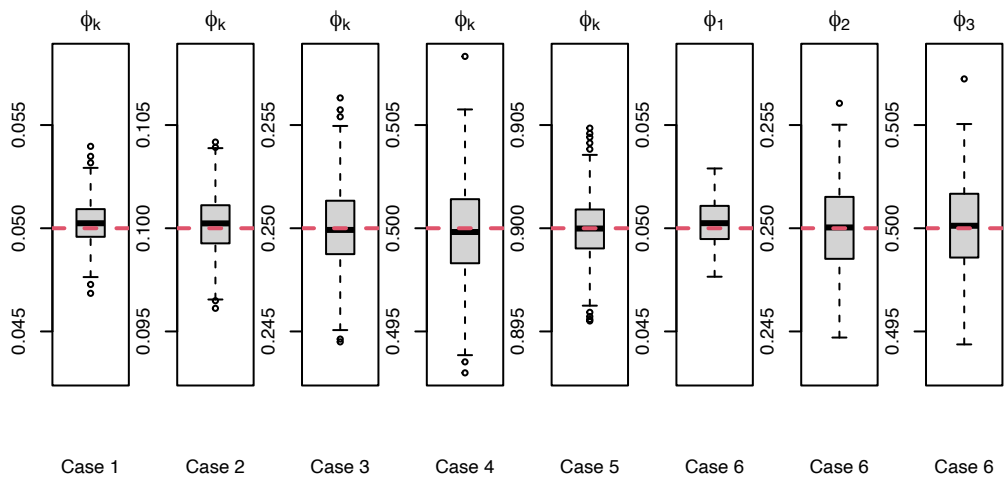

Figure 4.31: Scenario 6: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.51. Red lines correspond to true values. See also Table 4.52. Note that the estimates of ϕ_1, ϕ_2, and ϕ_3 are all merged into one vector for cases 1-5.

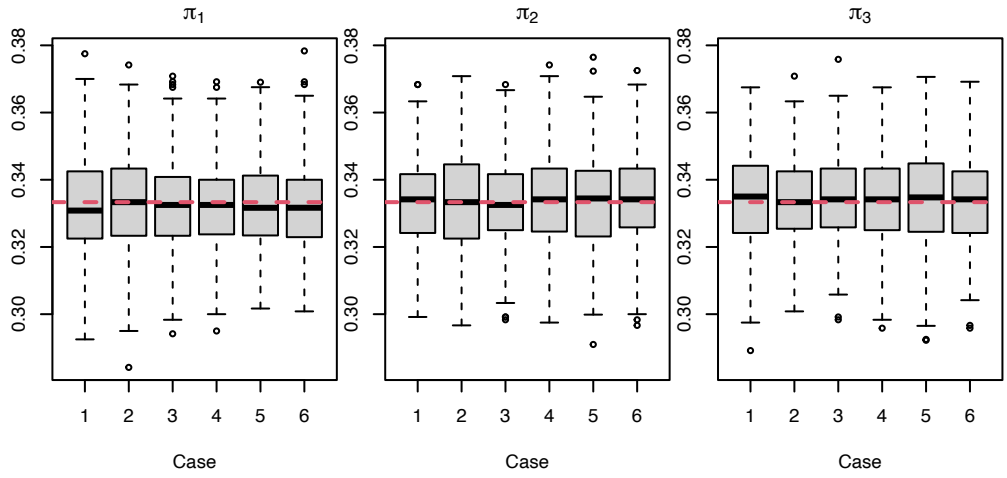

Figure 4.32: Scenario 6: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.51. Red lines correspond to true values. See also Table 4.53.

Table 4.52: Scenario 6: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.51.

ϕ_k	Case	Mean	SD
ϕ_1	1	0.05023	0.00098
	2	0.10026	0.00140
	3	0.25009	0.00189
	4	0.49992	0.00229
	5	0.89997	0.00148
	6	0.05027	0.00107
ϕ_2	1	0.05030	0.00100
	2	0.10006	0.00139
	3	0.25006	0.00190
	4	0.49978	0.00225
	5	0.90002	0.00145
	6	0.25001	0.00213
ϕ_3	1	0.05024	0.00092
	2	0.10024	0.00138
	3	0.24992	0.00189
	4	0.49989	0.00218
	5	0.89991	0.00131
	6	0.50008	0.00220

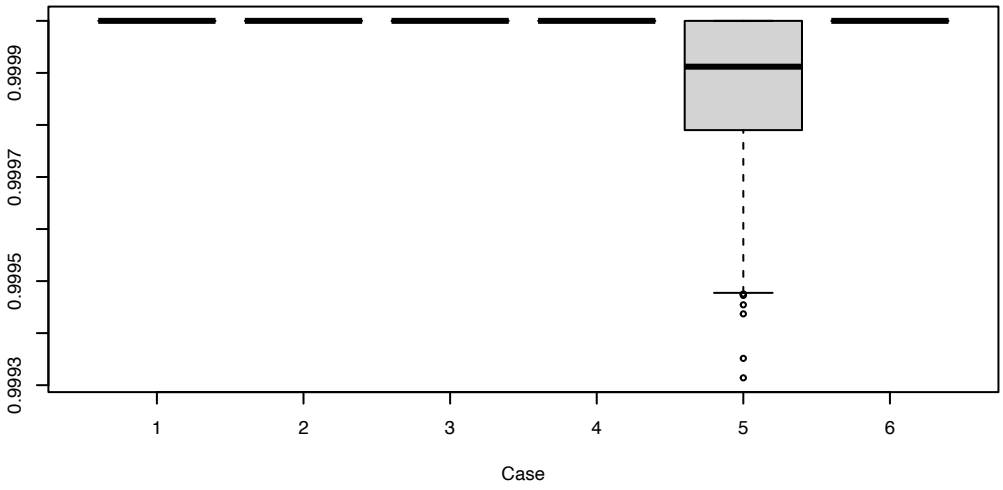

Figure 4.33: Scenario 6: Boxplots for the V-measures of the clustering obtained by the EM algorithm across the datasets simulated from the settings described in Table 4.51.

Table 4.53: Scenario 6: Mean and standard deviation (SD) for the estimates of π_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.51.

π_k	Case	Mean	SD
π_1	1	0.33265	0.01449
	2	0.33349	0.01426
	3	0.33261	0.01366
	4	0.33217	0.01318
	5	0.33227	0.01349
	6	0.33193	0.01300
π_2	1	0.33333	0.01331
	2	0.33320	0.01513
	3	0.33305	0.01295
	4	0.33402	0.01395
	5	0.33364	0.01422
	6	0.33457	0.01296
π_3	1	0.33401	0.01417
	2	0.33331	0.01346
	3	0.33434	0.01354
	4	0.33381	0.01388
	5	0.33409	0.01511
	6	0.33351	0.01314

Table 4.54: Scenario 6: Mean squared error for the estimates of ρ_{gk} for each k and each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.51.

	Case					
k	1	2	3	4	5	6
1	0.000226465	0.000237206	0.000281781	0.000417185	0.002150195	0.000264188
2	0.000228237	0.000235153	0.000278374	0.000419742	0.002098454	0.000294226
3	0.000221568	0.000234383	0.000277740	0.000411982	0.002113181	0.000357756

Table 4.55: Scenario 6: Mean squared error for the estimates of β_{0g} for each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.51.

Case					
1	2	3	4	5	6
0.00011	0.00011	0.00013	0.00020	0.00106	0.00015

4.4 Simulation scenarios for the ZIP mixture model with a covariate

In this section, we simulate data from the zero-inflated Poisson mixture model (as described in Section 3.5.2) when we have a cell-specific size factor (T_n), the cluster effects (ρ_{gk}), the baseline expression (β_{0g}), and include a covariate x_{n1}, for $n = 1, \ldots, N$. For this model, we simulate x_{n1} from a Bernoulli distribution with a 0.5 probability of success. We consider two scenarios. In scenario 1, we vary N (the number of cells) while fixing G (the number of genes) and all other parameters and hyperparameters. For scenario 2, we vary G, while N and all other parameters and hyperparametes are kept fixed. We simulate $S = 100$ datasets for all cases in each scenario. The parameters that we use to simulate data include:

- Number of clusters: $K = 2$.
- Probability of cluster assignments: $\pi_1 = \pi_2 = 0.5$.
- Probability of always zero in each cluster: $\phi_1 = \phi_2 = 0.1$.
- Baseline expression: $\beta_{0g} = 0.85$ for all g.
- Cluster effects: ρ_{g1}'s = $(2, -2)$ and ρ_{g2}'s = $(-2, 2)$, in which, for cluster one, the cluster effects on the first half of the genes are 2 and for the remaining half of the genes they are -2 and vice versa for the second cluster.
- Covariate coefficients: $\beta_{1g} = 1$ for the first half of the genes, and for the remaining half of the genes, $\beta_{1g} = 0.5$.
- Size factors: T_n's are simulated from a normal distribution with $\mu = 10$ and $\sigma = 0.5$.

4.4.1 Scenario 1

As mentioned in the previous paragraph, for this scenario, we consider different N values while fixing all other parameters and hyperparameters. Table 4.56 demonstrates the setting for this scenario.

For the probabilities of cluster assignments (π_k) and the probabilities of always zero (ϕ_k), respectively, as can be seen from Tables 4.57 and 4.58 and Figures 4.34 and 4.35, the resulting estimates get closer to their true values and as N increases and the standard deviation of these estimates decreases as N increases, as expected. We also calculate the MADs of the other parameters, including β_{0g}, β_{1g}, and ρ_{gk} across all genes and all the simulated datasets. As shown in Tables 4.59 and 4.60, the MADs are small, which shows the closeness of these estimates to

Table 4.56: ZIP mixture model with a covariate. Scenario 1: Values chosen for the number of observations N in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1	60				
2	120				
3	300	120	2	0.1	$1/K$
4	600				
5	1200				

their true values. In addition, as N increases, the MAD values decrease, as expected by the convergence properties of the EM algorithm. The computing time increases when N increases (Table B.25 and Figure B.25) and V-measures are all one, which shows a perfect match between inferred and true cluster assignments.

Table 4.57: Scenario 1: Mean and standard deviation (SD) for the estimates of π_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.56.

k	N	Mean	SD
1	60	0.491500000	0.065798002
	120	0.498166667	0.049585544
	300	0.499666667	0.029188424
	600	0.501066667	0.019969843
	1200	0.497058333	0.013312513
2	60	0.508500000	0.065798002
	120	0.501833333	0.049585544
	300	0.500333333	0.029188424
	600	0.498933333	0.019969843
	1200	0.502941667	0.013312513

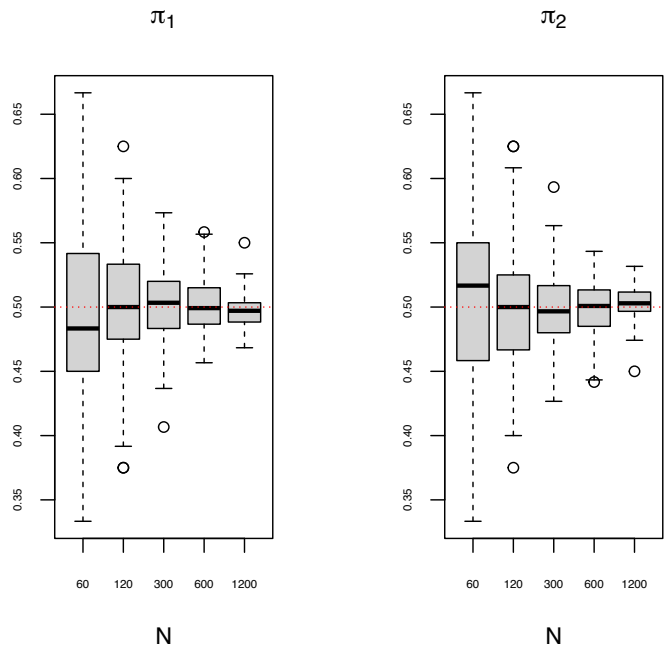

Figure 4.34: Scenario 1: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.56. Red lines correspond to true values. See also Table 4.57.

Table 4.58: Scenario 1: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.56.

k	N	Mean	SD
1	60	0.098992456	0.005007809
	120	0.099832251	0.003744747
	300	0.099809984	0.002182312
	600	0.099899911	0.001649528
	1200	0.099991528	0.001203047
2	60	0.100149594	0.005226926
	120	0.099848961	0.003716596
	300	0.100306134	0.002214400
	600	0.100145810	0.001825143
	1200	0.100001481	0.001173997

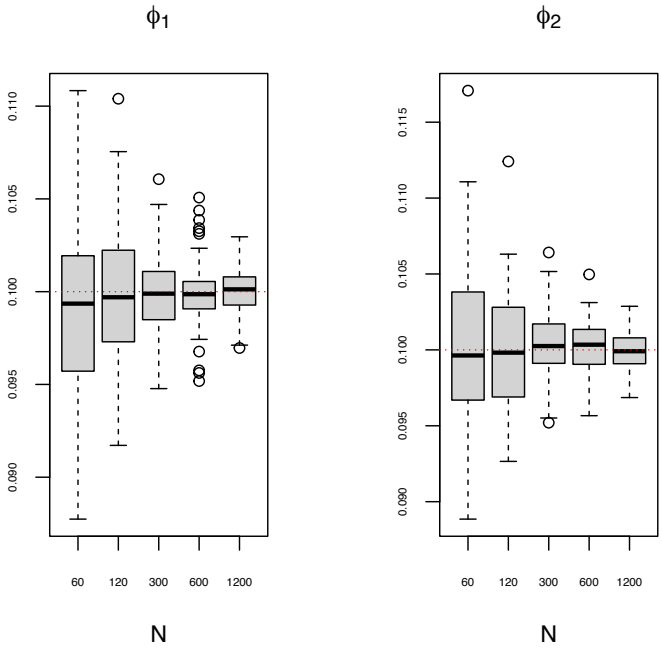

Figure 4.35: Scenario 1: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.56. Red lines correspond to true values. See also Table 4.58.

Table 4.59: Scenario 1: MAD for the estimates of ρ_{gk} for each k and each N, using the EM algorithm across the datasets simulated from the settings described in Table 4.56.

	N				
k	60	120	300	600	1200
1	0.038813812	0.026580682	0.016910423	0.011647520	0.008494867
2	0.035358002	0.024422462	0.015716372	0.010928285	0.007832469

Table 4.60: Scenario 1: MAD for the estimates of β_{0g} and β_{1g} for each N, using the EM algorithm across the datasets simulated from the settings described in Table 4.56.

	N				
	60	120	300	600	1200
β_{0g}	0.014636945	0.010251386	0.006453585	0.004745225	0.003249015
β_{1g}	0.016953099	0.011994497	0.007652635	0.005360614	0.003791324

4.4.2 Scenario 2

For this scenario, by fixing $N = 1200$ and all other parameters and hyperparameters, the number of genes varies according to the settings in Table 4.61.

Table 4.61: ZIP mixture model with a covariate. Scenario 2: Values chosen for the number of genes G in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1		12			
2		60			
3	1200	120	2	0.1	$1/K$
4		600			
5		1500			

In this scenario, the estimates of π_k's (probability of cluster assignments), are close to their true values, but their standard deviations remain somewhat the same while G increases (Table 4.62 and Figure 4.36). For the probability of always zero, we can observe from Table 4.63 and Figure 4.37 that the resulting estimates are close to their true values and the standard deviations decrease as G increases. For other parameter estimates, including β_{0g}, β_{1g}, and ρ_{gk} from Tables 4.64 and 4.65, the MADs remain almost the same when the number of genes increases as N remains fixed. The V-measures are equal to one in all cases. As expected, the computing time increases when G increases, particularly for the largest one when $G = 1500$ (Table B.26 and Figure B.26).

Table 4.62: Scenario 2: Mean and standard deviation (SD) for the estimates of π_k for each k and each G, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.61.

k	G	Mean	SD
1	12	0.499675000	0.014990127
	60	0.499125000	0.012900908
	120	0.499216667	0.014080407
	600	0.503308333	0.014740771
	1500	0.499600000	0.014515405
2	12	0.500325000	0.014990127
	60	0.500875000	0.012900908
	120	0.500783333	0.014080407
	600	0.496691667	0.014740771
	1500	0.500400000	0.014515405

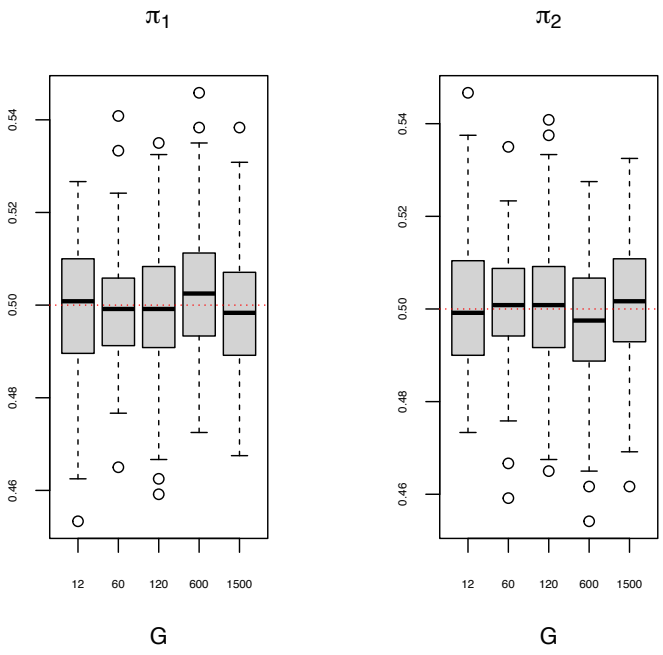

Figure 4.36: Scenario 2: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.61. Red lines correspond to true values. See also Table 4.62.

Table 4.63: Scenario 2: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.61.

k	G	Mean	SD
1	12	0.100470497	0.003632417
	60	0.099988024	0.001694311
	120	0.099896435	0.001094482
	600	0.100007377	0.000554468
	1500	0.099973886	0.000331099
2	12	0.099665322	0.004176079
	60	0.099985994	0.001601331
	120	0.100113810	0.001070322
	600	0.100034844	0.000530891
	1500	0.099958600	0.000329253

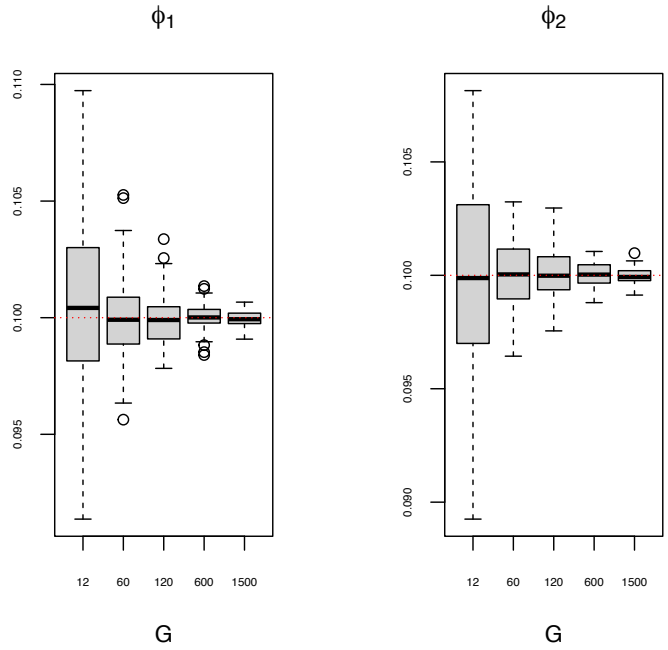

Figure 4.37: Scenario 2: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.61. Red lines correspond to true values. See also Table 4.63.

Table 4.64: Scenario 2: MAD for the estimates of ρ_{gk} for each k and each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.61.

	G				
k	12	60	120	600	1500
1	0.008111007	0.008324391	0.008290203	0.008281629	0.008424692
2	0.008118105	0.007922273	0.007625036	0.007775817	0.007789265

Table 4.65: Scenario 2: MAD for the estimates of β_{0g} and β_{1g} for each G, using the EM algorithm across the datasets simulated from the settings described in Table 4.61.

	G				
	12	60	120	600	1500
β_{0g}	0.003424299	0.003184400	0.003212991	0.003321381	0.003312820
β_{1g}	0.003699243	0.003602925	0.003818405	0.003781053	0.003764123

4.5 Simulation scenarios for the ZINB mixture model without covariates

In this section, we simulate data from the zero-inflated negative binomial mixture model without covariates (Section 3.6.1). We consider two scenarios, where the number of cells (N) varies in scenario one, and in the other scenario, the number of genes (G) varies, while holding all other parameters and hyperparameters fixed. For both scenarios, we simulate data from $K = 2$ clusters with equal probability of cluster assignments ($\pi_1 = \pi_2 = 0.5$). The probabilities of always zero are equal to 0.1 for both clusters ($\phi_1 = \phi_2 = 0.1$). The size parameters for the negative binomial components are $\nu_1 = 5$ and $\nu_2 = 20$. For the negative binomial rate parameters, we considered $\mu_{g1} = \mu_1 = 5$ and $\mu_{g2} = \mu_2 = 10$ for all g.

4.5.1 Scenario 1

As mentioned above, for this scenario, the number of cells (N) varies while we fix $G = 120$ and all other parameters and hyperparameters according to the setting in Table 4.66.

Table 4.66: ZINB mixture model without covariates. Scenario 1: Values chosen for the number of observations N in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1	60				
2	120				
3	300	120	2	0.1	$1/K$
4	600				
5	1200				

We can observe from Tables 4.67 and 4.68 and Figures 4.38 and 4.39 that the estimates of π_k and ϕ_k are close to their true values, and as N increases, the standard deviations decrease. The MSEs for the estimates of the rate parameters decrease as N increases (see Table 4.69). Furthermore, as shown in Tables 4.70 and Figure 4.40, for both clusters, the bias and the variance in the estimation of the size parameter decrease as N increases. Finally, computing time increases when N increases (Table B.27 and Figure B.27), and V-measures for all data sets are equal to one.

Table 4.67: Scenario 1: Mean and standard deviation (SD) for the estimates of π_k for each cluster k and each N, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.66.

k	N	Mean	SD
1	60	0.502500000	0.063712977
	120	0.499833333	0.045643239
	300	0.499433333	0.027784060
	600	0.499966667	0.020996472
	1200	0.503100000	0.014755813
2	60	0.497500000	0.063712977
	120	0.500166667	0.045643239
	300	0.500566667	0.027784060
	600	0.500033333	0.020996472
	1200	0.496900000	0.014755813

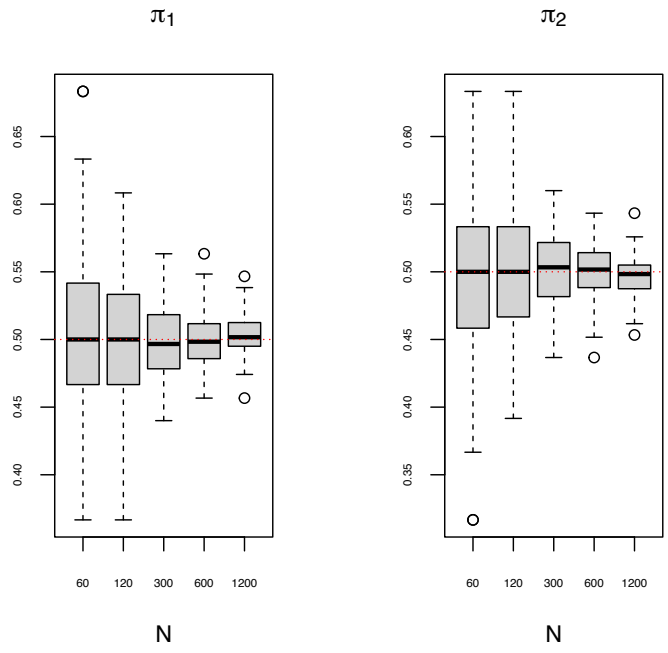

Figure 4.38: Scenario 1: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.66. Red lines correspond to true values. See also Table 4.67.

Table 4.68: Scenario 1: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.66.

k	N	Mean	SD
1	60	0.101422431	0.006166323
	120	0.100564181	0.004162313
	300	0.100183877	0.002496634
	600	0.099694053	0.002046551
	1200	0.100138043	0.001312946
2	60	0.099817364	0.005766071
	120	0.100215019	0.003390551
	300	0.099439385	0.002471910
	600	0.100079498	0.001403623
	1200	0.099994556	0.001015787

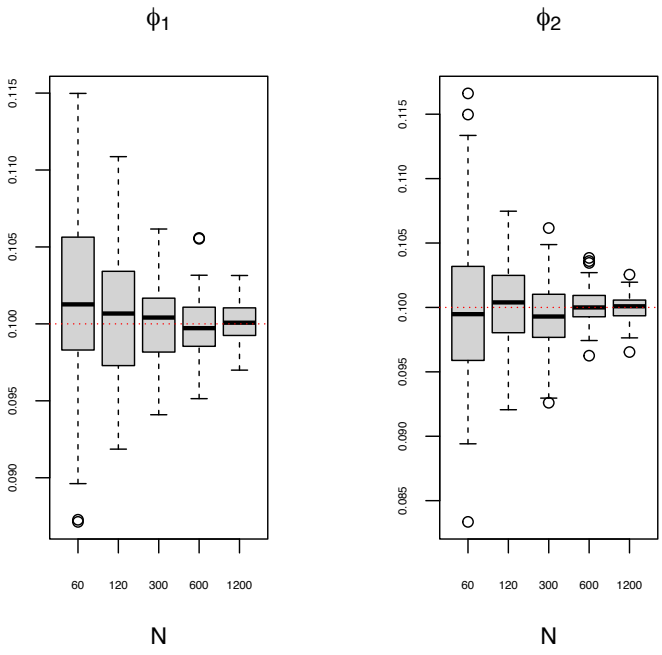

Figure 4.39: Scenario 1: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.66. Red lines correspond to true values. See also Table 4.68.

Table 4.69: Scenario 1: Mean squared error for the estimates of μ_{gk} for each k and each case, using the EM algorithm across the datasets simulated from the settings described in Table 4.66.

	N				
k	60	120	300	600	1200
1	0.40832	0.19947	0.08136	0.03927	0.01948
2	0.57480	0.27671	0.11284	0.05625	0.02844

Table 4.70: Scenario 1: Mean and standard deviation (SD) for the estimates of ν_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.66.

k	N	Mean	SD
1	60	5.48	0.38
	120	5.26	0.23
	300	5.10	0.12
	600	5.04	0.11
	1200	5.01	0.07
2	60	23.07	2.04
	120	21.29	1.20
	300	20.36	0.74
	600	20.21	0.52
	1200	20.11	0.33

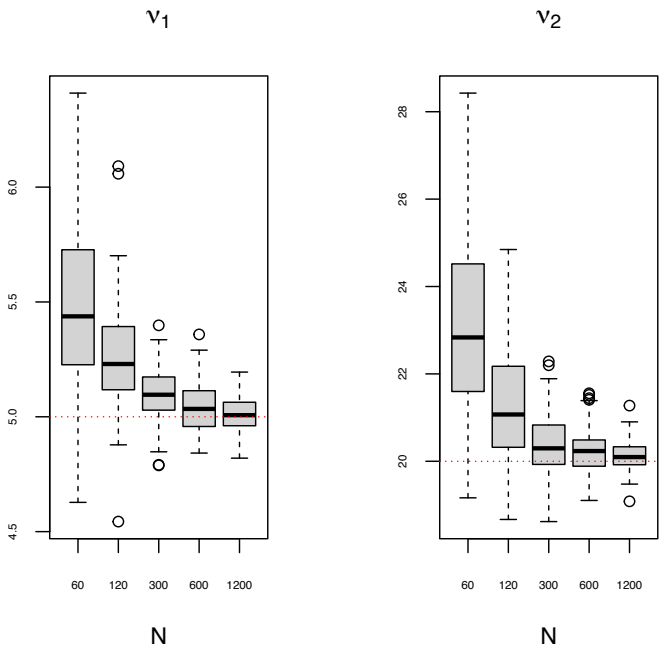

Figure 4.40: Scenario 1: Boxplots for the estimates of ν_k using the EM algorithm across the datasets simulated from the settings described in Table 4.66. Red lines correspond to true values. See also Table 4.70.

4.5.2 Scenario 2

For this scenario, by fixing $N = 1200$ and all other parameters and hyperparameters, the number of genes (G) varies between the simulated data according to the setting in Table 4.71.

Table 4.71: ZINB mixture model without covariates. Scenario 2: Values chosen for the number of genes G in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1		12			
2		60			
3	1200	120	2	0.1	$1/K$
4		600			
5		1500			

We can observe from Tables 4.72 and 4.73 and Figures 4.41 and 4.42 that for both parameters π_k and ϕ_k, their estimates are close to their true values. The standard deviation of the parameters estimates ϕ_k decreases as G increases; however, the standard deviations for the estimates of π_k remain almost the same with varying G. As Table 4.74 shows, no significant changes are observed on the MSEs of the parameter estimates of μ_{gk} for each cluster when G varies. Furthermore, from Table 4.75 and Figure 4.43, we can see that the size parameters are estimated close to their true values with a decrease in their standard deviations as G increases. According to Table B.28 and Figure B.28 the computing time increases as G increases. Finally, except for the simulated data sets when $G = 12$ with some misclassifications, the V-measures are very close to one for all the other cases (see Figure 4.44).

Table 4.72: Scenario 2: Mean and standard deviation (SD) for the estimates of π_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.71.

k	G	Mean	SD
1	12	0.499719868	0.017829928
	60	0.501075000	0.014799110
	120	0.501675000	0.016050257
	600	0.497741667	0.014496940
	1500	0.500300000	0.012287772
2	12	0.500280132	0.017829928
	60	0.498925000	0.014799110
	120	0.498325000	0.016050257
	600	0.502258333	0.014496940
	1500	0.499700000	0.012287772

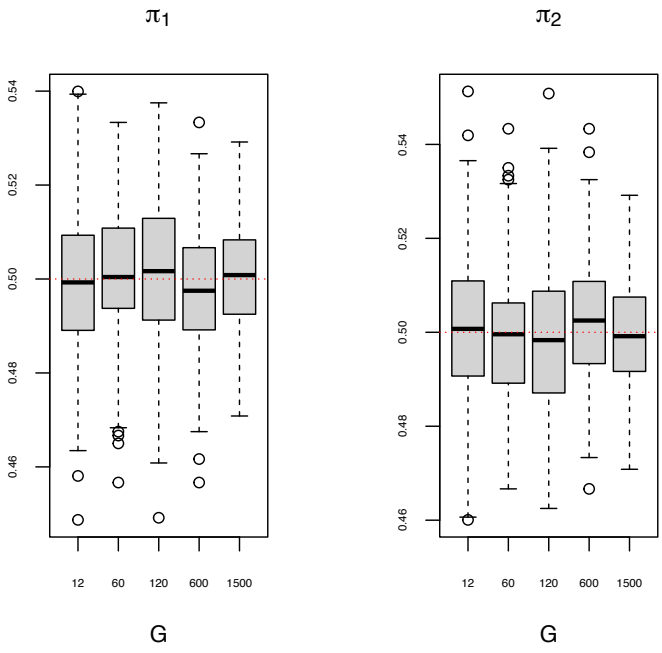

Figure 4.41: Scenario 2: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.71. Red lines correspond to true values. See also Table 4.72.

Table 4.73: Scenario 2: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.71.

k	G	Mean	SD
1	12	0.099852809	0.004301919
	60	0.100302132	0.002016654
	120	0.099789100	0.001318137
	600	0.100000309	0.000585295
	1500	0.100004958	0.000351525
2	12	0.100203671	0.003345701
	60	0.099927764	0.001476867
	120	0.100218634	0.001022206
	600	0.100067091	0.000544676
	1500	0.099958123	0.000300695

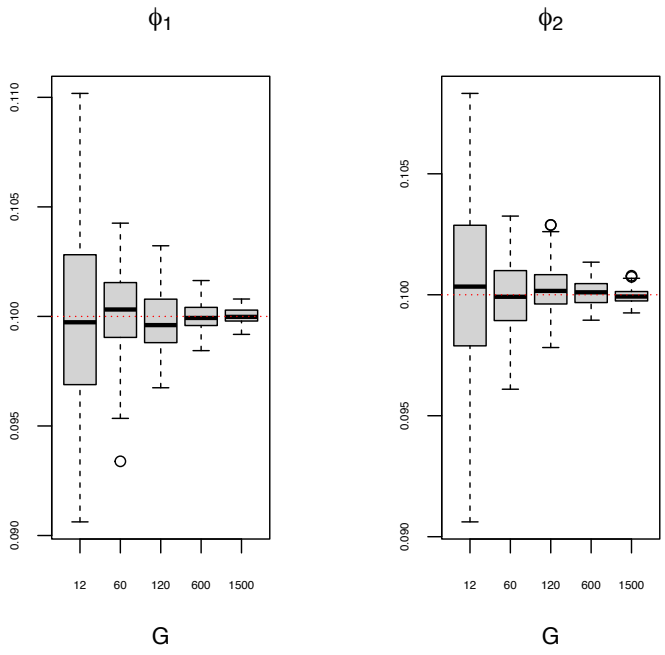

Figure 4.42: Scenario 2: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.71. Red lines correspond to true values. See also Table 4.73.

Table 4.74: Scenario 2: Mean squared error for the estimates of μ_{gk} for each k and each G, using the EM algorithm across the datasets simulated from the settings described in Table 4.71.

	G				
k	12	60	120	600	1500
1	0.02049	0.01917	0.01959	0.01990	0.01976
2	0.02816	0.02798	0.02851	0.02761	0.02783

Table 4.75: Scenario 2: Mean and standard deviation (SD) for the estimates of ν_k for each k and each case, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.71.

k	G	Mean	SD
1	12	4.9981	0.1790
	60	5.0416	0.0816
	120	5.0122	0.0732
	600	5.0186	0.0281
	1500	5.0218	0.0222
2	12	19.9360	1.0668
	60	20.0507	0.4643
	120	20.1685	0.3419
	600	20.1155	0.1467
	1500	20.1211	0.0917

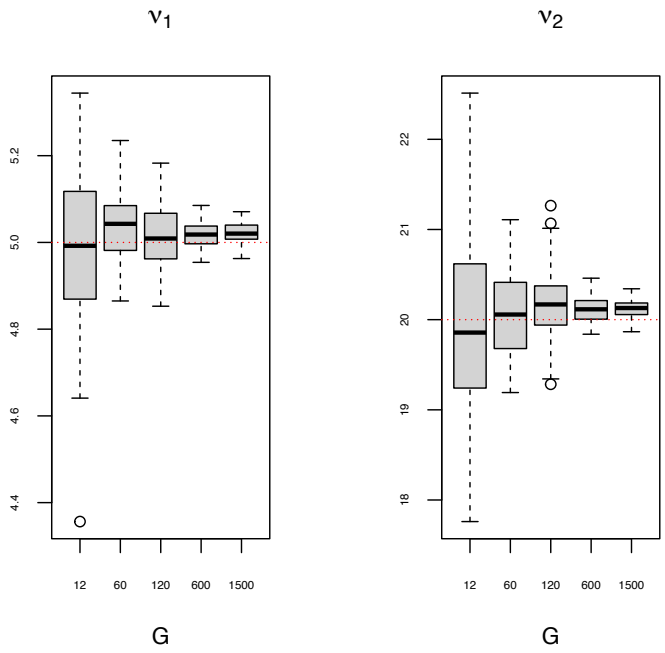

Figure 4.43: Scenario 2: Boxplots for the estimates of ν_k using the EM algorithm across the datasets simulated from the settings described in Table 4.71. Red lines correspond to true values. See also Table 4.75.

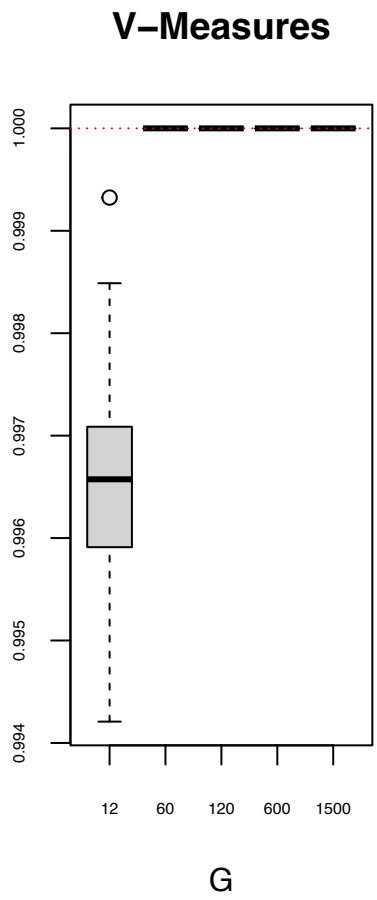

Figure 4.44: Scenario 2: Boxplots for the V-measures of the clustering obtained by the EM algorithm across the datasets simulated from the settings described in Table 4.71.

4.6 Simulation scenarios for the ZINB mixture model with a size factor

In this section, we simulate only one scenario for the case of ZINB mixture model with a size factor when the number of cells varies as $N = 60, 120, 300, 600$, and 1200. We choose $G = 120$ as the number of genes and $K = 2$ as the number of clusters. We simulate $S = 100$ datasets, and the true values of the parameters and hyperparameters are set as follows:

- $\pi = (0.5, 0.5)$;
- $\phi = (0.1, 0.2)$;
- T_n's are generated from a normal distribution with $\mu = 10$ and $\sigma = 0.5$;
- $\beta_{0g} = 0.85$ for all g;
- $\rho_{1g} = (2, \ldots, 2, -2, \ldots, -2)$, and
- $\rho_{2g} = (-2, \ldots, -2, 2, \ldots, 2)$.

It should be mentioned that, for the cluster effect parameters (ρ_{gk}'s), over each cluster, the first half of the genes, are assigned one value (either 2 or −2) and the remaining half are assigned another value (either 2 or −2) in such a way that their sums are equal to zero. The setting for this simulation scenario is shown in Table 4.76

Table 4.76: ZINB mixture model with a size factor. Scenario 1: Values chosen for the number of observations N in each case along with the fixed parameters used to simulate the datasets.

Case	N	G	K	ϕ_k	π_k
1	60				
2	120				
3	300	120	2	$(\phi_1 = 0.1, \phi_2 = 0.2)$	$1/K$
4	600				
5	1200				

For this scenario, we can see from Table 4.77 and Figure 4.45 that the estimated values of the probabilities of cluster assignment ($\hat{\pi}_k$) are approximately close to their true values and their standard deviations decrease as N increases. Furthermore, from Table 4.78 and Figure 4.46, we can observe similar behaviour for the estimated values of the probability of always zero ($\hat{\phi}_k$); that is, as N increases, their standard deviations decrease, and their estimated values are

close to their true values. Tables 4.79 and 4.80 demonstrate the MSEs of the estimated values of $\hat{\rho}_{gk}$ and $\hat{\beta}_{0g}$. As can be seen from these tables, the MSEs reduce as N becomes larger for both parameters. The behaviour of the estimated values of the size parameters (ν_1 and ν_2) are shown in Table 4.81 and Figure 4.47. As expected, the estimated values of ν_1 and ν_2 get closer to their true values, and their standard deviations decrease as N increases. For all cases, V-measures are equal to one, which shows the perfect performance of the inferred cluster assignments. Finally, Table B.29 and Figure B.29 show that the computing time increases when N increases.

Table 4.77: Scenario 1: Mean and standard deviation (SD) for the estimates of π_k for each k and each N, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.76.

k	N	Mean	SD
1	60	0.499000000	0.059403129
	120	0.499416667	0.044274533
	300	0.499700000	0.024989762
	600	0.496466667	0.018173129
	1200	0.497150000	0.012803652
2	60	0.501000000	0.059403129
	120	0.500583333	0.044274533
	300	0.500300000	0.024989762
	600	0.503533333	0.018173129
	1200	0.502850000	0.012803652

Table 4.78: Scenario 1: Mean and standard deviation (SD) for the estimates of ϕ_k for each k and each N, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.76.

k	N	Mean	SD
1	60	0.098709502	0.005962128
	120	0.099481329	0.003898979
	300	0.099814282	0.002657849
	600	0.099822926	0.001770076
	1200	0.100074985	0.001325751
2	60	0.198481193	0.006203301
	120	0.199096122	0.005127590
	300	0.200084764	0.003133217
	600	0.200102372	0.002030886
	1200	0.199840261	0.001712608

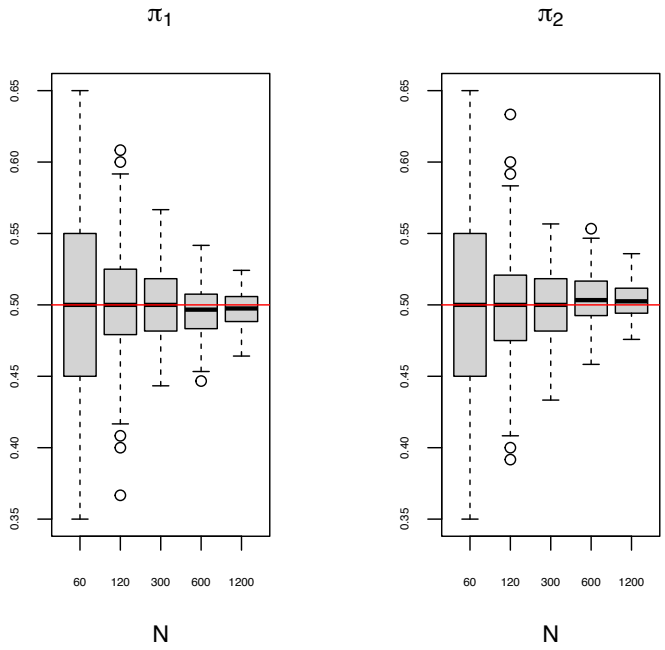

Figure 4.45: Scenario 1: Boxplots for the estimates of π_k using the EM algorithm across the datasets simulated from the settings described in Table 4.76. Red lines correspond to true values. See also Table 4.77.

Table 4.79: Scenario 1: Mean Squared error (MSE) for the estimates of ρ_{gk} for each k and each N, using the EM algorithm across the datasets simulated from the settings described in Table 4.76.

	N				
k	60	120	300	600	1200
1	0.08710	0.06189	0.04038	0.02848	0.01996
2	0.04873	0.03532	0.02200	0.01585	0.01127

Table 4.80: Scenario 1: Mean squared error (MSE) for the estimates of β_{0g} for each N, using the EM algorithm across the datasets simulated from the settings described in Table 4.76.

	N				
	60	120	300	600	1200
β_{0g}	0.04207	0.02902	0.01791	0.01278	0.00935

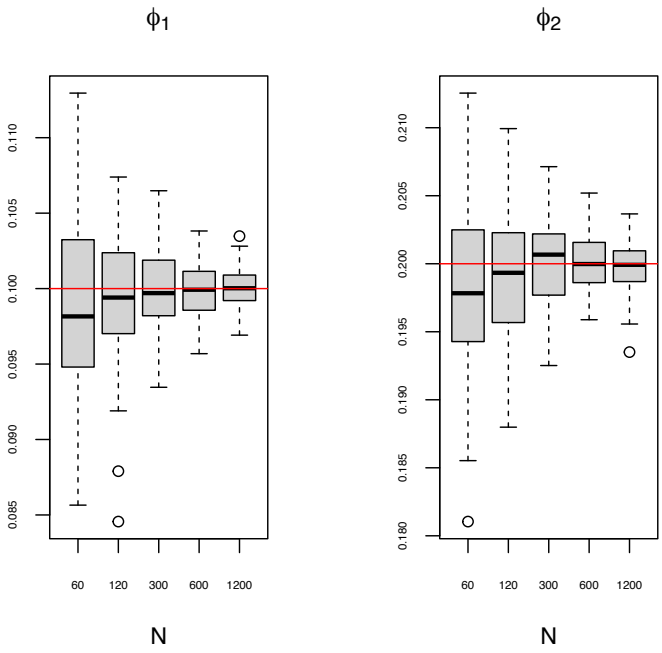

Figure 4.46: Scenario 1: Boxplots for the estimates of ϕ_k using the EM algorithm across the datasets simulated from the settings described in Table 4.76. Red lines correspond to true values. See also Table 4.78.

Table 4.81: Scenario 1: Mean and standard deviation (SD) for the estimates of ν_k for each k and each N, obtained using the EM algorithm across the datasets simulated from the settings described in Table 4.76.

k	N	Mean	SD
1	60	5.2327	0.1597
	120	5.1426	0.1225
	300	5.0433	0.0672
	600	5.0199	0.0461
	1200	5.0104	0.0368
2	60	21.1015	0.8673
	120	20.5514	0.6456
	300	20.2056	0.3604
	600	20.0906	0.2556
	1200	20.0455	0.1791

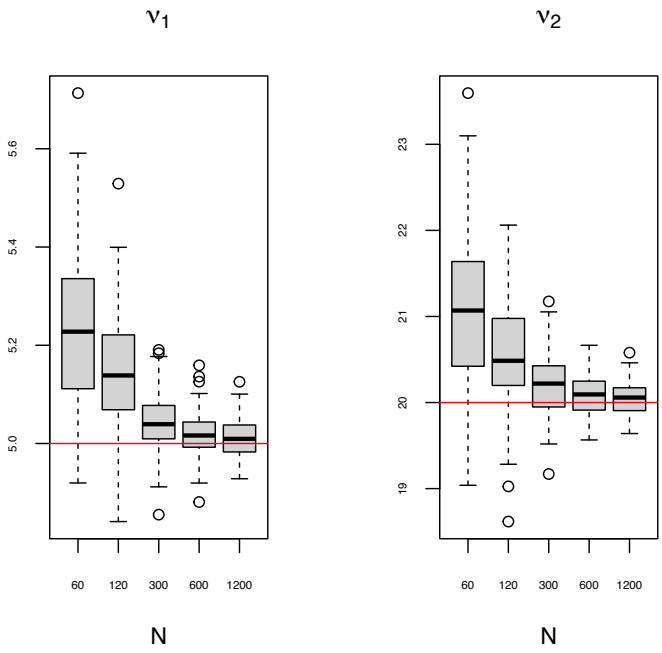

Figure 4.47: Scenario 1: Boxplots for the estimates of ν_k using the EM algorithm across the datasets simulated from the settings described in Table 4.76. Red lines correspond to true values. See also Table 4.81.

Chapter 5

Data Analysis

In this chapter, we apply some of the proposed ZIP and ZINB mixture models introduced in Chapter 3 to two publicly available datasets. In Section 5.1, we consider scRNA-seq data from mouse embryonic stem cells collected by Klein et al. (2015) and fit the ZIP mixture model without covariates as well as the ZIP mixture model with a size factor. In Section 5.2, we analyze scRNA-seq data from mouse liver tissue cells profiled by Han et al. (2018) using the ZIP mixture model without covariates and the ZINB mixture without covariates.

5.1 Mouse Embryonic Stem Cell (MESC) data

Klein et al. (2015) developed a laboratory platform (called inDrop from indexing Droplets) for indexing thousands of individual cells for RNA sequencing. Klein et al. then used inDrop to obtain single-cell RNA sequencing data from mouse embryonic stem cells before (day 0) and after leukemia inhibitory factor (LIF) withdrawal (days 2, 4, and 7). Read counts across all cells and genes for the different experiment days in Klein et al. (2015) are publicly available through the Gene Expression Omnibus online repository under the accession code GSE65525.

For our analysis, we consider the pooled data for day 0 (933 cells) and day 4 (683 cells) for a total of N = 1,616 cells and G = 24,175 genes. Then, we perform a filtering (selection) step by filtering out genes with very little variation across all cells. This is a common step in the analysis of scRNA-seq data (Klein et al., 2015, Zeisel et al., 2015). In this case, we filter out genes with a read count interquartile range across cells smaller than one (IQR = $Q_3 - Q_1 \leq 1$), resulting in 4,514 genes initially selected. From these 4,514 genes, we select 100 of them with the highest read count standard deviations across cells. Therefore, we continue the data analysis in this section using the read count data for N = 1,616 cells and the selected G = 100 most variable genes. Note that our choice to select 100 genes was based on reducing the computation complexity of running different models for different number of clusters with

different initializations. However, one can consider a larger number of genes and compare the results.

We fit the mouse embryonic stem cell data (MESC) data considering the ZIP mixture model without covariates (Section 3.5.1) and with a size factor (as the total number of read counts for each cell before performing any gene filtering) as shown in Equation (3.30) of Section 3.5.2. For each model, we apply the proposed EM algorithm considering different choices of K (total number of clusters) and two clustering initialization methods: K-means and random clustering. After obtaining initial cluster assignments for the cells by K-means or random clustering, we can find the initial parameter values required to start the EM algorithm as follows. For the cluster probabilities, the π_k's, we set their initial values to the proportion of cells assigned to each initial cluster. For the probabilities of always zero, the ϕ_k's, we set each $\phi_k^{(0)}$ to the proportion of zero entries in each cluster. For the case of the ZIP mixture model without covariates (simple ZIP model), for each initial cluster, we take the mean read count for each gene as the initial values of the λ_{gk}'s. For the ZIP mixture model with a size factor, we initialize the β_{0g}'s at zero, and the cluster effects, the ρ_{gk}'s, as the mean read count for each gene for each initial cluster.

For each choice of K and each initialization method (random or K-means), we run the EM algorithm 32 times corresponding to 32 different initialization runs from different seeds. Next, for each initialization method, we choose the run with the smallest Aikaike Information Criterion (AIC) for each possible total number of clusters K. For each initialization method, after choosing the best run over each K, we use the elbow method to select the optimum number of clusters. The elbow takes the point with the highest AIC (on the y-axis) and the point with the highest K (in the x-axis) and defines a line, usually going from the top-left to the bottom-right on the plot. The optimum point is then determined to be the one that is the farthest away below this line.

In what follows, we present the results of fitting the ZIP mixture model without covariates (simple ZIP) and with a size factor to the MESC data.

5.1.1 Results of fitting the ZIP mixture model without covariates to the MESC data

This section presents the results of fitting the ZIP mixture model without covariates (simple ZIP) to the MESC dataset. As mentioned above, for our analysis, we use the pooled data of day 0 and day 4 over the 100 most variable selected genes and 1,616 cells.

Figures 5.1 and 5.3 show the boxplots of AIC values over the 32 EM runs for different values of K for the random and K-means cluster initialization approaches, respectively. Figures

5.2 and 5.4 show the corresponding smallest AIC for each K for the random and K-means initialization methods, respectively. Based on the elbow method, both initialization methods lead to $K = 4$ as the best number of clusters for the simple ZIP model (see the red point in Figures 5.2 and 5.4). Next, as a final choice between these two results with $K = 4$, one using random clustering initialization and the other K-means, we select the one with the lowest AIC. In this case, the AIC from the K-means initialization approach reaches a lower value than that of the random initialization. Therefore, we select the EM run with the best AIC for $K = 4$ from the K-means initialization approach and present its results in what follows.

Figure 5.5 and Table 5.1 present the co-clustering plot and confusion matrix, respectively, between cell experiment days (day 0 and day 4) and the inferred cell clusters (1, 2, 3, and 4) for the best EM algorithm run with $K = 4$. The co-clustering plot allows us to observe the percentage of cells from each experiment day present in each inferred cluster. From both Figure 5.5 and Table 5.1, we can see that most cells (approximately 99%) of day 4 are present in the inferred cluster 2 and cells from day 0 fall mainly into the inferred clusters 1, 3, and 4. Interestingly, Klein et al. (2015) found in their analyses that cells from day 0 belong to three main subpopulations plus two other rare subpopulations when clustering only day 0 cells via hierarchical clustering.

Figure 5.6 shows the heatmap of the data (read counts across all 1,616 cells and all 100 selected genes) with cells (rows) ordered by their inferred cluster assignments. The heatmap also contains annotation for each cell's experiment day (0 or 4). Figure 5.7 presents a dimensionality reduction visualization of the data using t-SNE (Van der Maaten and Hinton, 2008). In the t-SNE plot, circle and triangle points correspond to cells from day 0 and day 4, respectively, and the colours to the inferred four clusters. Moreover, Figure 5.8 shows the cluster assignment expected values (or probabilities), i.e., the $\hat{Z}_{nk}$'s, which we used to determine the final inferred cluster assignment of each cell as shown in Equation (3.20). We can observe that overall the proposed EM algorithm assigned cells to their clusters with high (close to 1) probabilities.

Table 5.2 shows the estimated cluster proportion, $\hat{\pi}_k$, for each of the inferred clusters. Cluster 2 has the highest proportion of cell assignments ($\hat{\pi}_2 = 45.91\%$) compared to the other three clusters. The estimated probability of always zero for each cluster is shown in Table 5.3 in which cluster 2 shows the highest probability with $\hat{\phi}_2 = 1.191\%$. Finally, Figure 5.9 shows the heatmap of the estimates of the rate parameters ($\hat{\lambda}_{gk}$'s) for each cluster (rows) over the 100 selected genes (columns) when fitting the ZIP simple model to the MESC data.

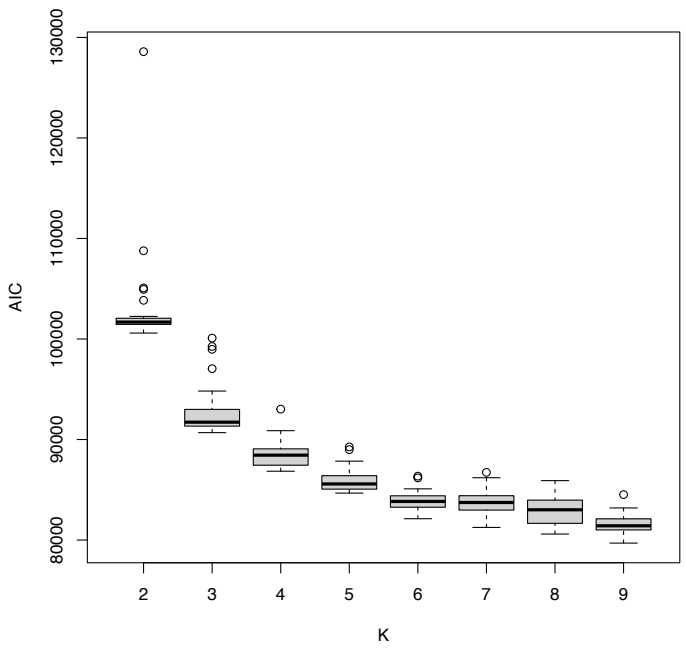

Figure 5.1: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the simple ZIP model to the MESC dataset with random clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

Table 5.1: Confusion matrix between the EM clustering result when fitting the simple ZIP model to the MESC data and the experiment day labels. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model (see Figure 5.4).

	Inferred cluster			
Day	1	2	3	4
0	218	64	396	255
4	0	676	0	7

Table 5.2: Estimates of π_k for the MESC dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model.

k	$\hat{\pi}_k$
1	0.13599
2	0.45910
3	0.24507
4	0.15984

Table 5.3: Estimates of ϕ_k for the MESC dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model.

k	$\hat{\phi}_k$
1	0.00077
2	0.01191
3	0.00077
4	0.00127

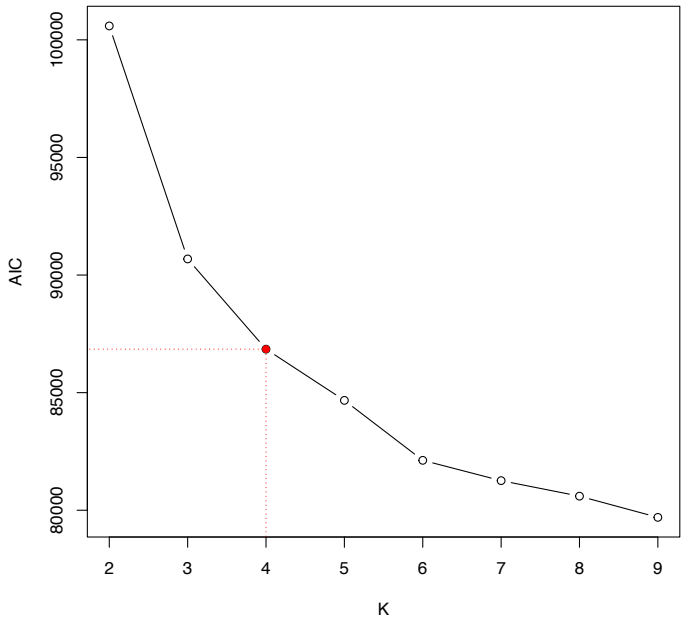

Figure 5.2: Plot of the best AIC for each K obtained from applying the EM algorithm under the simple ZIP model to the MESC dataset with random clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 4$.

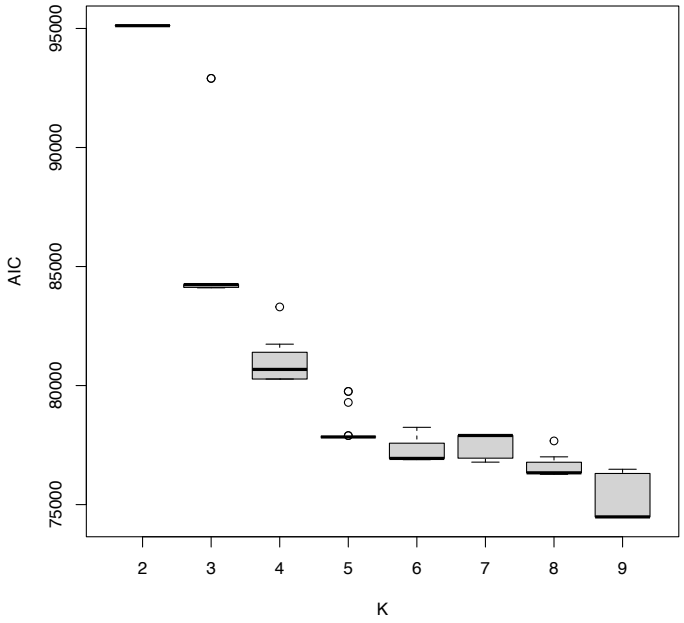

Figure 5.3: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the simple ZIP model to the MESC data set with K-means clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

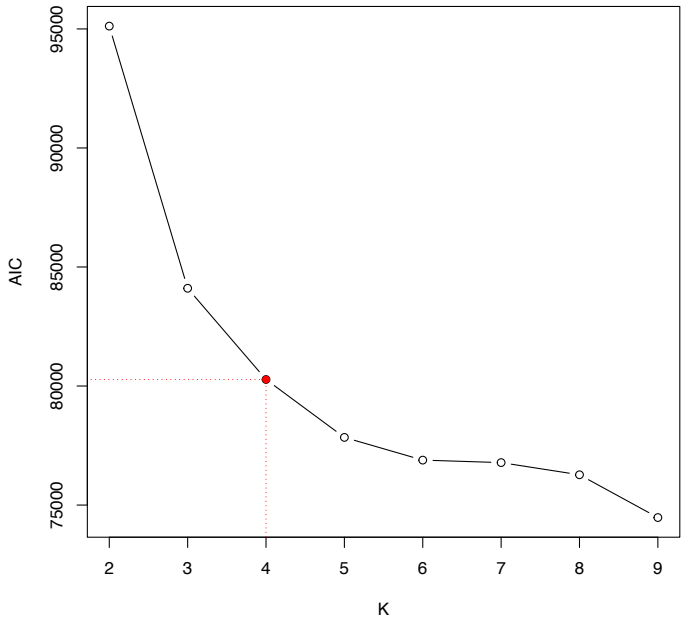

Figure 5.4: Plot of the best AIC for each K obtained from applying the EM algorithm under the simple ZIP model to the MESC dataset with K-means clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 4$.

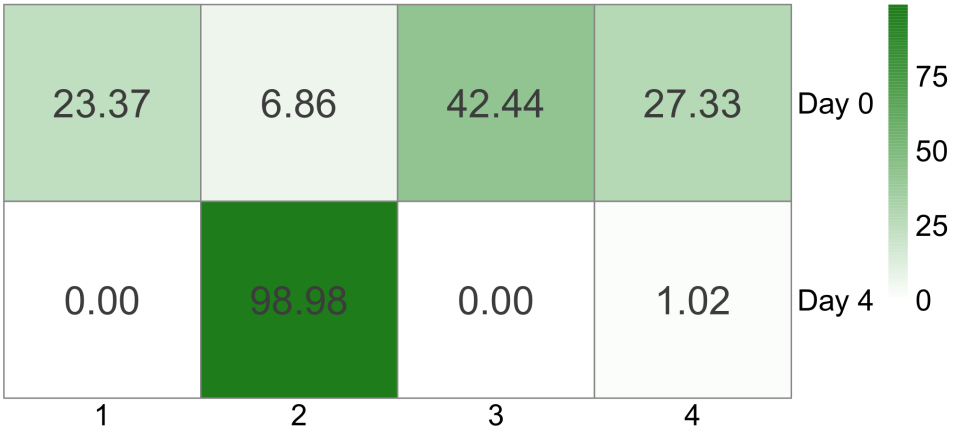

Figure 5.5: MESC dataset. Co-clustering between experiment days (0 and 4; rows) and inferred clusters by the proposed EM algorithm (1, 2, 3, and 4; columns). Each entry a_{ij} represents the % of cells from day i that are present in the inferred cluster j. Rows sum up to 100%. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model (see Figure 5.4).

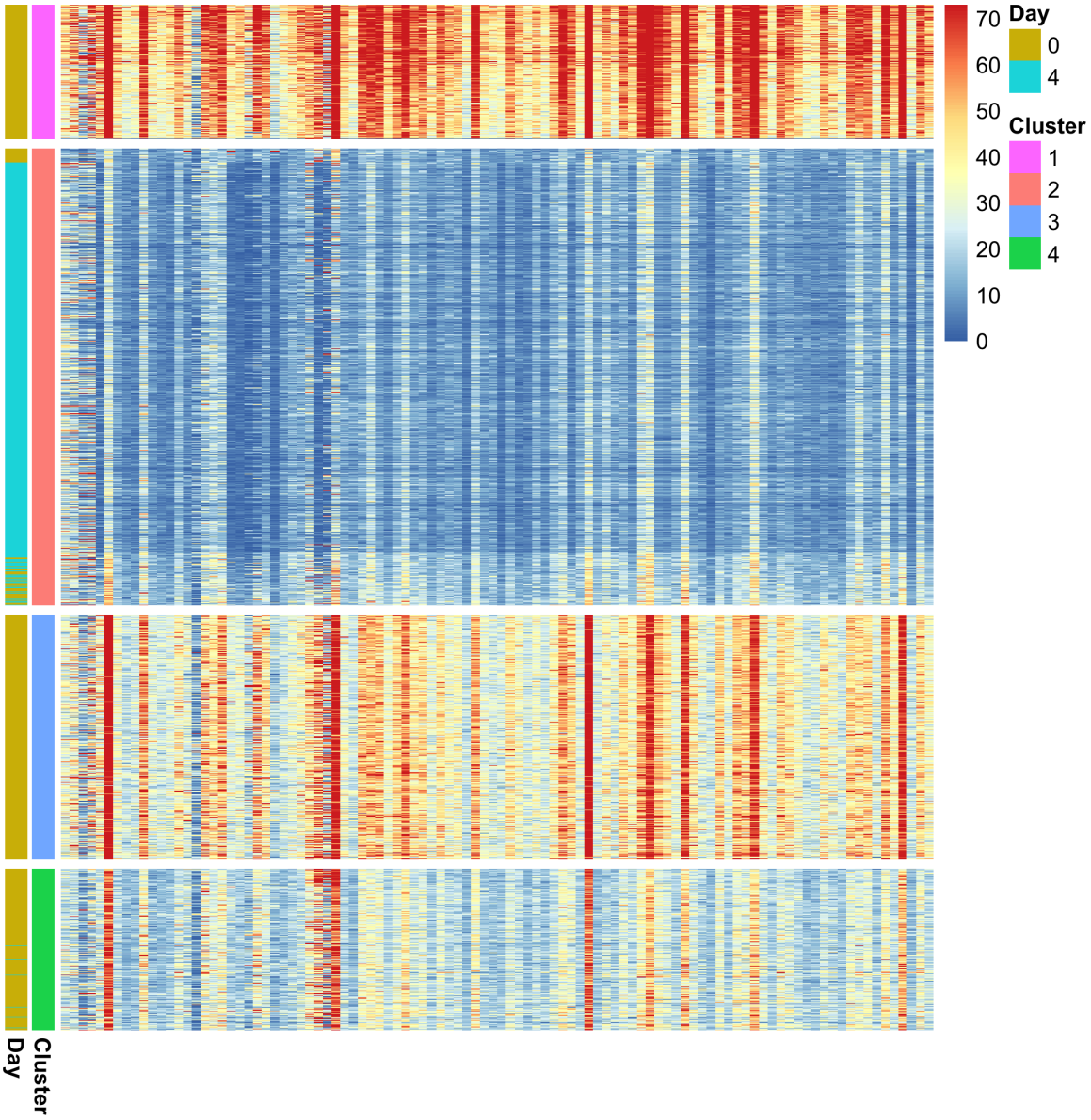

Figure 5.6: Heatmap of MESC data displaying read counts across all 1,616 cells (rows) and all 100 selected genes (columns). Cells (rows) are ordered by their inferred cluster assignments obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model. The first column on the left shows the annotation for each cell's experiment day (0 or 4). Dark blue colours represent low read count values, and dark red colours represent high read count values. Note that to facilitate visualization under this colour scheme, read counts with values higher than the 95th percentile were truncated at the value of the 95th percentile.

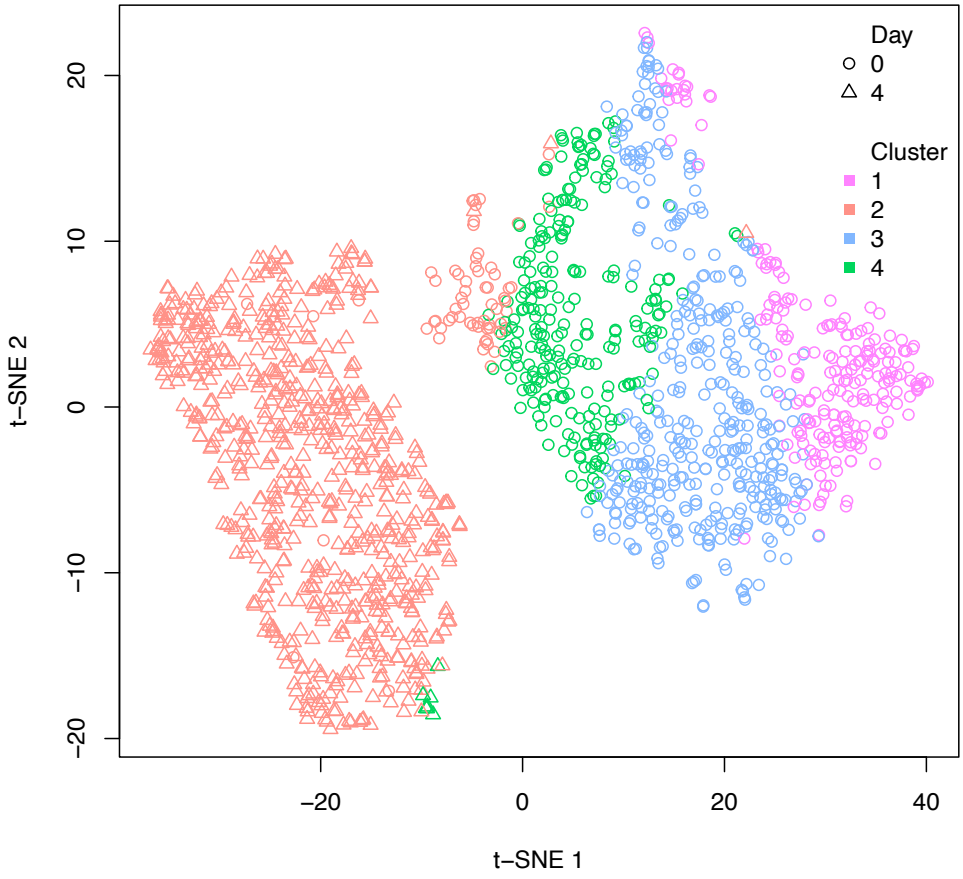

Figure 5.7: *t*-SNE plot for the MESC dataset and the clustering obtained from the best EM algorithm run (*K*-means initialization and $K = 4$) under the simple ZIP model. Each point represents a cell with the shape symbol indicating the experiment day label (day 0 or day 4), and the colour the corresponding inferred cluster (1, 2, 3, or 4).

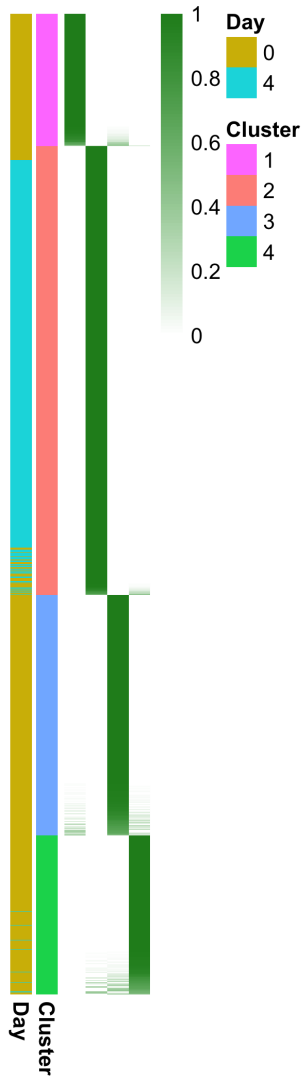

Figure 5.8: Heatmap of the $\hat{Z}_{nk}$'s for MESC dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model. Each row shows the estimated probability of a cell n belonging to each cluster k (columns). Rows are ordered by the final inferred cluster assignments determined by Eq. (3.20). The labels on the left show the assigned clusters and the day labels. Dark colours represent high probabilities.

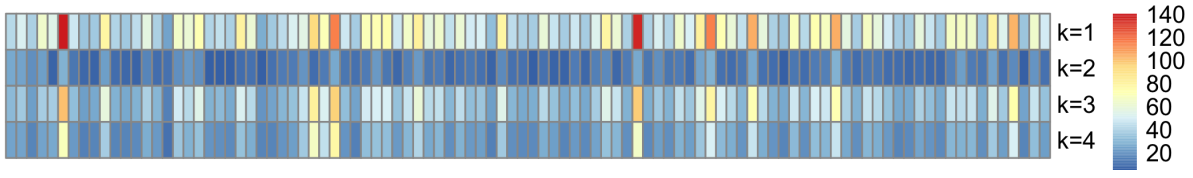

Figure 5.9: Heatmap of the $\hat{\lambda}_{gk}$'s for MESC data set obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model. Each row corresponds to a cluster, and each column to a gene. Dark blue colours represent low values, and dark red colours represent high values of $\hat{\lambda}_{gk}$.

5.1.2 Results of fitting the ZIP mixture models with size factor to the MESC data

In this section, we present the results of fitting the ZIP mixture model with size factor (as shown in Equation (3.30) of Section 3.5.2) to the same MESC dataset analyzed in Section 5.1.1; that is, the pooled data of day 0 and day 4 over the 100 most variable selected genes and 1,616 cells.

Figures 5.10 and 5.12 show the boxplots of AIC values over the 32 runs of the EM algorithm for each K using the random and K-means clustering initialization approaches, respectively. Figures 5.11 and 5.13 show the corresponding smallest AIC for each K for random and K-means initialization methods, respectively. Based on the elbow method, the random initialization approach leads to $K = 6$ clusters as the best number of clusters for the ZIP mixture model with size factor; however, the K-means initialization method leads to $K = 4$ as the optimum number for clusters (see the red point in Figures 5.11 and 5.13). As a final choice, we choose the best number of clusters between these two initialization methods based on the lowest AIC. Thus, as K-means initialization leads to the smallest AIC value, we choose the EM run with the best AIC from the K-means method when $K = 4$ and present its results in the following.

Figure 5.14 and Table 5.4 show the co-clustering plot and confusion matrix, respectively, between cell experiment days (0 and 4) and the inferred four clusters for the best EM algorithm run ($K = 4$ clusters and K-means clustering initialization method) for the ZIP mixture model with size factor. From both Figure 5.14 and Table 5.4, we can see that all cells (100%) of day 4 are present in the inferred cluster 1 and, interestingly, most cells (approximately 98.5%) of day 0 are in the inferred cluster 4 and only a few of cells from day 0 are in the other clusters; that is, 0.21% in cluster 1, 0.11% in cluster 2, and 1.18% in cluster 3. These clustering results are similar to the ones presented by Qi et al. (2020).

Similarly to the previous section, Figure 5.15 shows the heatmap of the read counts across all 1,616 cells and all 100 selected genes with cells (rows) ordered by their inferred cluster assignments from the ZIP mixture model with size factor. The heatmap also shows each cell's experiment day (0 or 4). Figure 5.16 shows the t-SNE representation of the data in two dimensions, where circle and triangle points correspond to cells from day 0 and day 4, respectively, and the colours to the inferred four clusters. As in Section 5.1.1, we can observe in Figure 5.17 that overall the proposed EM algorithm assigned cells to their clusters with high (close to 1) probabilities.

The estimated cluster proportions ($\hat{\pi}_k$'s) are presented in Table 5.5. We can see from the table that more than 50% of the cells belong to cluster 4 (56.83%), and 42.4% of the cells fall into cluster 1 and only a few of them are assigned to the other two clusters. Table 5.6 shows

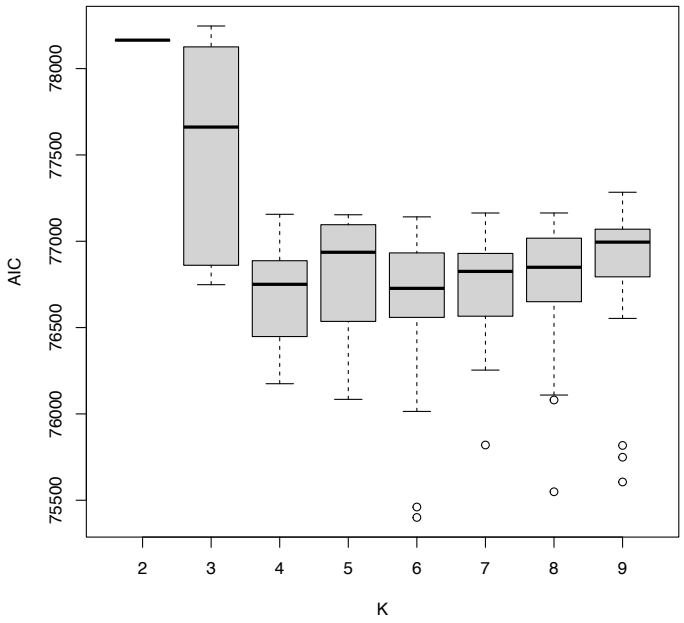

Figure 5.10: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the ZIP mixture model with size factor to the MESC dataset with random clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

the estimated probability of always zero for each cluster k, and we can see that $\hat{\phi}_1 = 0.01415$ has the highest probability of always zero compared with the other clusters. Figure 5.18 shows the heatmap of the estimates of β_{0g} (baseline expression) and ρ_{gk} (cluster effect) over the 100 selected genes (columns) when fitting the ZIP mixture model with size factor. The $\hat{\beta}_{0g}$'s are shown in the first row, and the $\hat{\rho}_{gk}$'s for each cluster k are presented in rows 2 to 5.

Table 5.4: Confusion matrix between the EM clustering result when fitting the ZIP mixture model with size factor to the MESC data and the experiment day labels. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor (see Figure 5.13).

	Inferred cluster			
Day	1	2	3	4
0	2	1	11	919
4	683	0	0	0

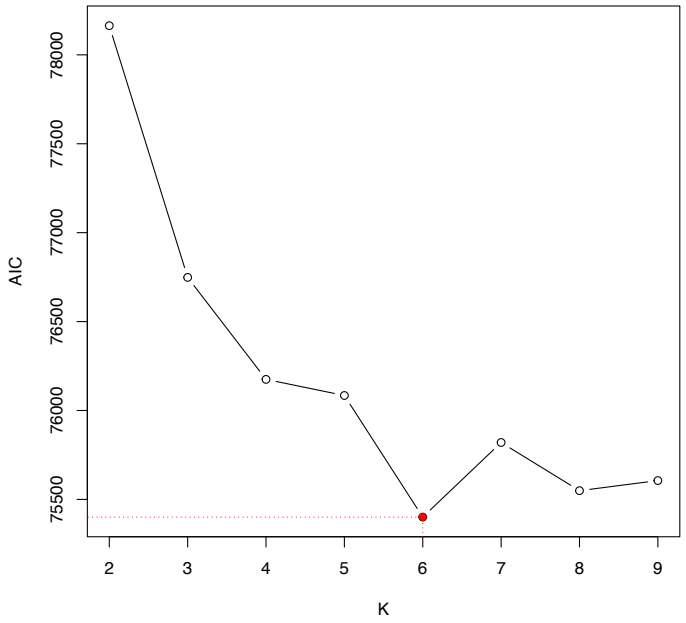

Figure 5.11: Plot of the best AIC for each K obtained from applying the EM algorithm under the ZIP mixture model with size factor to the MESC dataset with random clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 6$.

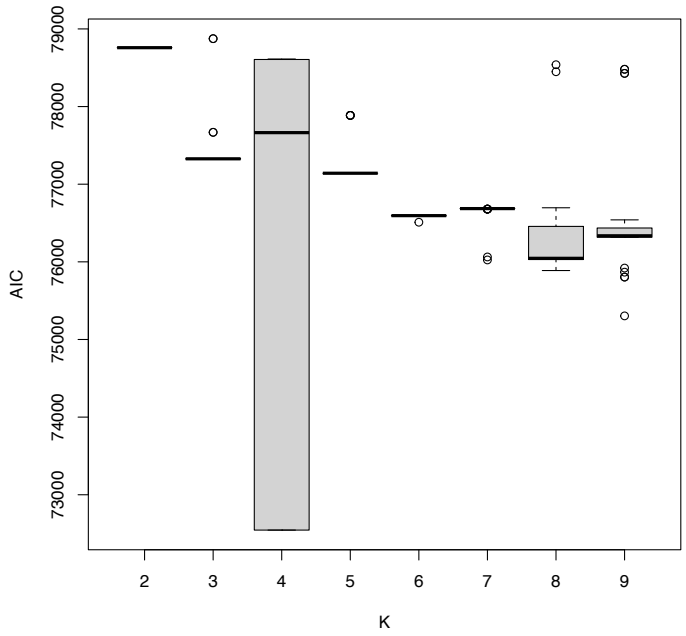

Figure 5.12: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the ZIP mixture model with size factor to the MESC data set with K-means clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

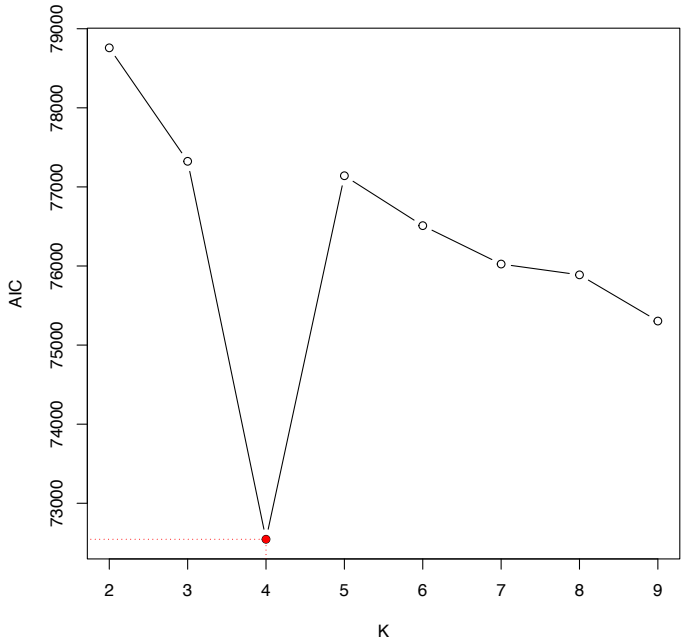

Figure 5.13: Plot of the best AIC for each K obtained from applying the EM algorithm under the ZIP mixture model with size factor to the MESC dataset with K-means clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 4$.

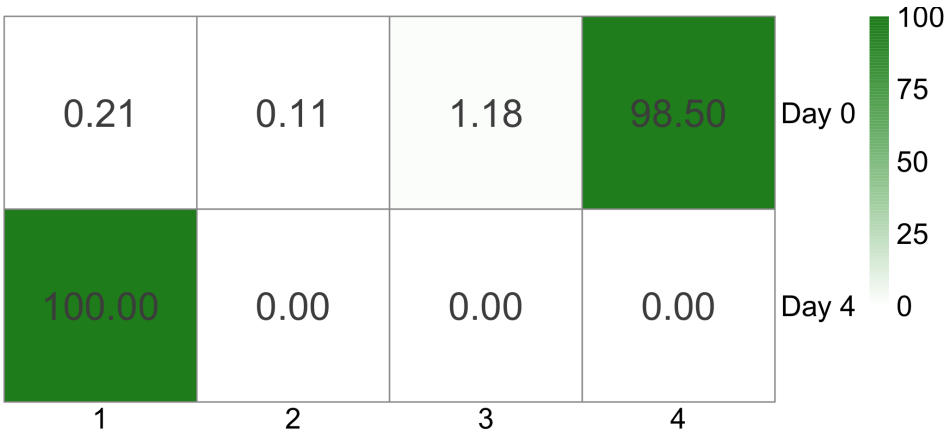

Figure 5.14: MESC dataset. Co-clustering between experiment days (0 and 4; rows) and inferred clusters by the proposed EM algorithm (1, 2, 3, and 4; columns). Each entry a_{ij} represents the % of cells from day i that are present in the inferred cluster j. Rows sum up to 100%. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor (see Figure 5.13).

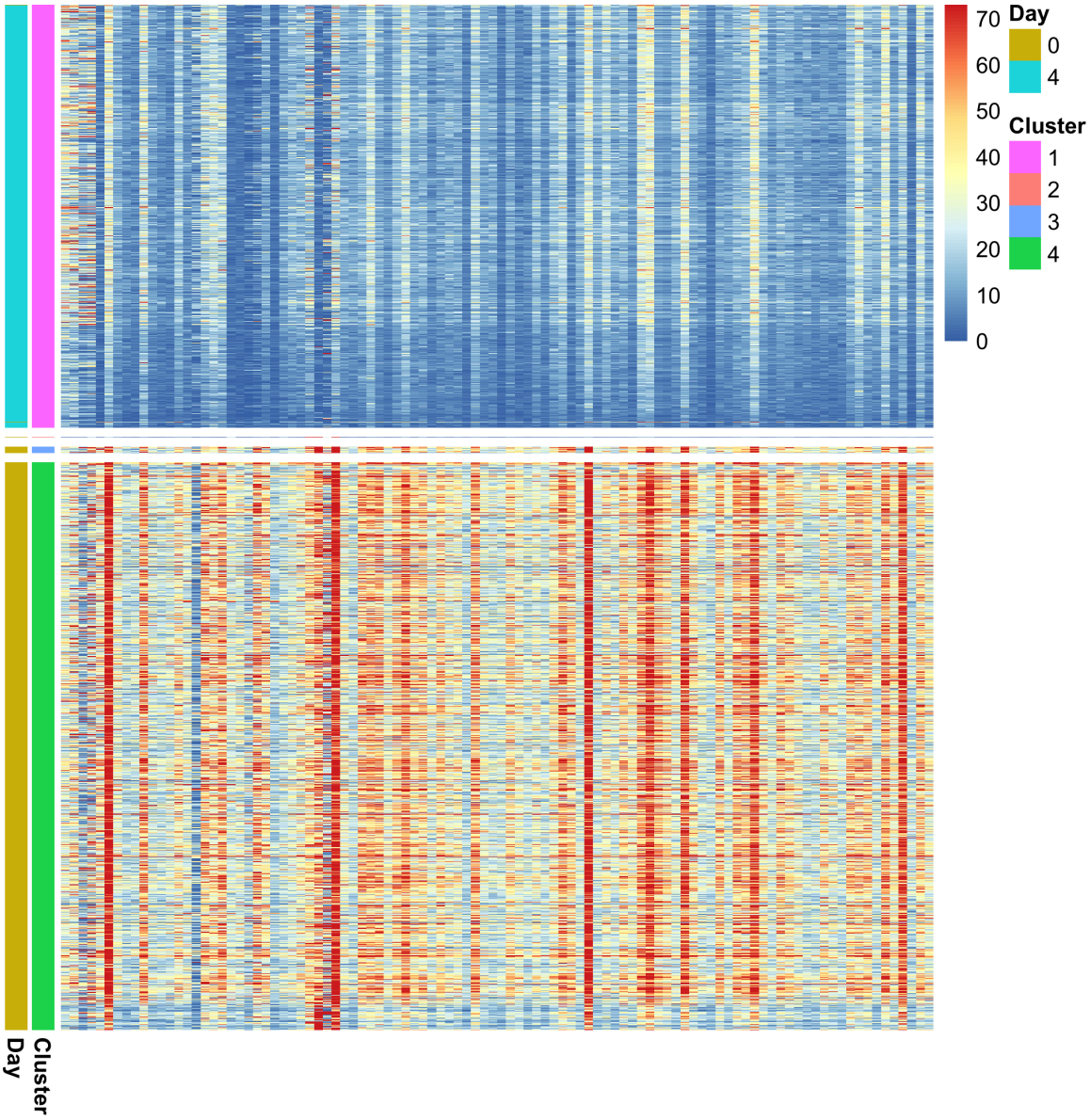

Figure 5.15: Heatmap of MESC data displaying read counts across all 1,616 cells (rows) and all 100 selected genes (columns). Cells (rows) are ordered by their inferred cluster assignments obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor. The first column on the left shows the annotation for each cell's experiment day (0 or 4). Dark blue colours represent low read count values, and dark red colours represent high read count values. Note that to facilitate visualization under this colour scheme, read counts with values higher than the 95th percentile were truncated at the value of the 95th percentile.

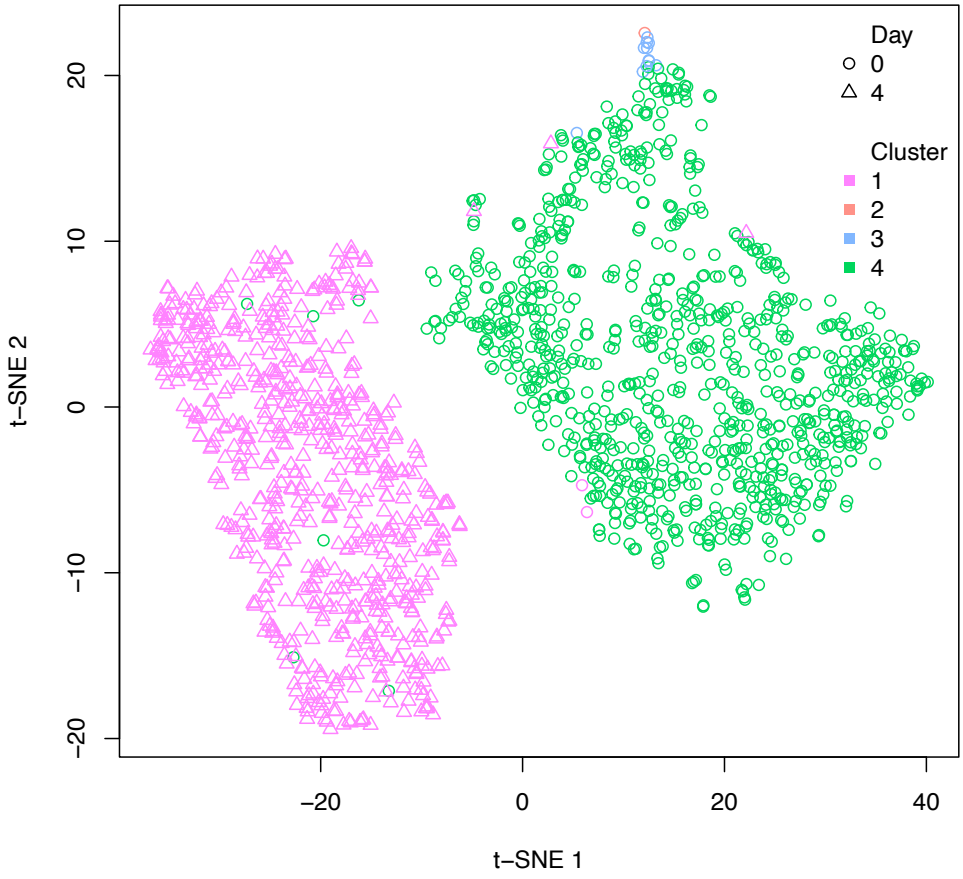

Figure 5.16: *t*-SNE plot for the MESC dataset and the clustering obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor. Each point represents a cell with the shape symbol indicating the experiment day label (day 0 or day 4), and the colour the corresponding inferred cluster (1, 2, 3, or 4).

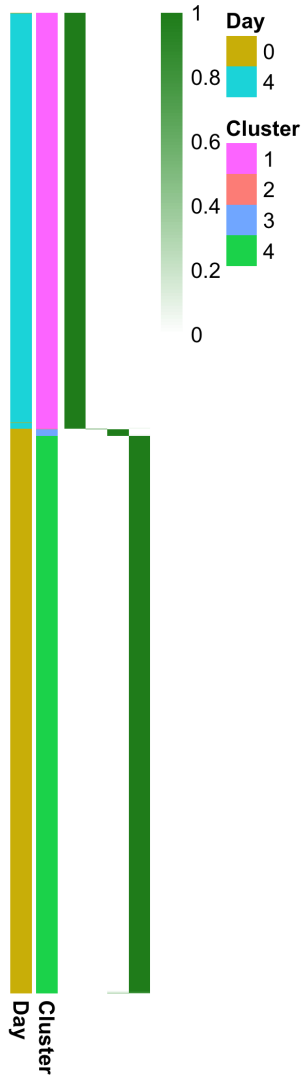

Figure 5.17: Heatmap of the $\hat{Z}_{nk}$'s for MESC dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor. Each row shows the estimated probability of a cell n belonging to each cluster k (columns). Rows are ordered by the final inferred cluster assignments determined by Eq. (3.20). The labels on the left show the assigned clusters and the day labels. Dark colours represent high probabilities.

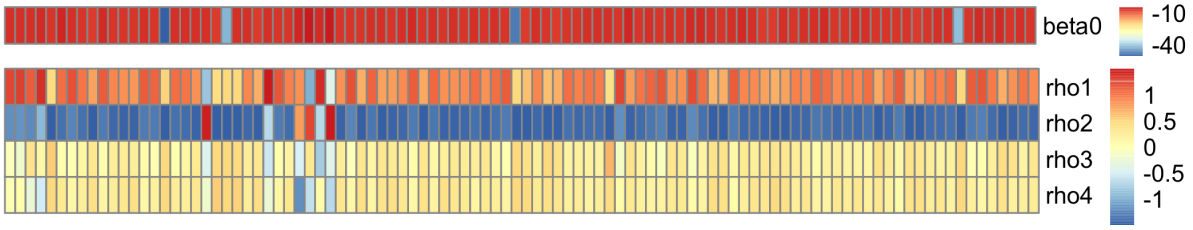

Figure 5.18: Heatmap of the $\hat{\beta}_{0g}$'s and $\hat{\lambda}_{gk}$'s for the MESC dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor. $\hat{\beta}_{0g}$'s are shown in the first row, and the $\hat{\rho}_{gk}$'s for each cluster k are presented in rows 2 to 5. The columns correspond to the 100 selected genes. Dark blue colours represent low values, and dark red colours represent high values of $\hat{\beta}_{0g}$ and $\hat{\rho}_{gk}$.

Table 5.5: Estimates of π_k for the MESC dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor.

k	$\hat{\pi}_k$
1	0.42384
2	0.00062
3	0.00727
4	0.56828

Table 5.6: Estimates of ϕ_k for the MESC dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the ZIP mixture model with size factor.

k	$\hat{\phi}_k$
1	0.01415
2	0.00000
3	0.00086
4	0.00072

5.1.3 Model selection for the MESC dataset

Table 5.7 shows the AIC values obtained from the best EM algorithm runs when fitting the simple ZIP model (ZIP mixture model without covariates) and the ZIP mixture model with size factor. These values correspond to the red points in Figures 5.4 and 5.13. Table 5.7 shows that the ZIP mixture model with a size factor leads to a smaller AIC value than the simple ZIP model. Therefore, the ZIP mixture model with a size factor fits the MESC data better than the ZIP simple model. As shown in Section 5.1.2, the selected ZIP mixture model with size factor resulted in two main clusters, one with cells from day 0 and the other with cells from day 4.

Table 5.7: AIC values corresponding to the best EM runs when fitting the simple ZIP model and the ZIP mixture model with size factor to the MESC dataset (see Figures 5.4 and 5.13).

Model	AIC
Simple ZIP	80274.22
ZIP mixture model with size factor	72544.74

5.2 Liver Data

Han et al. (2018) collected thousands of single-cell transcriptome profiles from several mouse tissues, organs, and cell cultures using Microwell-seq as a high-throughput and low-cost scRNA-seq platform. These data are publicly accessible through the Gene Expression Omnibus online repository under the accession code GSE108097. Han et al. also made their data available at `https://figshare.com/s/865e694ad06d5857db4b`. For the data analysis in this thesis, we consider a subset of the scRNA-seq data from mouse liver tissue provided by Han et al. (2018). The liver data's total number of cells and genes are N = 4,685 and G = 15,491, respectively. We first select N = 1,000 cells using random sampling, and then for these randomly selected cells, we filter out genes with the highest variation as the most remarkable genes. As mentioned earlier, the process of gene filtering and selecting the most remarkable (highly variable) genes is a common step in analyzing scRNA-seq data Klein et al. (2015), Zeisel et al. (2015). For this data, we choose 100 of the genes with the highest standard deviations of read count across cells. Therefore, the data analysis results presented in this section are based on the randomly selected 1,000 cells and 100 selected highly variable genes of the liver tissue data in Han et al. (2018), which we refer to simply as liver data from now on.

We fit the liver data considering the ZIP mixture model without covariates (simple ZIP, Section 3.5.1) and the ZINB mixture model without covariates (simple ZINB, Section 3.6.1) via the EM algorithm. Similarly to Section 5.1, we apply the proposed EM algorithm to the

liver data considering different choices of K (total number of clusters) and two clustering initialization approaches: K-means and random clustering. After obtaining the initial cluster assignments for the cells using the two initialization methods, we can find the initial starting points of the EM algorithm for each model fitting. We can find the starting points for the cluster probabilities (π_k's), proportions of always zero in each cluster (ϕ_k's) for both the simple ZIP and ZINB mixture models and the rate parameters of simple ZIP model as described earlier in Section 5.1. The initial rate parameters for the simple ZINB mixture model for each cluster are the mean read counts for each gene (similar to the initial rates for the simple ZIP mixture model). Finally, we calculate the initial values for the size parameters (ν_k's) for the simple ZINB mixture model as follows. For each k, first, we calculate μ and σ be the mean and standard deviation of read counts over all genes and cells in the initial cluster k. Then, we calculate the initial value for ν_k as:

$$\nu_k^{(0)} = \left[\left(\frac{\sigma}{\mu}\right)^2 - \frac{1}{\mu}\right]^{-1}.$$

Again, similar to section 5.1, for each choice of K and the initialization method, we run the proposed EM algorithm 32 times as the 32 different initialization run from different seeds, and choose the run with smallest AIC for each possible K for each initialization method. Then, after selecting the best run for each K, we use the elbow method to find the optimum number of clusters K.

We present the results of fitting the ZIP and ZINB mixture models without covariates to the liver data in the following sections.

5.2.1 Result of fitting the ZIP mixture model without covariates to the liver data

This section presents the results of fitting the ZIP mixture model (simple ZIP) to the liver data set. The data correspond to the 1,000 randomly selected cells and the 100 selected highly variable genes for our analysis.

Figures 5.19 and 5.21 show the boxplots of the AIC values over the 32 runs of the EM algorithm for each possible K for the random and K-means initialization methods, respectively. Figures 5.20 and 5.22 show the smallest AIC for each K for random and K-means initialization methods, respectively. Based on the elbow method, the random initialization approach leads to $K = 6$ clusters as the best number of clusters. The K-means approach yields $K = 4$ as the optimum number of clusters. As the best EM run from K-means initialization has the lowest value of AIC, in what follows, we present the results of the EM run with $K = 4$ from the K-means initialization method.

Figure 5.23 and Table 5.8 show the co-clustering and confusion matrix, respectively, be-

tween the cell type labels provided in the dataset by Han et al. (2018) and the inferred four clusters for the best EM run ($K = 4$ clusters and K-means initialization approach) for the ZIP mixture model without covariates. From both Figure 5.23 and Table 5.8, we can see that B cells (62.16%) are mostly assigned to the inferred cluster 3, dendritic cells are present equally in clusters 1 (50%) and 3 (50%), endothelial cells (91.76%) are mainly assigned to cluster 3, all epithelial cells are part of the inferred cluster 2, 77.59% of erythroblast cells and 84.62% of granulocyte cells are assigned to cluster 3, 90.14% of hepatocyte cells are in the inferred cluster 2, 67.14% and 32.86% of Kuppfer cells are assigned to clusters 1 and 3, respectively. 58% of the macrophage cells are present in cluster 3, and 42% of them are in cluster 1. All (100%) of the neutrophil cells and 92.65% of the T cells are assigned to the inferred cluster 3.

Similarly to previous sections, Figure 5.24 shows the heatmap of the read counts across the randomly selected 1000 cells and all 100 selected genes with cells (rows) ordered by their inferred cluster assignment from the simple ZIP mixture model. The heatmap also shows each cell's type. Figure 5.25 shows the t-SNE representation of the data in two dimensions, where different point shapes correspond to the cell type and point colours to the inferred four clusters. As in Sections 5.1.1 and 5.1.2, we can observe in Figure 5.26 that overall the proposed EM algorithm assigned cells to their clusters with high (close to 1) probabilities.

The estimated cluster proportions ($\hat{\pi}_k$'s) are presented in Table 5.9. We can see from the table that (60.929%) of the cells are assigned in the inferred cluster 3, 26.114% are assigned to cluster 1, 11.957% fall into cluster 2, and only 1% of them fall into cluster 4. Table 5.10 shows the estimated probability of always zero for each cluster k. We can see that the higher probability estimates are for the inferred clusters 2 and 4 ($\hat{\phi}_2 = 0.26895$ and $\hat{\phi}_4 = 0.13620$), respectively, followed by clusters 1 and 3 ($\hat{\phi}_1 = 0.09183$ and $\hat{\phi}_3 = 0.08780$). Finally, Figure 5.27 shows the heatmap of the estimates of the rate parameters ($\hat{\lambda}_{gk}$'s) for each cluster (rows) over the 100 selected genes (columns) when fitting the ZIP simple model to the liver data.

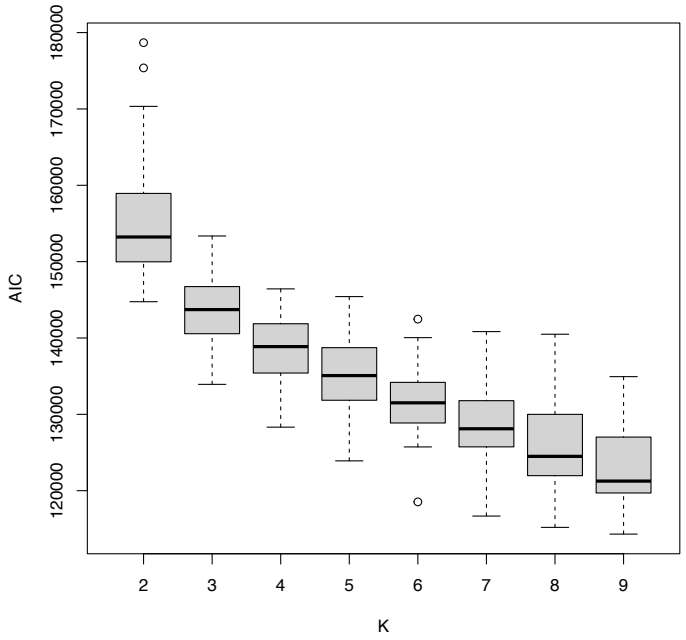

Figure 5.19: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the simple ZIP model to the liver dataset with random clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

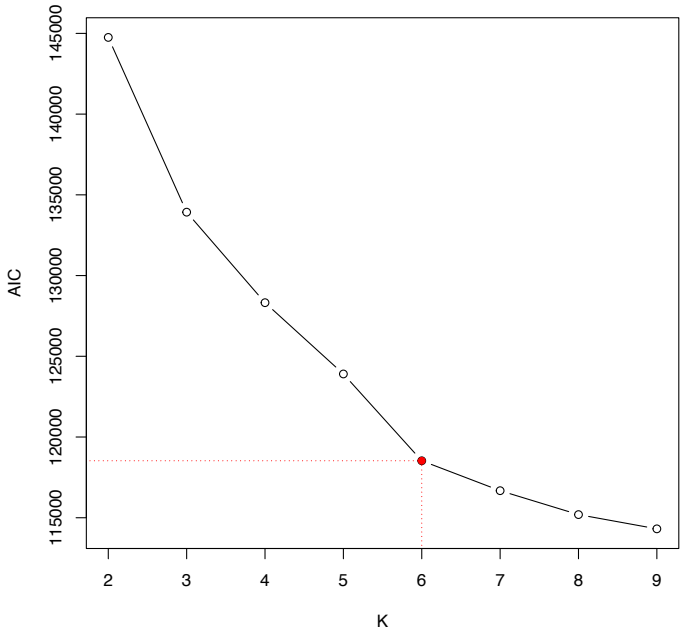

Figure 5.20: Plot of the best AIC for each K obtained from applying the EM algorithm under the simple ZIP model to the liver dataset with random clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 6$.

Table 5.8: Confusion matrix between the EM clustering result when fitting the simple ZIP model to the liver data and the cell types. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model (see Figure 5.23).

	Inferred cluster			
Cell type	1	2	3	4
B cell	3	1	23	10
Dendritic cell	54	0	54	0
Endothelial cell	17	5	245	0
Epithelial cell	0	32	0	0
Erythroblast	9	17	90	0
Granulocyte	5	1	33	0
Hepatocyte	2	64	5	0
Kuppfer cell	141	0	69	0
Macrophage	21	0	29	0
Neutrophil	0	0	2	0
T cell	5	0	63	0

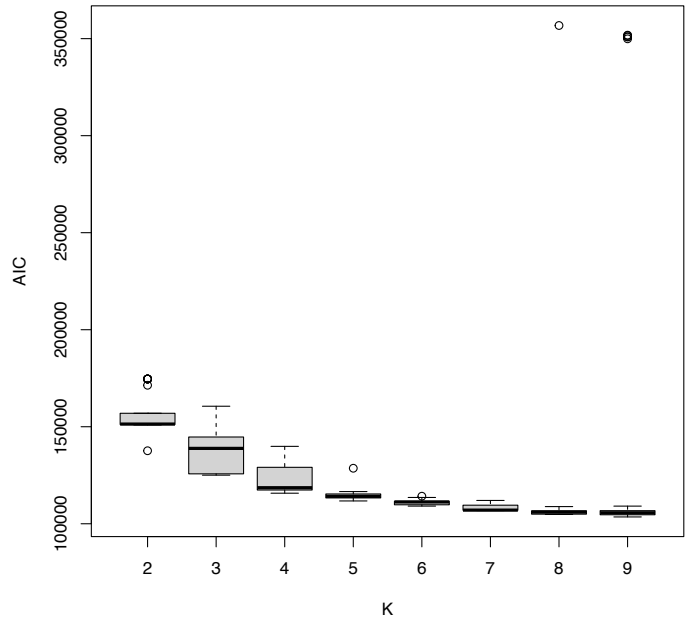

Figure 5.21: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the simple ZIP model to the liver data set with K-means clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

Table 5.9: Estimates of π_k for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model.

k	$\hat{\pi}_k$
1	0.26114
2	0.11957
3	0.60929
4	0.01000

Table 5.10: Estimates of ϕ_k for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model.

k	$\hat{\phi}_k$
1	0.09183
2	0.26895
3	0.08780
4	0.13620

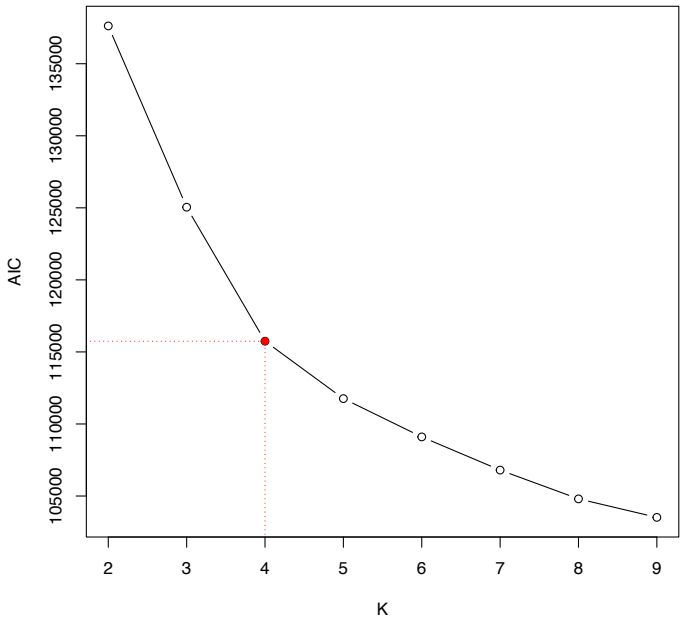

Figure 5.22: Plot of the best AIC for each K obtained from applying the EM algorithm under the simple ZIP model to the liver dataset with K-means clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 4$.

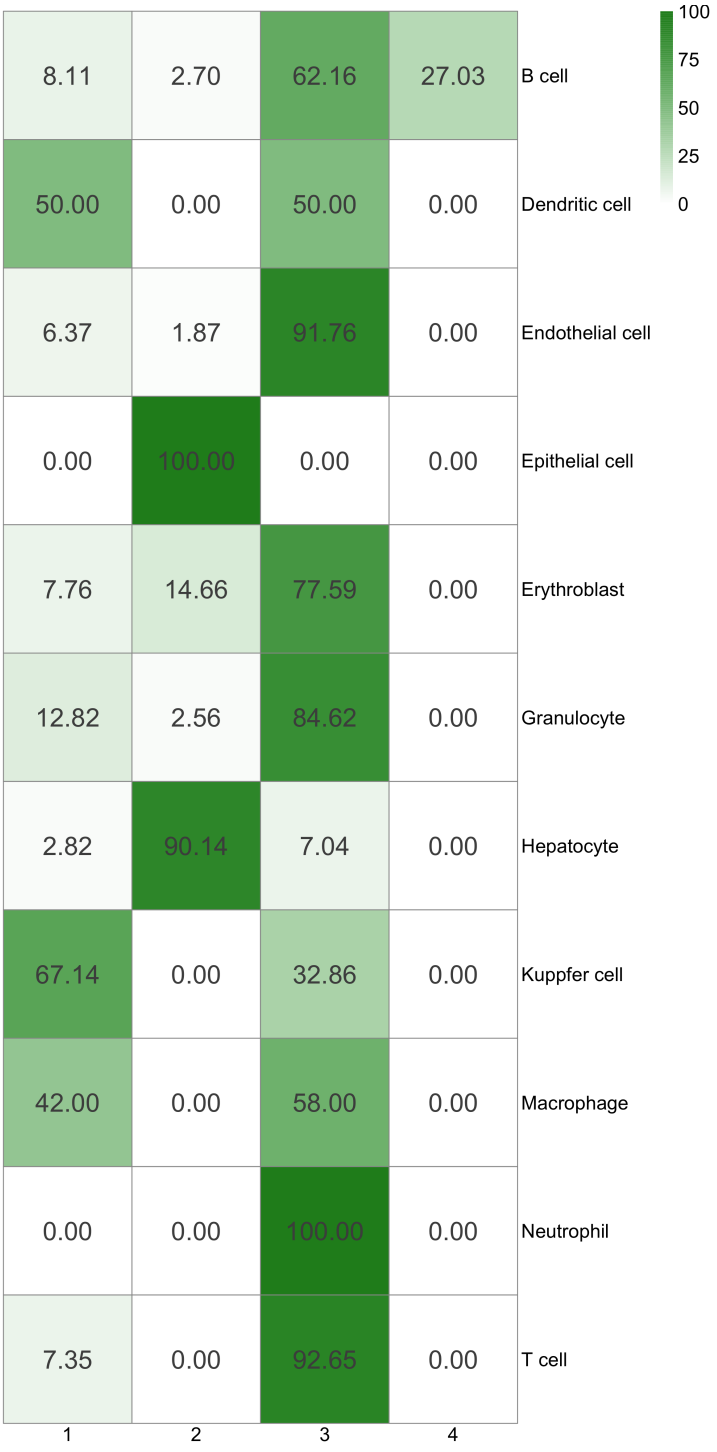

Figure 5.23: Liver dataset. Co-clustering between cell types (rows) and inferred clusters by the proposed EM algorithm (columns). Each entry a_{ij} represents the % of cells from type i that are present in the inferred cluster j. Rows sum up to 100%. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model (see Figure 5.22).

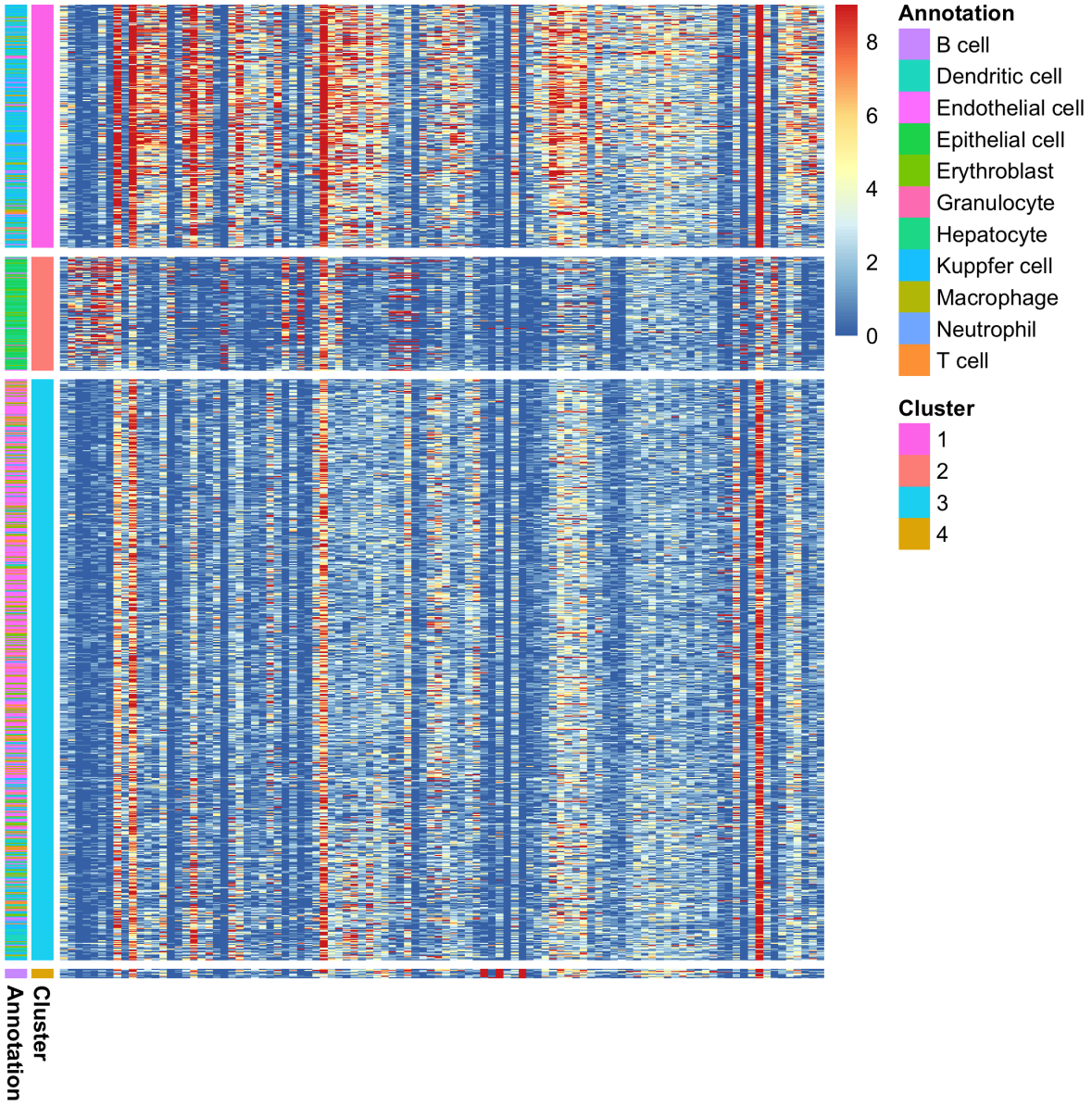

Figure 5.24: Heatmap of the liver data displaying read counts across all 1,000 randomly selected cells (rows) and all 100 selected genes (columns). Cells (rows) are ordered by their inferred cluster assignments obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model. The first column on the left shows the annotation for each cell type. Dark blue colours represent low read count values, and dark red colours represent high read count values. Note that to facilitate visualization under this colour scheme, read counts with values higher than the 95th percentile were truncated at the value of the 95th percentile.

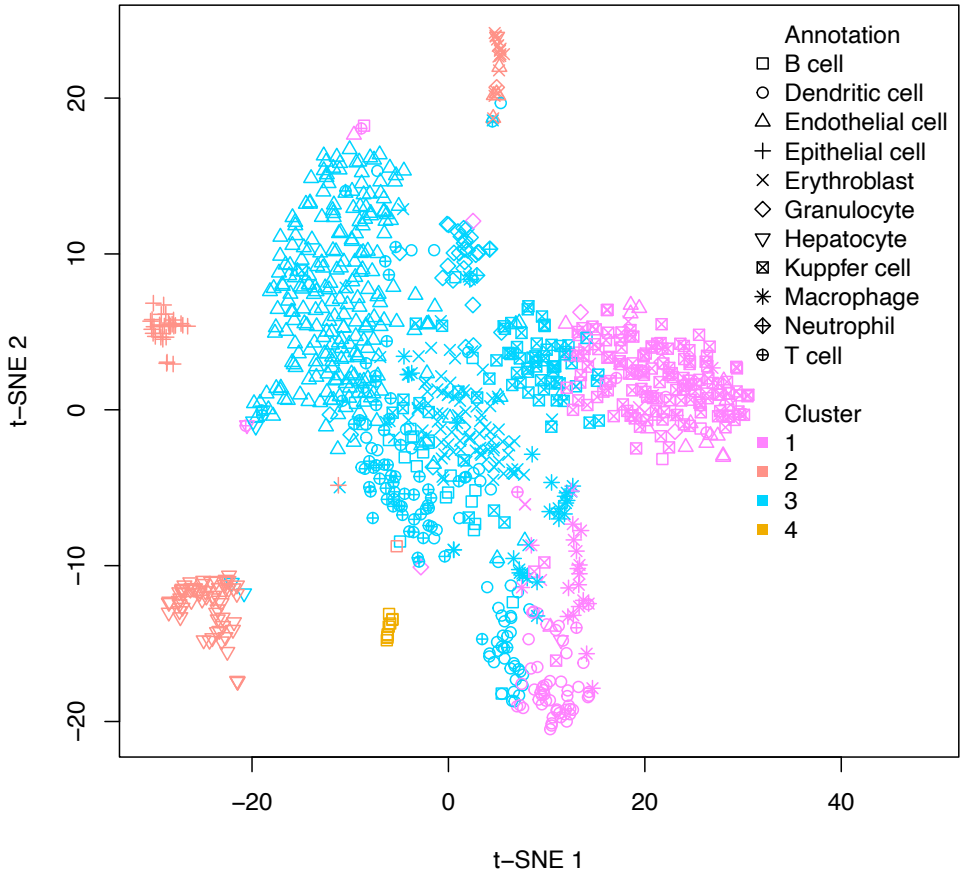

Figure 5.25: *t*-SNE plot for the liver dataset and the clustering obtained from the best EM algorithm run (*K*-means initialization and $K = 4$) under the simple ZIP model. Each point represents a cell with the shape symbol indicating the cell type label, and the colour the corresponding inferred cluster (1, 2, 3, or 4).

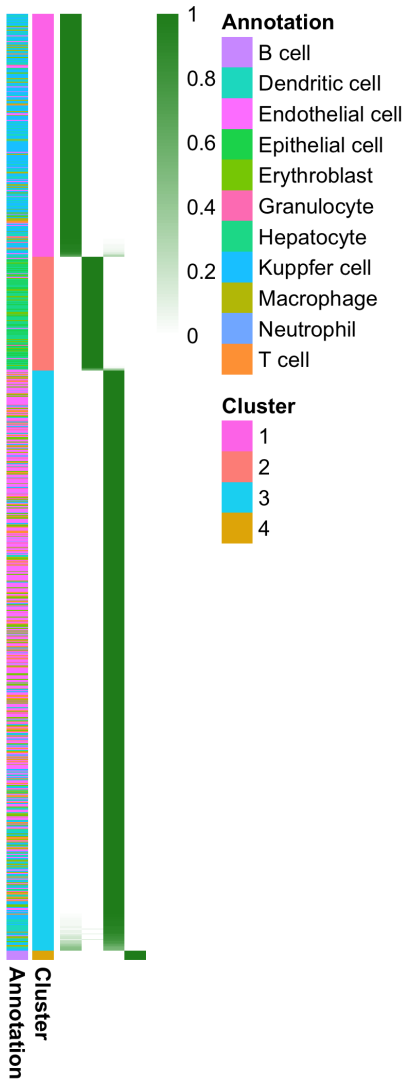

Figure 5.26: Heatmap of the $\hat{Z}_{nk}$'s for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model. Each row shows the estimated probability of a cell n belonging to each cluster k (columns). Rows are ordered by the final inferred cluster assignments determined by Eq. (3.20). The labels on the left show the assigned clusters and the cell type labels. Dark colours represent high probabilities.

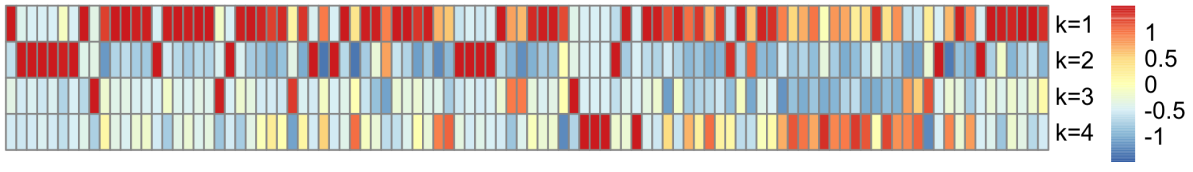

Figure 5.27: Heatmap of the $\hat{\lambda}_{gk}$'s for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 4$) under the simple ZIP model. Each row corresponds to a cluster, and each column to a gene. Dark blue colours represent low values, and dark red colours represent high values of $\hat{\lambda}_{gk}$.

5.2.2 Result of fitting the ZINB mixture model without covariates to the liver data

In this section, the results of fitting the ZINB mixture model without covariates (simple ZINB) for the liver data (over the selected 1000 cells and 100 selected mostly variable genes, same data as in Section 5.2.1) are presented.

Similarly to the previous sections, Figures 5.28 and 5.30 present the boxplots of the 32 runs of the EM algorithm for different values of K and the random and K-means initialization approaches, respectively. Figures 5.29 and 5.31 present the corresponding smallest AIC for each K for the random and K-means initialization methods, respectively. Based on the elbow method, random initialization results in $K = 4$ as the optimum number of clusters, while K-means leads to $K = 5$ as the best number of clusters. As the one from K-means clustering initialization has the smallest AIC, we choose that EM run for further analysis in this section. Therefore, for the simple ZINB mixture model, we choose the EM run with $K = 5$ from the K-means initialization method, and we present its results in what follows.

Figure 5.32 and Table 5.11 show the co-clustering and confusion matrix, respectively, between the cell types and the inferred five clusters for the best EM run ($K = 5$ clusters and K-means initialization approach) for the ZINB mixture model without covariates. From both Figure 5.32 and Table 5.11, we can see that B cells are assigned primarily to 3 clusters, with 43.24% of them in cluster 3, 35.14% in cluster 5, and 13.51% in cluster 2. Dendritic cells are primarily present in clusters 2 (85.19%), endothelial cells (91.39%) are dense in cluster 3, epithelial (90.62%) cells are mainly in the inferred cluster 1. 75% of erythroblast cells and 69.23% of Granulocyte cells fall into cluster 3. 84.51% of Hepatocyte cells are in the inferred cluster 1, Kuppfer cells are mostly assigned to clusters 3 and 4, with 19.05% and 80%, respectively. 70% of macrophage cells fall into cluster 2 and 22% of them are in cluster 3, and the remaining 8% in cluster 4. Neutrophil cells are assigned equally (50%) to the inferred clusters 3 and 5. Finally, 77.94% of T cells fall into the inferred cluster 3.

Figure 5.33 shows the heatmap of the read counts across the randomly selected 1000 cells and all 100 most variable selected genes with cells (rows) ordered by their inferred cluster assignment from the simple ZINB mixture model. The heatmap also shows the annotation for each cell's type. Figure 5.34 shows the t-SNE representation of the data in two dimensions, where different point shapes correspond to the cell type and the colors to the inferred five clusters. As in the previous sections, we can observe in Figure 5.35 that overall the proposed EM algorithm assigned cells to their clusters with high (close to 1) probabilities.

The estimated cluster proportions ($\hat{\pi}_k$'s) are presented in Table 5.12. We can see from the table that 49.157% of the cells are assigned to the inferred cluster 3, 20.986% are assigned

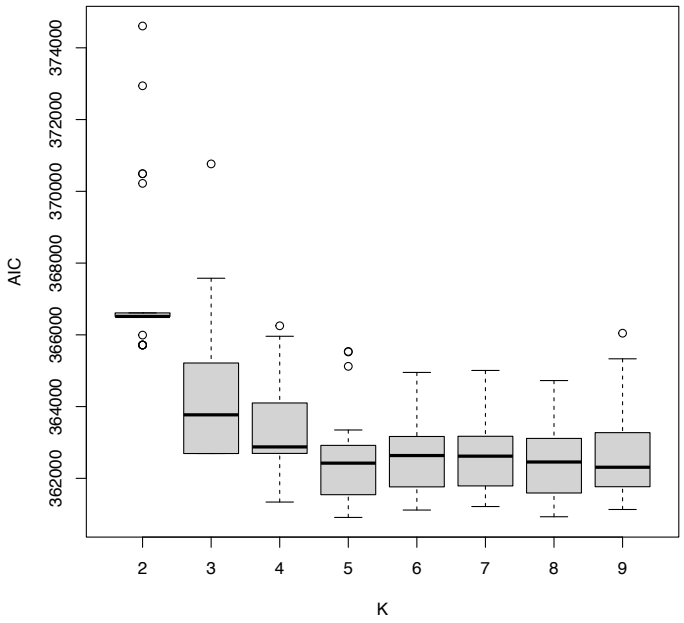

Figure 5.28: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the simple ZINB model to the liver dataset with random clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

to cluster 4, 15.235% fall into cluster 2, 9.163% are in cluster 1, and only 5.54% of them are assigned to cluster 5. Table 5.13 shows the estimated probability of always zero for each cluster k, and we can see that $\hat{\phi}_1 = 0.07266$ is the highest probability of always zero compared with the other clusters. The estimates of the size parameters for each cluster (ν_k's) are shown in Table 5.14. We can see that clusters 3, 4, and 2 have the higher estimated values ($\hat{\nu}_3 = 4.11696$, $\hat{\nu}_4 = 3.75589$, $\hat{\nu}_2 = 2.12675$), followed by the smaller estimated values for the other two clusters ($\hat{\nu}_1 = 1.93291$, $\hat{\nu}_5 = 0.66367$), demonstrating the presence of overdispersion in the liver data. Finally, Figure 5.36 shows the heatmap of the estimates of the rate parameters ($\hat{\mu}_{gk}$'s) for each cluster (rows) over the 100 selected genes (columns) when fitting the ZINB simple model to the liver data.

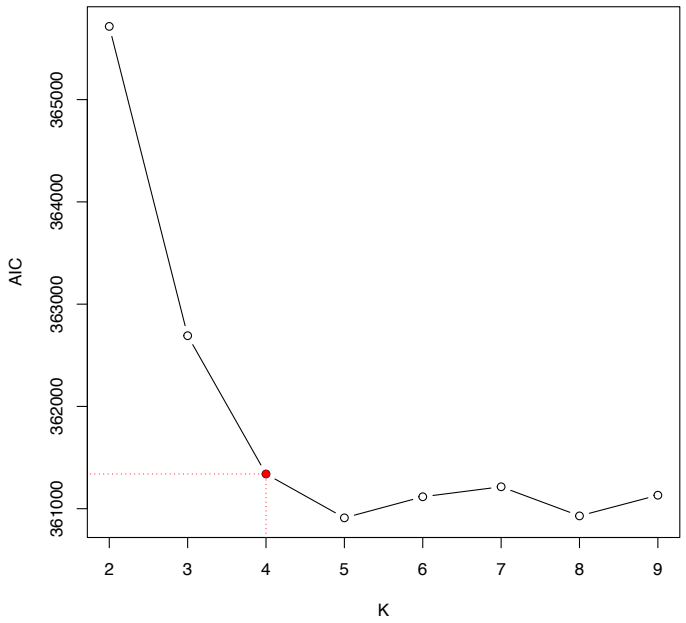

Figure 5.29: Plot of the best AIC for each K obtained from applying the EM algorithm under the simple ZINB model to the liver dataset with random clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 4$.

Table 5.11: Confusion matrix between the EM clustering result when fitting the simple ZINB model to the liver data and the cell type labels. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model (see Figure 5.32).

	Inferred cluster				
Cell type	1	2	3	4	5
B cell	0	5	16	3	13
Dendritic cell	0	92	12	3	1
Endothelial cell	0	3	244	17	3
Epithelial cell	29	0	0	0	3
Erythroblast	1	3	87	9	16
Granulocyte	0	3	27	3	6
Hepatocyte	60	1	1	0	9
Kuppfer cell	0	2	40	168	0
Macrophage	0	35	11	4	0
Neutrophil	0	0	1	0	1
T cell	2	9	53	2	2

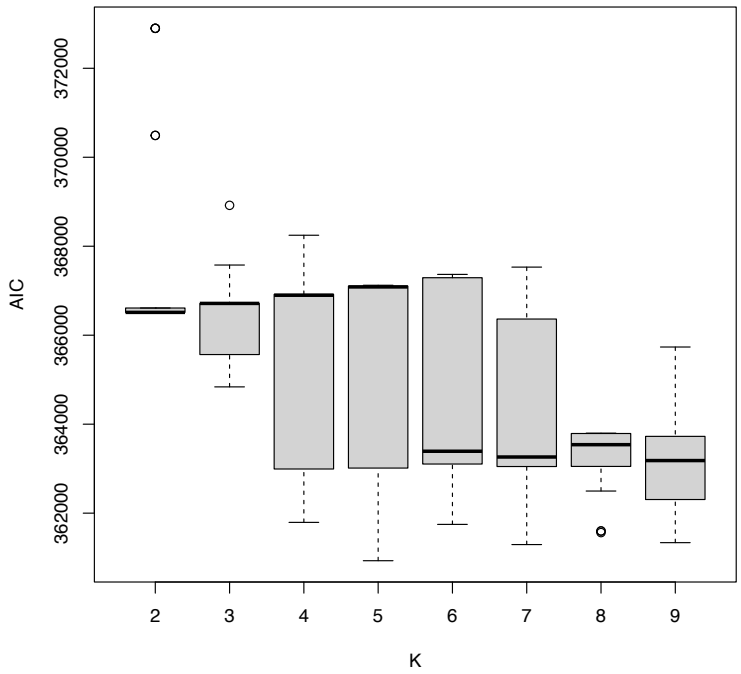

Figure 5.30: Boxplots of AIC values for different K number of clusters obtained from applying the EM algorithm under the simple ZINB model to the liver data set with K-means clustering initialization. Each boxplot contains 32 AIC values corresponding to 32 initialization runs from different seeds.

Table 5.12: Estimates of π_k for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model.

k	$\hat{\pi}_k$
1	0.09163
2	0.15235
3	0.49157
4	0.20986
5	0.05458

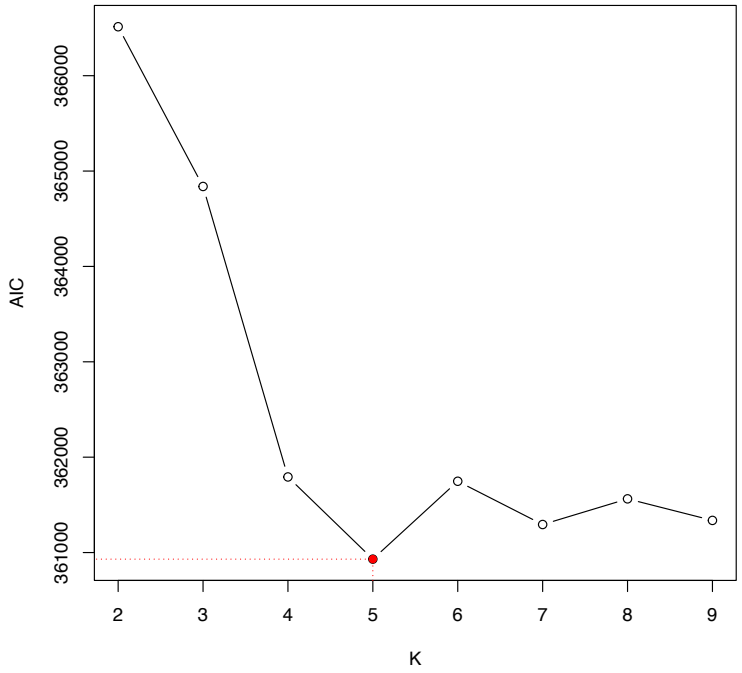

Figure 5.31: Plot of the best AIC for each K obtained from applying the EM algorithm under the simple ZINB model to the liver dataset with K-means clustering initialization. Based on the elbow method, the optimal EM run corresponds to the point in red when $K = 5$.

Table 5.13: Estimates of ϕ_k for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model.

k	$\hat{\phi}_k$
1	0.07266
2	0.00000
3	0.00005
4	0.00000
5	0.00007

Table 5.14: Estimates of ν_k for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model.

k	$\hat{\nu}_k$
1	1.93291
2	2.12675
3	4.11696
4	3.75589
5	0.66367

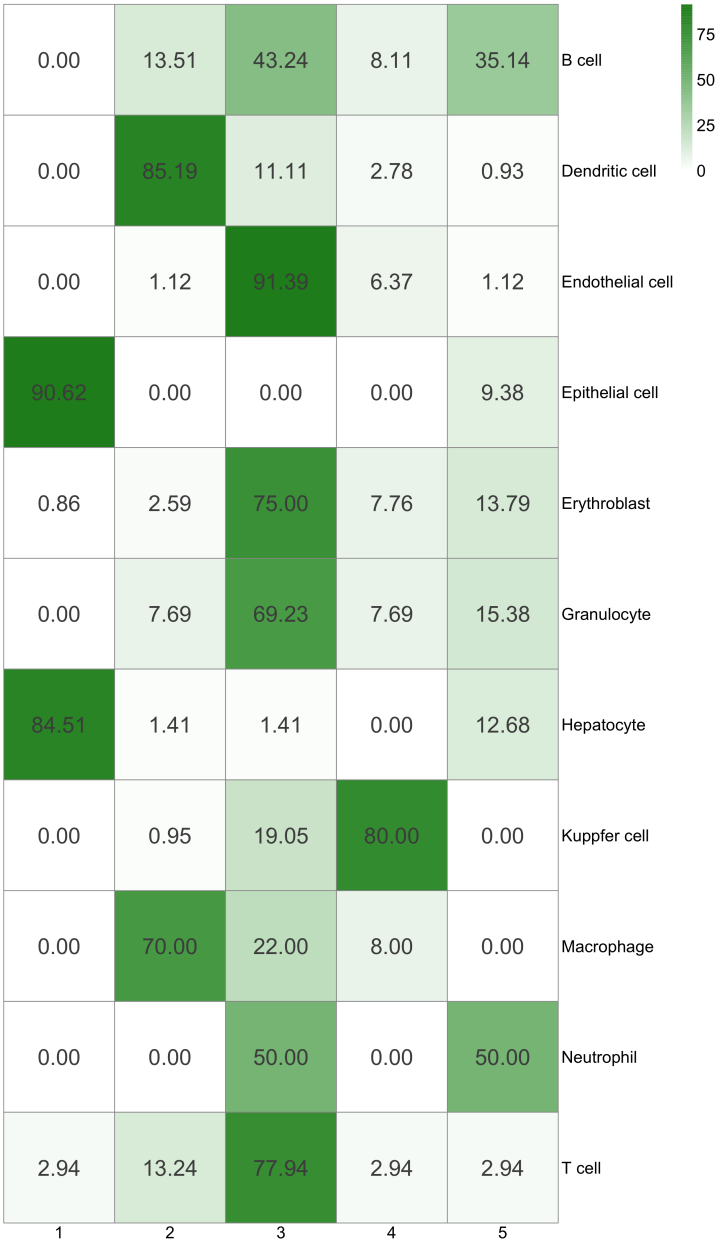

Figure 5.32: Liver dataset. Co-clustering between cell types (rows) and inferred clusters by the proposed EM algorithm (1, 2, 3, 4, and 5; columns). Each entry a_{ij} represents the % of cells from type i that are present in the inferred cluster j. Rows sum up to 100%. Inferred clusters are from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model (see Figure 5.31).

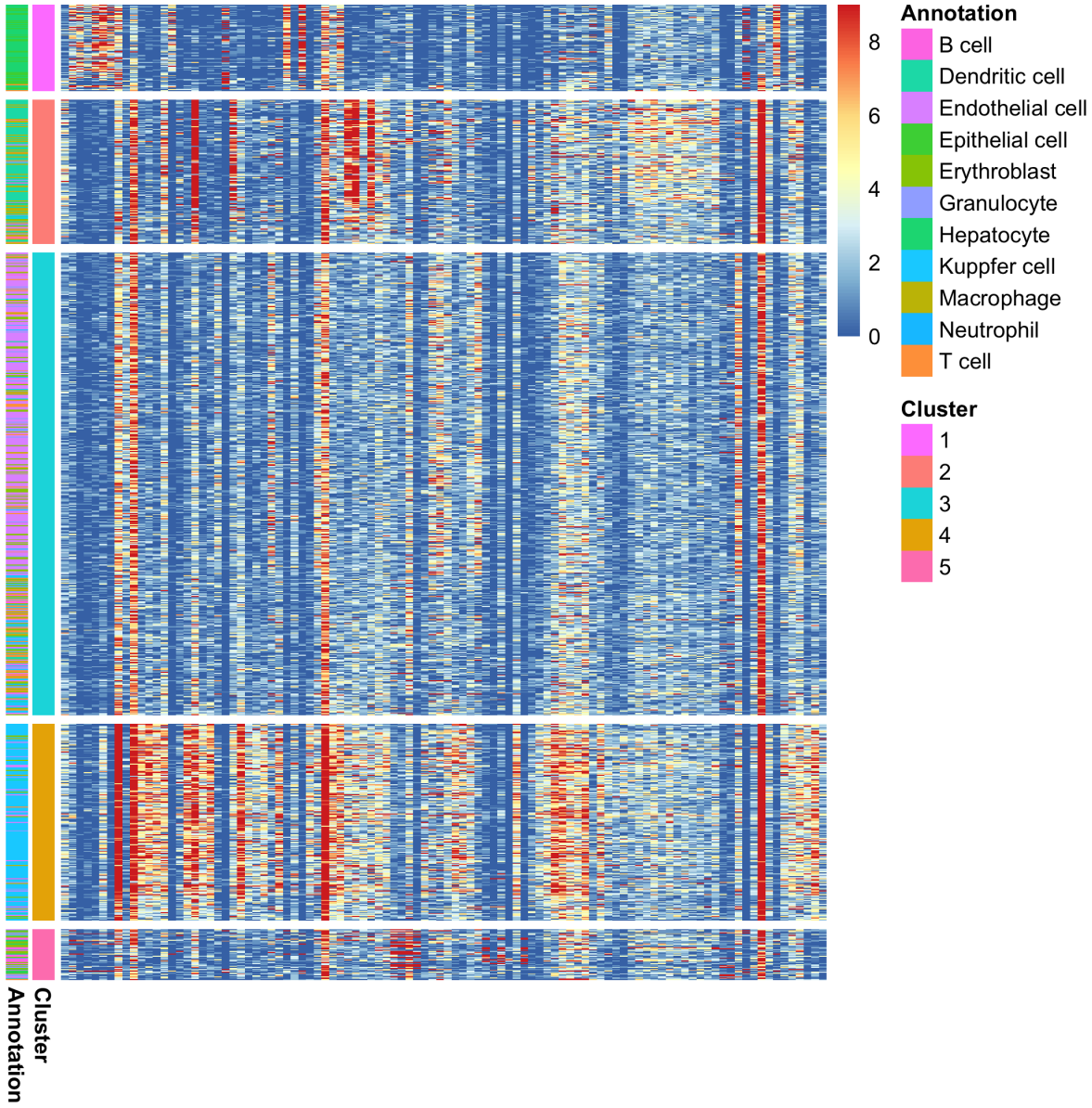

Figure 5.33: Heatmap of the liver data displaying read counts across all 1,000 randomly selected cells (rows) and all 100 selected genes (columns). Cells (rows) are ordered by their inferred cluster assignments obtained from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model. The first column on the left shows the annotation for each cell type. Dark blue colours represent low read count values, and dark red colours represent high read count values. Note that to facilitate visualization under this colour scheme, read counts with values higher than the 95th percentile were truncated at the value of the 95th percentile.

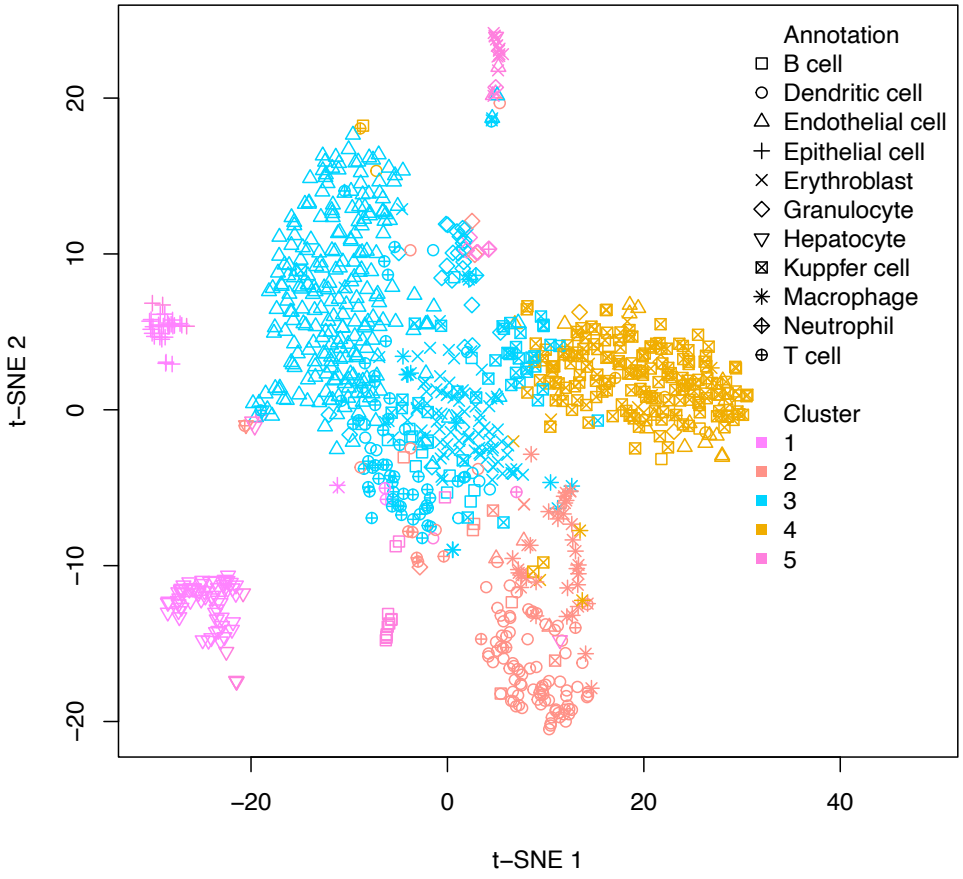

Figure 5.34: *t*-SNE plot for the liver dataset and the clustering obtained from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model. Each point represents a cell with the shape symbol indicating the cell type label, and the colour the corresponding inferred cluster (1, 2, 3, 4, or 5).

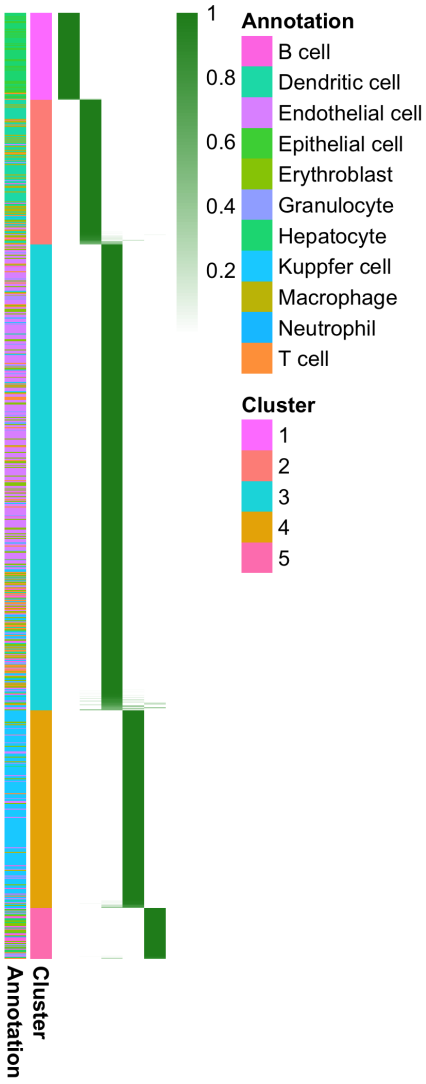

Figure 5.35: Heatmap of the $\hat{Z}_{nk}$'s for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model. Each row shows the estimated probability of a cell n belonging to each cluster k (columns). Rows are ordered by the final inferred cluster assignments determined by Eq. (3.20). The labels on the left show the assigned clusters and the cell type labels. Dark colours represent high probabilities.

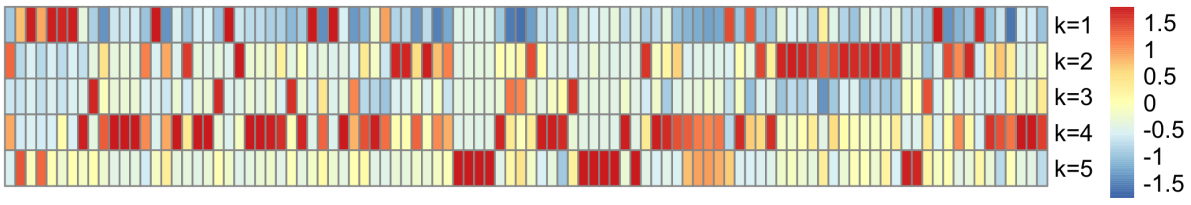

Figure 5.36: Heatmap of the $\hat{\mu}_{gk}$'s for the liver dataset obtained from the best EM algorithm run (K-means initialization and $K = 5$) under the simple ZINB model. Each row corresponds to a cluster, and each column to a gene. Dark blue colours represent low values, and dark red colours represent high values of $\hat{\mu}_{gk}$.

5.2.3 Model selection for the liver data

Table 5.15 shows the AIC values obtained from the best EM algorithm runs when fitting the simple ZIP model (ZIP mixture model without covariates) and the simple ZINB model (ZINB mixture model without covariates) to the liver data. These values correspond to the red points in Figures 5.22 and 5.31. Thus, based on the AIC values shown in Table 5.15, the simple ZINB model leads to a smaller AIC value than the simple ZIP model and, therefore, we can conclude that the simple ZINB fits the liver data better than the ZIP simple model. As shown in Section 5.2.2, the selected simple ZINB model resulted in five clusters comprising different cell types in the liver tissue.

Table 5.15: AIC values corresponding to the best EM runs when fitting the simple ZIP model and simpe ZINB model to the Liver dataset (see Figures 5.22 and 5.31).

Model	AIC
Simple ZIP	115744.1
Simple ZINB	360931

Chapter 6

Conclusion and Future Work

In this thesis, we focus on the application of the EM algorithm framework from different perspectives for the cases when the missingness in the data is evident, such as grouped data, which was the focus of Chapter 2, and for the situations when the incomplete data problem is not evident such as the finite mixture models for zero-inflated counts that were considered in Chapters 3, 4 and 5.

In Chapter 2, we considered normally distributed grouped data for univariate, bivariate, and multivariate cases. For this type of data, individual observation values are not available; instead, we only have access to non-overlapping intervals with the frequencies of observations falling into each of those intervals. Estimating the parameters of normally grouped data via the exact maximum likelihood estimation (exact MLE) method does not lead to a closed-form solution. Thus, as an alternative to numerical optimization tools such as the Newton-Raphson algorithm, we proposed estimating the mean and variance parameters of the normally distributed grouped data using the EM and MCEM algorithms, which lead to closed-form updates of the parameter along each algorithm iteration. Also, in this thesis, we provided a comprehensive approach including the exact maximum likelihood, the EM and the MCEM algorithms for parameter estimation for univariate, bivariate, and multivariate normally distributed grouped data and compared the estimated results on both real and simulated data for all the approaches.

To the author's knowledge, no other previous studies on this topic considered the problem of parameter estimation on the normally distributed grouped data from all three approaches, particularly for the bivariate and multivariate grouped data, as we did in this thesis. Therefore, the proposed statistical framework in this work can be applied to the univariate, bivariate, and multivariate grouped data following other distributions for comparing MLE parameter estimates with those obtained from EM and MCEM algorithms.

We performed simulations considering various scenarios (different sample sizes and different numbers of intervals) to study the performance of the proposed EM and MCEM estimates.

Our simulation results showed that both bias and variance of the parameter estimates decrease as the sample size increases. Moreover, based on the boxplots, for the scenarios in which n (sample size) was small compared with the number of bins/intervals (k), the exact MLE method did not work as well as the EM and MCEM approaches. That might be due to some intervals with no observations or very few observations, where, in these cases, the EM and MCEM use the expectations or average over the simulated samples. However, this needs further investigation and could be part of the potential for future work on this topic to study the changes in the parameter estimates while the ratio of n over k changes.

Furthermore, for most parameters, the results from the EM and MCEM algorithms were similar to the ones from the exact MLE, as expected by the EM convergence properties shown, for example, in Chapters 1 and 3 of McLachlan and Krishnan (2008)). Lastly, we applied the proposed EM and MCEM algorithms to the well-known Galton data, and, as expected, the estimated parameters using the EM and MCEM algorithms were very close to the ones obtained using the exact MLE method. We also found the standard errors of the mean parameter estimates for both EM and MCEM approaches to assess their accuracy.

As mentioned earlier, the exact MLE method for the parameter estimation does not have closed formulae. Moreover, this method is susceptible to the selection of initial values and the optimization method, which could be the drawback of the exact MLE method. Instead, the parameter estimation resulting from EM and MCEM approaches has closed-form formulae, and their results are close to the ones from exact MLE. Moreover, the EM and MCEM approaches are less sensitive to the initial values. However, a challenge regarding parameter estimation using the EM framework is dealing with complicated integrations, particularly for bivariate and multivariate situations. Fortunately, this thesis appropriately tackled this issue for the normally distributed grouped data by using the proper computer packages (such as tmvtnorm in R for truncated multivariate normal functions). Another solution, instead of using EM, is to find the parameter estimates through the MCEM approach to avoid dealing with complicated integrations; however, this approach is more computationally expensive than regular EM. The calculation of the complex integrations might be a challenging issue when the distribution of grouped data is not normal and no computer package is available to assist with them.

In future work on this topic, we can apply the comprehensive parameter estimation methods using all three approaches, including exact MLE, EM, and MCEM framework for grouped data arising from other distributions, or we can consider grouped data on the variables of a regression model and find the estimated coefficients of the regression model using the EM or MCEM algorithm. More discoveries regarding varying the ratio n/k for the parameter estimation performance using the proposed methods can also be considered as future work for normally distributed grouped data and grouped data arising from other distributions.

Chapters 3, 4, and 5, we studied another application of the EM algorithm for cases where the missingness of data was not evident by proposing a novel mixture-model-based clustering approach for Single-cell RNA sequencing data (scRNA-seq) and its application on both real data and simulation studies.

One important characteristic of scRNA-seq data is the excess of zeros (zero inflation), mostly due to technological noise or true biological zero. Therefore, in Chapter 3 of this thesis, we proposed a novel model-based clustering approach that takes into account the feature of zero inflation for the scRNA-seq data. Our proposed clustering algorithm is based on a mixture of zero-inflated Poisson (ZIP) or zero-inflated negative binomial distributions to cluster single-cell RNA sequencing data based on their transcriptome profiles. We derived an EM algorithm to obtain cluster assignments and estimate the parameters for each proposed ZIP and ZINB mixture model with and without covariates. According to Table 3.2 in Chapter 3, there are some model-based clustering algorithms proposed for this data; however, to our knowledge, these studies did not directly consider the feature of excess of zeros through their proposed probabilistic mixture models as we did in this study.

In Chapter 4, we studied the performance of the proposed clustering algorithm on simulated data under various scenarios for each proposed mixture model. For the ZIP mixture model without covariates, we examined the estimation of the model parameters for six different scenarios, including varying the number of cells (N), the number of genes (G), the number of clusters (K), the probabilities of cluster assignments, the probabilities of always zero, and the rate parameters across clusters. We also studied parameter estimation under six scenarios for the ZIP mixture model with only a size factor. Changes in the number of cells, genes, and clusters were studied in Scenarios 1, 2, and 3. In Scenario 4, we studied the parameter estimation when we use true values as the initialization points compared to when we initialize the EM based on K-means clustering. Changes in the probabilities of cluster assignments were studied in Scenario 5, and Scenario 6 investigated changes in the probabilities of always zero. We also investigated parameter estimation for the ZIP mixture model with one covariate considering two scenarios, one varying the number of cells and another the number of genes. We also performed simulation studies for two scenarios varying the number of cells and the number of genes under the ZINB mixture model without covariates. Finally, we studied parameter estimation under the ZINB mixture model with a size factor when the number of cells varies.

Results from the simulation studies in Chapter 4 showed that for all of the estimated parameters, the bias, standard deviation, and MSE (or MAD) decreased as N (the number of cells) increased, as expected from the convergence properties of the EM algorithm. When the number of genes (G) increased, in terms of bias, our proposed EM framework performed well, as almost all the estimated parameters were close to their true value of the parameters. The

standard deviations and MSEs (or MADs) for the estimated values of the cluster assignment probabilities and the rate parameters (for all models, including without covariates, with a size factor, and with covariates) remained almost the same when G increased. However, when estimating the probability of always zero, we observed a decrease in standard deviation when increasing G. Furthermore, for the case that the number of clusters (K) increased, almost for all cases, the standard deviations, MSE, or MAD increased while all the estimates remained close to their true values.

In Chapter 5, we applied the proposed models and EM algorithms presented in Chapter 3 to the MESC and liver datasets. For the MESC data, we considered and compared the results from the ZIP mixture models without covariates and with a size factor. Using the AIC criterion for comparing the models, we selected the ZIP mixture model with a size factor as the final choice, resulting in $K = 4$ clusters, which clustered cells mainly in 2 clusters (all cells from experiment day 4 were assigned to cluster 1 and 98.5% of the cells from experiment day 0 were assigned to cluster 4). Next, we fitted the simple ZIP and ZINB models (i.e., without covariates or size factors) to the liver tissue data. Comparing the AIC from the final fits between the simple ZIP and ZINB models led to the simple ZINB model with $K = 5$ clusters as the model that best fitted the liver data. One of the challenges in analyzing real data is the selection/filtering of genes before model fitting. Different gene filtering methods are available in the literature, and we have applied one of them in our analyses. A sensitivity analysis considering different filtering methods could be further investigated in future work.

All in all, in Chapters 3, 4, and 5 of this thesis, a novel model-based clustering algorithm was proposed for single-cell RNA sequencing data that takes into account the feature of zero-inflation for this data. The proposed model was either a mixture of zero-inflated Poisson or a mixture of zero-inflated negative binomial distributions. Parameter estimation for all proposed models was conducted through the EM framework (as one of the applications of EM when the missingness in data is not evident). Moreover, we considered and studied the proposed clustering algorithm for mixture models without and with covariates describing the rate parameter of the Poisson or negative binomial distributions. The performance of the proposed models was studied and evaluated by using different metrics (standard deviation, MSE, MAD) for a variety of scenarios, including the cases in which we varied the number of cells (N), the number of genes (G), or the number of clusters (K). Finally, comparing the performance of some of our proposed clustering algorithms based on AIC values was implemented on two real data sets that are publicly available.

Future work regarding the proposed clustering methodology includes obtaining standard errors for some parameters (for example, rate parameters and probabilities of always zero) within the EM framework, as we did in Section 2.2.3 of Chapter 2. In addition, as future work, we can

take a Bayesian approach and derive a variational Bayes algorithm to find an approximation to the posterior distribution of the parameters for the ZIP and ZINB mixture models, resulting in uncertainty measures for all parameters via credible intervals. As mentioned earlier, a sensitivity analysis on the choice of gene filtering methods prior to model fitting could be further assessed in future studies. Finally, we could consider scRNA-seq data from tumours (cancer cells), which usually undergo several copy number changes in their genome, and extend our proposed ZIP and ZINB mixture models to account for those copy number changes since these changes may affect gene expression and, therefore, cell clustering.

In summary, in this thesis, we studied the application of the EM algorithm for both cases when the incomplete data problem is evident and when it is not evident. Chapter 2 of this thesis considered the situations in which missingness is evident for the normally distributed grouped data (univariate, bivariate, and multivariate cases). A comprehensive parameter estimation approach for these data using exact MLE, EM, and MCEM was proposed in Chapter 2, followed by studying and comparing the methods on simulated and real data sets. Chapters 3, 4, and 5 of the thesis explored the application of the EM algorithm for the situations in which the missingness in data is not evident by considering and proposing a novel model-based clustering of mixtures of either zero-inflated Poisson or zero-inflated negative binomial models for scRNA-seq data. We studied the performance of all the proposed models under different simulation scenarios. Finally, some of the proposed models were applied to real data, and their results were compared with each other based on their AIC values to select the best fit.

Bibliography

T. S. Andrews and M. Hemberg. Identifying cell populations with scrnaseq. *Molecular aspects of medicine*, 59:114–122, 2018. → pages 37

S. Ayramo and T. Karkkainen. Introduction to partitioning-based clustering methods with a robust example. *University of Jyvaskyla, Reports of the Department of Mathematical Information Technology*, Series C, 2006. → pages 39

H.-H. Bock and E. Diday. *Analysis of Symbolic Data*. Berlin, Heidelberg:Springer, 2000. → pages 9

J. G. Booth and J. P. Hobert. Maximizing generalized linear mixed model likelihoods with an automated monte carlo em algorithm. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(1):265–285, 1999. → pages 22, 23

I. V. Cadez, P. Smyth, G. J. McLachlan, and C. E. McLaren. Maximum likelihood estimation of mixture densities for binned and truncated multivariate data. *Machine Learning*, 47(1): 7–34, 2002. → pages 2, 10

T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. → pages 40

G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, 2002. → pages 4, 12

J. Chen, A. Schlitzer, S. Chakarov, F. Ginhoux, and M. Poidinger. Mpath maps multi-branching single-cell trajectories revealing progenitor cell progression during development. *Nature communications*, 7(1):1–15, 2016a. → pages 41, 42

X.-d. Chen, H.-x. Shi, and X.-r. Wang. Hierarchical mixture models for zero-inflated correlated count data. *Acta Mathematicae Applicatae Sinica, English Series*, 32(2):373–384, 2016b. → pages 46

Y.-J. Chen and T. Miljkovic. From grouped to de-grouped data: A new approach in distribution fitting for grouped data. *Statistical Computation and Simulation*, 89(2):272–291, 2018. → pages 9

Y. Cheng and X. Ma. scgac: a graph attentional architecture for clustering single-cell rna-seq data. *Bioinformatics*, 38(8):2187–2193: https://doi.org/10.1093/bioinformatics/btac099, 2022. → pages 40, 41, 43

H. Cho, C. Liu, J. Park, and D. Wu. biznb: Bivariate zero-inflated negative binomial model estimator. *R package version 1.0.4*, 2019. → pages 46

N. Cloonan, A. R. Forrest, G. Kolle, B. B. Gardiner, G. J. Faulkner, M. K. Brown, D. F. Taylor, A. L. Steptoe, S. Wani, G. Bethel, et al. Stem cell transcriptome profiling via massive-scale mrna sequencing. *Nature methods*, 5(7):613–619, 2008. → pages 37

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1): 1–38, 1977. → pages 1, 5, 7, 10, 12

G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis. Single-cell rna-seq denoising using a deep count autoencoder. *NATURE COMMUNICATIONS*, 10:390: https://doi.org/10.1038/s41467–018–07931–2, 2019. → pages 41, 43

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. → pages 40, 43

J. J. Faraway. *Extending the linear model with R: generalized linear, mixed effects and non-parametric regression models*. CRC press, 2016. → pages 45

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002. → pages 40

F. Galton. *Natural Inheritance*. London: Macmillan and Co, 1889. → pages 2, 11, 21

G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. Siam, 2007. → pages 39, 40

G. H. Givens and J. A. Hoeting. *Computational Statistics*. A John Wiley & Sons, Inc., publication, 2013. → pages 1, 7, 8

D. Grun, A. Lyubimova, L. Kester, K. Wiebrands, O. Basak, N. Sasaki, H. Clevers, and A. van Oudenaarden. Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature*, 525:251–255, 2015. → pages 40, 41

D. Grun, M. J. Muraro, J.-C. Boisset, H. Clevers, E. J. de Koning, and A. van Oudenaarden. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell*, 19:266–277, 2016. → pages 40, 41

M. Guo, H. Wang, S. S. Potter, J. A. Whitsett, and Y. Xu. Sincera: a pipeline for single-cell rna-seq profiling analysis. *PLoS computational biology*, 11(11):e1004575, 2015. → pages 41, 42

X. Han, R. Wang, Y. Zhou, L. Fei, H. Sun, S. Lai, A. Saadatpour, Z. Zhou, H. Chen, Fang, D. Huang, Y. Xu, W. Huang, M. Jiang, X. Jiang, J. Mao, Y. Chen, C. Lu, J. Xie, Q. Fang, Y. Wang, R. Yue, T. Li, H. Huang, S. H. Orkin, G.-C. Yuan, M. Chen, and G. Guo. Mapping the mouse cell atlas by microwell-seq. *Cell*, 172:1091–1107: https://doi.org/10.1016/j.cell.2018.02.001, 2018. → pages 149, 169, 171

J. A. Hanley. Transmuting women into men: Galton's family data on human stature. *The American Statistician*, 58(3):237–243, 2004. → pages 21

D. F. Heitjan. Inference from grouped continuous data: A review. *Statistical Science*, 4(2): 164–179, 1989. → pages 10

D. F. Heitjan. Regression with bivariate grouped data. *Biometrics*, 47(2):549–562, 1991. → pages 2, 10

L. Hi, H. Sieling, and T. Aspelmeier. Fdrseg: Fdr-control in multiscale change-point segmentation. *R package version 1.0-3*, 2017. → pages 71

J. M. Hilbe. *Negative binomial regression*. Cambridge University Press, 2011. → pages 63, 64

M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Can shared-neighbor distances defeat the curse of dimensionality? In *Scientific and Statistical Database Management: 22nd International Conference, SSDBM 2010, Heidelberg, Germany, June 30–July 2, 2010. Proceedings 22*, pages 482–500. Springer, 2010. → pages 42

M.-W. Hu, D. W. Kim, S. Liu, D. J. Zack, S. Blackshaw, and J. Qian. Panoview: An iterative clustering method for single-cell rna sequencing data. *PLoS computational biology*, 15(8): e1007040, 2019. → pages 41, 43

J. Z. Huang, X. Wang, X. Wu, and L. Zhou. Estimation of a probability density function using interval aggregated data. *Statistical Computation and Simulation*, 86(15):1–13, 2016. → pages 9

S. Jackman, A. Tahk, A. Zeileis, C. Maimone, J. Fearon, and Z. Meers. pscl: Political science computational laboratory. *R package version 1.5.5*, 2020. → pages 46

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2017. → pages 39

Z. Ji and H. Ji. Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic Acids Research*, 44(13):e117, 2016. → pages 3, 41, 42

H. Jiang, L. L. Sohn, H. Huang, and L. Chen. Single cell clustering based on cell-pair differentiability correlation and variance analysis. *Bioinformatics*, 34 (21):3684–3694: doi:10.1093/bioinformatics/bty390, 2018. → pages 41, 42

M. Jochmann. Zic: Bayesian inference for zero-inflated count models. *R package version 0.9.1*, 2017. → pages 46

P. Jones and G.J.McLachlan. Algorithm AS 254: Maximum likelihood estimation from grouped and truncated data with finite normal mixture models. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 39(2):273–282, 1990. → pages 205

P. V. Kharchenko, L. Silberstein, and D. T. Scadden. Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740–742, 2014. → pages 38

J. Kim, D. E. Stanescu, and K. J. Won. Cellbic: bimodality-based top-down clustering of single-cell rna sequencing data reveals hierarchical structure of the cell type. *Nucleic Acids Research*, 46 (21):e124:doi: 10.1093/nar/gky698, 2018. → pages 41, 42

V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, et al. Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483–486, 2017. → pages 40, 41

A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner. Droplet barcoding for single cell transcriptomics applied to embryonic stem cells. *Cell*, 161 (5):1187–1201: doi:10.1016/j.cell.2015.04.044, 2015. → pages 38, 149, 151, 169

D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21(1):1–35, 2020. → pages 38

L. P. Lan Jiang, Huidong Chen and G.-C. Yuan. Giniclust: detecting rare cell types from single-cell gene expression data with gini index. *Genome Biology*, 17(144), 2016. → pages 41, 43

J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, D. A. El-ad, M. D. Tadmor, O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, et al. Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197, 2015. → pages 41, 42

R.-Y. Li, Z. Wang, J. Guan, S. Zhou, S. Member, and IEEE. Effectively clustering single cell rna sequencing data by sparse representation. *LATEX CLASS FILES*, 14 (8):DOI 10.1109/TCBB.2021.3128576, 2015. → pages 41, 43

H. K. Lim, W. K. Li, and L. Philip. Zero-inflated poisson regression mixture model. *Computational Statistics & Data Analysis*, 71:151–158, 2014. → pages 46

Y. Liu, J. L. Warren, and H. Zhao. A hierarchical bayesian model for single-cell clustering using rna-sequencing data. *The annals of applied statistics*, 13(3):1733, 2019. → pages 3, 41, 42

Z. Liu. Clustering single-cell rna-seq data with regularized gaussian graphical model. *Genes*, 12 (2):311:https://doi.org/10.3390/genes12020311, 2021. → pages 41, 43

R. Lopez, J. Regier, M. B. Cole, M. I. Jordan1, and N. Yosef. Deep generative modeling for single-cell transcriptomics. *Nat Methods*, 15(12):1053–1058: doi:10.1038/s41592–018–0229–2, 2018. → pages 40, 41, 43

T. A. Louis. Finding the observed information matrix when using the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):226–233, 1982. → pages 20

O. Lyashevska, D. J. Brus, and J. van der Meer. Mapping species abundance by a spatial zero-inflated p oisson model: a case study in the w adden s ea, the n etherlands. *Ecology and evolution*, 6(2):532–543, 2016. → pages 45

I. C. Macaulay and T. Voet. Single cell genomics: Advances and future perspectives. *PLOS Genetics*, 10(1), 2014. → pages 37

B. G. Manjunath and S. Wilhelm. Moments calculation for the doubly truncated multivariate normal density. *Journal of Behavioral Data Science*, 1(1):17–33, 2021. doi: 10.35566/jbds/v1n1/p2. → pages 16, 19, 207, 208

C. E. McCulloch. Maximum likelihood algorithms for generalized linear mixed models. *Journal of the American Statistical Association*, 92(437):162–170, 1997. → pages 22, 23

G. J. McLachlan and P. N. Jones. Fitting mixture models to grouped and truncated data via the em algorithm. *Biometrics*, 44(2):571–578, 1988. → pages 2, 10, 12

G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. New Jersey: John Wiley & Sons, 2008. → pages 1, 3, 4, 5, 7, 8, 10, 12, 13, 20, 21, 24, 71, 191

X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993. → pages 7

C. Minoiu and S. G. Reddy. Estimating poverty and inequality from grouped data: How well do parametric methods perform? *Income Distribution*, 18(2):160–178, 2009. → pages 9

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5(7):621–628, 2008. → pages 37

M. T. P. Lin and J. W. K. Ho. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biology*, 18(59), 2017. → pages 40, 41

P. Papastamoulis. poisson.glm.mix: Fit high dimensional mixtures of poisson glms. *R package version 1.3*, 2022. → pages 46

C. Park. A quantile implementation of the em algorithm and applications to parameter estimation with interval data. *In Technical Report TR2006-05-CP, Department of Mathematical Sciences, Clemson, SC: Clemson University*, 2006. → pages 12

S. Park and H. Zhao. Spectral clustering based on learning similarity matrix. *Bioinformatics*, 34(12):2069–2076: doi: 10.1093/bioinformatics/bty050, 2018. → pages 41, 43

E. Pierson and C. Yau. Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(241), 2015. → pages 44

S. Pilosof, C. W. Dick, C. Korine, B. D. Patterson, and B. R. Krasnov. Effects of anthropogenic disturbance and climate on patterns of bat fly parasitism. *PloS one*, 7(7):e41487, 2012. → pages 45

S. Prabhakaran, E. Azizi, A. Carr, , and D. Pe'er. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. *JMLR Workshop Conf Proc*, 48:1070–1079, 2016. → pages 3, 41, 42

R. Qi, A. Ma, Q. Ma, and Q. Zou. Clustering and classification methods for single-cell rna-sequencing data. *Briefings in Bioinformatics*, 21 (4):1196–1208. doi: 10.1093/bib/bbz062, 2020. → pages 160

P. Qiu. Embracing the dropouts in single-cell rna-seq analysis. *Nature Communication*, 11 (1169), 2020. → pages 44

A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007. → pages 73

A.-E. Saliba, A. J. Westermann, S. A. Gorski, and J. Vogel. Single-cell rna-seq: advances and future challenges. *Nucleic acids research*, 42(14):8845–8860, 2014. → pages 37

R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015. → pages 41, 42

R. S. Satija. R toolkit for single cell genomics: single cell integration in seurat v3.0. *satijalab.org-https://satijalab.org/seurat/*, 2015. → pages 41, 42

F. SHI and H. HUANG. Identifying cell subpopulations and their genetic drivers from single-cell rna-seq data using a biclustering approach. *COMPUTATIONAL BIOLOGY*, 24:663–674: DOI: 10.1089/cmb.2017.0049, 2017. → pages 41, 42

M. B. Stewart. On least squares estimation when the dependent variable is grouped. *The Review of Economic Studies*, 50(4):737–753, 1983. → pages 10

Z. Sun, T. Wang, K. Deng, X.-F. Wang, R. Lafyatis, Y. Ding, M. Hu, and W. Chen. Dimm-sc: a dirichlet mixture model for clustering droplet-based single cell transcriptomic data. *Bioinformatics*, 34(1):139–146, 2018. → pages 3, 41, 42

G. M. Tallis. Approximate maximum likelihood estimates from grouped data. *Technometrics*, 599-606:9(4), 1967. → pages 10

A. Tanay and A. Regev. Scaling single-cell genomics from phenomenology to mechanism. *Nature*, 541(7637):331–338, 2017. → pages 37

B. Tasic, V. Menon, T. N. Nguyen, T. K. Kim, T. Jarsky, Z. Yao, B. Levi, L. T. Gray, S. A. Sorensen, T. Dolbeare, et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature neuroscience*, 19(2):335–346, 2016. → pages 41, 42

M. Teimouri. Em algorithm for mixture of skew-normal distributions fitted to grouped data. *Journal of Applied Statistics*, pages 1154–79. DOI: 10.1080/02664763.2020.1759032, 2020. → pages 2, 10

T. Tian, J. Wan, Q. Song, and Z. Wei. Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4):191–198, 2019. doi: 10.1038/s42256-019-0037-0. URL https://doi.org/10.1038/s42256-019-0037-0. → pages 41, 43, 44

T. Tian, J. Zhang, X. Lin, Z. Wei, and H. Hakonarson. Model-based deep embedding for constrained clustering analysis of single cell rna-seq data. *Nature Communications*, 12 (1):1873, 2021. doi: 10.1038/s41467-021-22008-3. URL https://doi.org/10.1038/s41467-021-22008-3. → pages 41, 43, 44

C. Trapnell. Defining cell types and states with single-cell genomics. *Genome research*, 25 (10):1491–1498, 2015. → pages 37

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. → pages 151

J. I. Velez and J. C. Correa. Bootstrap-based inference for grouped data. *Revista de la Facultad de Ciencias*, 4(2):74–82, 2015. → pages 9, 10

B. Wang, J. Zhu, E. Pierson, D. Ramazzotti, and S. Batzoglou. Visualization and analysis of single-cell rnarna-seq data by kernel-based similarity learning. *Nature Methods*, 14(4), 2017. → pages 40, 41

Z. Wang, A. Zeileis, S. Jackman, B. Ripley, and P. Breheny. mpath: Regularized linear models. *R package version 0.4-2.22*, 2022. → pages 46

G. C. Wei and M. A. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85 (411):699–704, 1990a. → pages 8, 10, 13

G. C. Wei and M. A. Tanner. Posterior computations for censored regression data. *Journal of the American Statistical Association*, 85(411):829–839, 1990b. → pages 8, 10, 13

N. Wei, Y. Nie, L. LiuID, X. ZhengID, and H.-J. Wu. Secuer: Ultrafast, scalable and accurate clustering of single-cell rna-seq data. *PLOS Computational Biology*, page https://doi.org/10.1371/journal.pcbi.1010753, 2022. → pages 41, 43

J. Wengrzik and J. Timm. Comparing several methods to fit finite mixture models to grouped data by the em algorithm. *Proceedings of the World Congress on Engineering*, I:958–966, 2011. → pages 10

F. A. Wolf, P. Angerer, and F. J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):1–5, 2018. → pages 41, 42

M. S. Workie and A. G. Azene. Bayesian zero-inflated regression model with application to under-five child mortality. *Journal of Big Data*, 8(4), 2021. → pages 45

X. Wu and J. M. Perloff. Gmm estimation of a maximum entropy distribution with interval data. *Econometrics*, 138:532–546, 2007. → pages 9

J. Xia, J. Mi, and Y. Zhou. On the existence and uniqueness of the maximum likelihood estimators of normal and lognormal population parameters with grouped data. *Probability and Statistics*, 2009, Article ID 310575:16 pages, 2009. → pages 12

C. Xu and Z. Su. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12):1974–1980, 2015. → pages 41, 42

X. Xue, Q. Qi, D. Sotres-Alvarez, S. C. Roesch, M. M. Llabre, S. A. Bainter, Y. Mossavar-Rahmani, R. Kaplan, and T. Wang. Modeling daily and weekly moderate and vigorous physical activity using zero-inflated mixture poisson distribution. *Statistics in Medicine*, 39 (30):4687–4703, 2020. → pages 45

T. Yee and C. Moler. Vgam: Vector generalized linear and additive models. *R package version 1.1-7*, 2022. → pages 46

B. Yu, C. Chen, R. Qi, R. Zheng, P. J. Skillman-Lawrence, X. Wang, A. Ma, and H. Gu. scgmai: a gaussian mixture model for clustering single-cell rna-seq data based on deep autoencoder. *Briefings in Bioinformatics*, 22(4):1–10: https://doi.org/10.1093/bib/bbaa316, 2021. → pages 41, 43

A. Zeisel, A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015. → pages 41, 42, 149, 169

A. W. Zhang, C. O'Flanagan, E. A. Chavez, J. L. Lim, N. Ceglia, A. McPherson, M. Wiens, P. Walters, T. Chan, B. Hewitson, D. Lai, A. Mottok, C. Sarkozy, L. Chong, T. Aoki, X. Wang, A. P. Weng, J. N. McAlpine, S. Aparicio, C. Steidl, and S. P. S. Kieran R Campbell and. Probabilistic cell-type assignment of single-cell rna-seq for tumor microenvironment profiling. *Nat Method*, 16 (10):1007–1015. doi:10.1038/s41592–019–0529–1., 2019. → pages 3, 54

Q. Zhang and G. Y. Yi. Zipbayes: Bayesian methods in the analysis of zero-inflated poisson model. *R package version 1.0.2*, 2021. → pages 46

J. žurauskienė and C. Yau. pcareduce: Hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, 17(140), 2016. → pages 40, 41

Appendix A

Appendix of Ch 2

A.1 Expectations for the E-step of the EM algorithm for univariate normal grouped data

In Section 2.2.1, to find the updated estimates of the parameters, we have to calculate the following expectations:

$$E_{\theta^{(p)}}\Big(X|X \in \mathcal{X}_i\Big)$$

and

$$E_{\theta^{(p)}}\Big((X-\mu^{(p+1)})^2|X \in \mathcal{X}_i\Big),$$

with respect to the density $f(x;\theta)/P_i(\theta)$, where $f(x;\theta)$ is the univariate normal distribution. Let $\mathcal{X}_i = (a,b)$, these expectations can be obtained as follows:

$$\begin{aligned}
E_{\theta^{(p)}}\Big[X|X \in \mathcal{X}_i\Big] = [F(b)-F(a)]E(X) &= \int_a^b x\frac{1}{\sqrt{2\pi}\sigma^{(p)}}e^{-\frac{1}{2\sigma^{2(p)}}(x-\mu^{(p)})^2}dx \\
&= \int_{a^*}^{b^*}(\sigma^{(p)}t+\mu^{(p)})\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}t^2}dt \\
&= \sigma^{(p)}\int_{a^*}^{b^*} t\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}t^2}dt + \mu^{(p)}\int_{a^*}^{b^*}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}t^2}dt \\
&= \mu^{(p)}\Big[F(b^*)-F(a^*)\Big] - \sigma^{(p)}\Big[f(b^*)-f(a^*)\Big],
\end{aligned}$$

where $t = \frac{x-\mu^{(p)}}{\sigma^{(p)}}$, $a^* = \frac{a-\mu^{(p)}}{\sigma^{(p)}}$, and $b^* = \frac{b-\mu^{(p)}}{\sigma^{(p)}}$. See also Jones and G.J.McLachlan (1990).

$$
\begin{aligned}
[F(b)-F(a)]E(X-\mu^{(p+1)})^2 &= \int_a^b (x-\mu^{(p+1)})^2 \frac{1}{\sqrt{2\pi}\sigma^{(p)}} e^{-\frac{1}{2\sigma^2(p)}(x-\mu^{(p)})^2} dx = \\
&\int_a^b x^2 \frac{1}{\sqrt{2\pi}\sigma^{(p)}} e^{-\frac{1}{2\sigma^2(p)}(x-\mu^{(p)})^2} dx - \\
&2\mu^{(p+1)} \int_a^b x \frac{1}{\sqrt{2\pi}\sigma^{(p)}} e^{-\frac{1}{2\sigma^2(p)}(x-\mu^{(p)})^2} dx. \\
&\mu^{2(p+1)} \int_a^b \frac{1}{\sqrt{2\pi}\sigma^{(p)}} e^{-\frac{1}{2\sigma^2(p)}(x-\mu^{(p)})^2} dx.
\end{aligned}
$$

Now let $a^* = \frac{a-\mu^{(p)}}{\sigma^{(p)}}$, $b^* = \frac{b-\mu^{(p)}}{\sigma^{(p)}}$, $t = \frac{x-\mu^{(p)}}{\sigma^{(p)}}$, and using

$$
\begin{aligned}
&\int_a^b x^2 \frac{1}{\sqrt{2\pi}\sigma^{(p)}} e^{-\frac{1}{2\sigma^2(p)}(x-\mu^{(p)})^2} dx = \\
&\int_{a^*}^{b^*} (\sigma^{(p)}t + \mu^{(p)})^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt = \\
&\sigma^{2(p)} \int_{a^*}^{b^*} t^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt + \mu^{2(p)} \int_{a^*}^{b^*} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt + \\
&2\sigma^{(p)}\mu^{(p)} \int_{a^*}^{b^*} t \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt \\
&\sigma^{2(p)} \Big[[F(b^*) - F(a^*)] - [b^* f(b^*) - a^* f(a^*)] \Big] + \\
&\mu^{2(p)} [F(b^*) - F(a^*)] - 2\sigma^{(p)}\mu^{(p)} [f(b^*) - f(a^*)],
\end{aligned}
$$

we obtain

$$
\begin{aligned}
E(X-\mu^{(p+1)})^2 &= \frac{1}{F(b^*) - F(a^*)} \times \\
&\sigma^{2(p)} \Big[\Big(F(b^*) - F(a^*) \Big) - \Big(b^* f(b^*) - a^* f(a^*) \Big) \Big] + \\
&\Big(\mu^{(p+1)} - \mu^{(p)} \Big)^2 \Big[F(b^*) - F(a^*) \Big] + \\
&2\sigma^{(p)} \Big[\Big(\mu^{(p+1)} - \mu^{(p)} \Big) \Big[f(b^*) - f(a^*) \Big] \Big].
\end{aligned}
$$

A.2 Expectations for the E-step of the EM algorithm for multivariate normal grouped data

In Section 2.2.2, to find the estimates of the parameters using the EM approach, the expectations in the general form of

$$E_{\Theta^{(p)}}\Big[X_i\Big|(X_1,\ldots,X_d)\in(\mathcal{X}_1,\ldots\mathcal{X}_d)\Big]$$

and

$$E_{\Theta^{(p)}}\Big[(X_i-\mu^{(p+1)})(X_j-\mu^{(p+1)})^T\Big|(X_1,\ldots,X_d)\in(\mathcal{X}_1,\ldots\mathcal{X}_d)\Big]$$

should be found. In what follows we present the main steps of the calculations. For further details see Manjunath and Wilhelm (2021).

Using the moment generation function for the multivariate normal, the expectations can be obtained as follows:

$$E(X_i)=\frac{\partial m(t)}{\partial t_i}|_{t=0}=\sum_{k=1}^{d}\sigma_{i,k}(F_k(a_k)-F_k(b_k)), \tag{A.1}$$

where

$$F_i(x)=\int_{a_1^*}^{b_1^*}\cdots\int_{a_{i-1}^*}^{b_{i-1}^*}\int_{a_{i+1}^*}^{b_{i+1}^*}\cdots\int_{a_d^*}^{b_d^*}\phi_{\alpha\Sigma}(x_1,\ldots,x_{i-1},x,x_{i+1},\ldots,x_d)dx_d\ldots dx_{i+1}dx_{i-1}\ldots dx_1$$

$$a_i^*=a_i-\sum_{k=1}^{d}\sigma_{i,k}t_k,$$

$$b_i^*=b_i-\sum_{k=1}^{d}\sigma_{i,k}t_k,$$

and at $t_k=0$, for all $k=1,2,\ldots,d$, $a_i^*=a_i$ and $b_i^*=b_i$. It should be noted that in $F_i(x)$, $\phi_{\alpha,\Sigma}(x)$ arises from:

$$\phi_{\alpha,\mu,\Sigma}(x)=\begin{cases}\frac{\phi_{\mu,\Sigma}(x)}{P(a\le X\le b)} & \text{for } a\le x\le b,\\ 0 & \text{otherwise.}\end{cases}$$

$$m(t)=\exp\Big(\frac{1}{2}t^T\Sigma t\Big)\Phi_{\alpha\Sigma},$$

where for $\zeta = \Sigma t$ we have:

$$\Phi_{\alpha\Sigma} = \frac{1}{\alpha(2\pi)^{(\frac{d}{2})}|\Sigma|^{\frac{1}{2}}} \int_{a-\zeta}^{b-\zeta} exp\Big(-\frac{1}{2}x^T\Sigma^{-1}x\Big)dx$$

for the case of $\mu = 0$ and $\alpha = P(a < X < b)$. Considering $Y \sim N(\mu, \Sigma)$ with $a^* < y < b^*$, then using the transformation, $X = Y - \mu \sim N(0, \Sigma)$, which changes within the range of $a = a^* - \mu < x < b^* - \mu = b$. So, for the general case μ (not the case of $\mu = 0$), using the transformation idea for the expectation, we will have $E(Y) = E(X) + \mu$, then for the multivariate normal expectation we obtain:

$$E(Y_i) = \sum_{k=1}^{d} \sigma_{i,k}(F_k(a_k) - F_k(b_k)) + \mu_i$$

Similarly, we can show that for all $t_k = 0$, $k = 0, \ldots, d$ (see Manjunath and Wilhelm (2021)) we obtain:

$$\begin{aligned} E(X_iX_j) &= \frac{\partial^2 m(t)}{\partial t_j \partial t_i}|_{t=0} = \sigma_{i,j} + \sum_{k=1}^{d} \sigma_{i,k} \frac{\sigma_{j,k}(a_kF_k(a_k) - b_kF_k(b_k))}{\sigma_{k,k}} \\ &+ \sum_{k=1}^{d} \sigma_{i,k} \sum_{q\neq k} (\sigma_{j,q} - \frac{\sigma_{k,q}\sigma_{j,k}}{\sigma_{k,k}}) \Bigg[\Big(F_{k,q}(a_k, a_q) - F_{k,q}(a_k, b_q)\Big) \\ &- \Big(F_{k,q}(b_k, a_q) - F_{k,q}(b_k, b_q)\Big) \Bigg] \end{aligned}$$

where

$$F_{k,q}(x, y) = \int_{a_1^*}^{b_1^*} \cdots \int_{a_{k-1}^*}^{b_{k-1}^*} \int_{a_{k+1}^*}^{b_{k+1}^*} \cdots \int_{a_{q-1}^*}^{b_{q-1}^*} \int_{a_{q+1}^*}^{b_{q+1}^*} \cdots \int_{a_d^*}^{b_d^*} \phi_{\alpha\Sigma}(x, y, x_{-k,-q}) dx_{-k,-q},$$

and

$$x_{-k,-q} = (x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_{q-1}, x_{q+1}, \ldots, x_d)^T$$

for $k \neq q$. As the covariance matrix is invariant to the shift of the variables we will have

$$cov(Y_i, Y_j) = cov(X_i, X_j) = E(X_iX_j) - E(X_i)E(X_j)$$

All of these expectations are calculated at the current state of the parameters $\mu^{(p)}$ and $\Sigma^{(p)}$.

Appendix B

Appendix of Ch 4

B.1 Simulation scenarios for the ZIP mixture model without covariates

B.1.1 Scenario 1

Table B.1: Scenario 1: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by N, across the datasets simulated from the settings described in Table 4.2.

N	Mean	SD
12	16.78125	8.02588
60	6.84766	0.91407
120	5.96094	0.41354
600	5.00000	0.00000
1200	5.00000	0.00000

Table B.2: Scenario 1: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by N, across the datasets simulated from the settings described in Table 4.2.

N	Mean	SD
12	0.03111	0.01597
60	0.05055	0.01114
120	0.08514	0.01712
600	0.32546	0.03525
1200	0.62588	0.03852

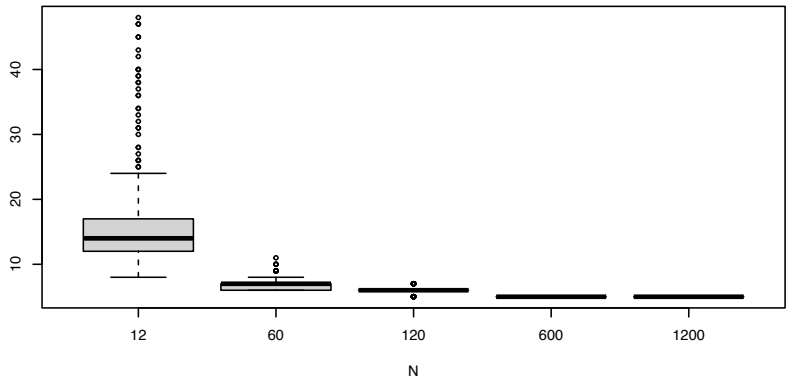

Figure B.1: Scenario 1: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.2. See also Table B.1.

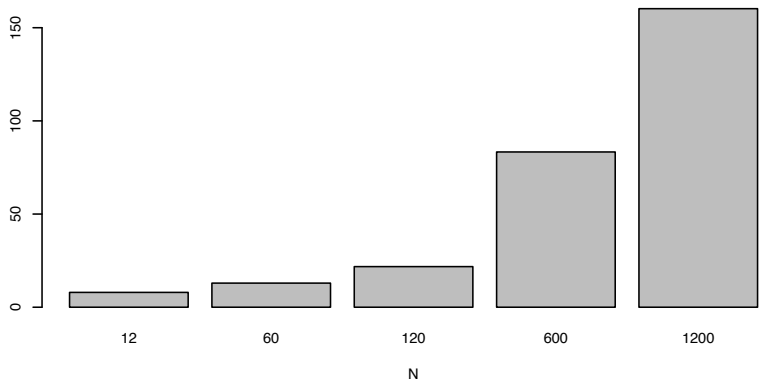

Figure B.2: Scenario 1: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.2. See also Table B.2.

B.1.2 Scenario 2

Table B.3: Scenario 2: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by G, across the datasets simulated from the settings described in Table 4.6.

G	Mean	SD
12	4.95703	0.20318
60	5.00000	0.00000
120	5.00000	0.00000
600	5.28906	0.98739
1500	5.01562	0.19741

Table B.4: Scenario 2: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by G, across the datasets simulated from the settings described in Table 4.6.

G	Mean	SD
12	0.07286	0.01496
60	0.33071	0.03244
120	0.62677	0.03691
600	3.87268	0.55939
1500	9.12037	0.58807

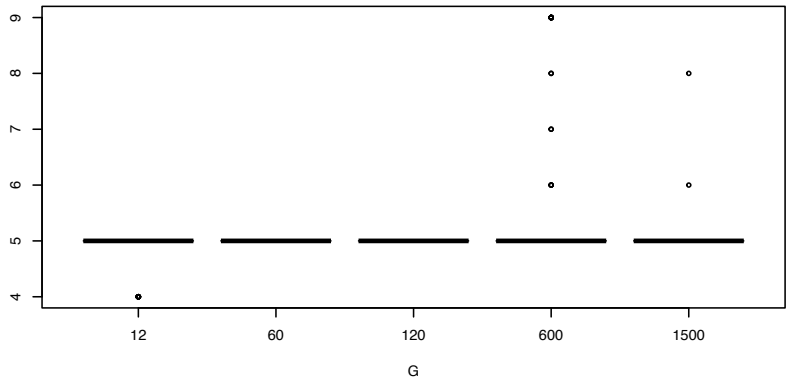

Figure B.3: Scenario 2: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.6. See also Table B.3.

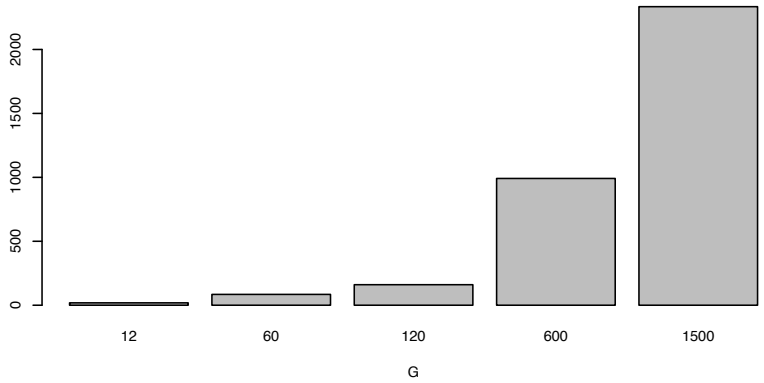

Figure B.4: Scenario 2: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.6. See also Table B.4.

B.1.3 Scenario 3

Table B.5: Scenario 3: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by K, across the datasets simulated from the settings described in Table 4.10.

K	Mean	SD
1	3	0
2	3	0
3	5	0
5	5	0

Table B.6: Scenario 3: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by K, across the datasets simulated from the settings described in Table 4.10.

K	Mean	SD
1	0.15568	0.01381
2	0.28696	0.02467
3	0.62479	0.03260
5	1.07090	0.06759

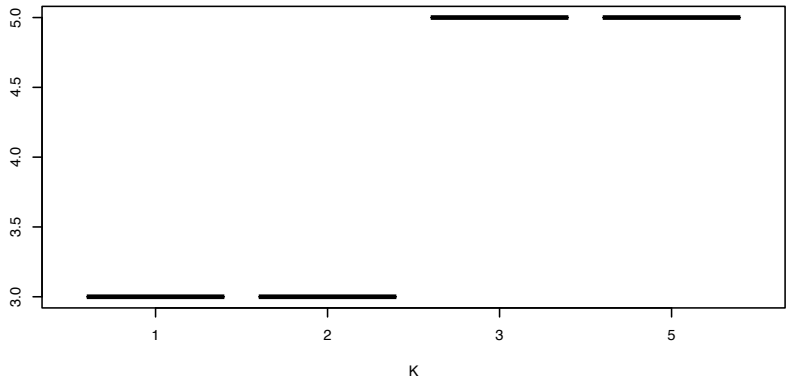

Figure B.5: Scenario 3: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.10. See also Table B.5.

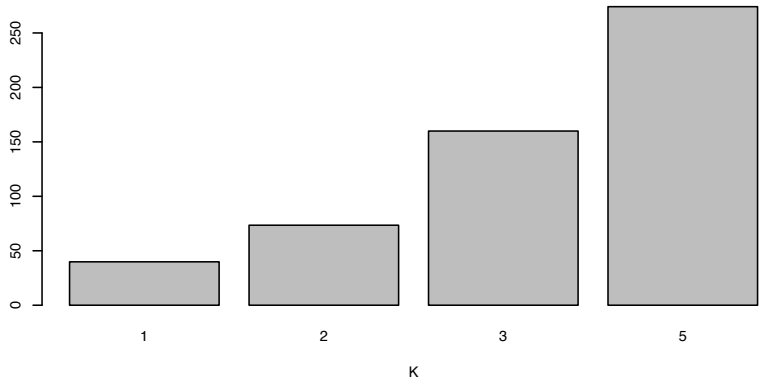

Figure B.6: Scenario 3: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.10. See also Table B.6.

B.1.4 Scenario 4

Table B.7: Scenario 4: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by case, across the datasets simulated from the settings described in Table 4.14.

Case	Mean	SD
1	3	0
2	3	0
3	3	0
4	3	0

Table B.8: Scenario 4: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by case, across the datasets simulated from the settings described in Table 4.14.

Case	Mean	SD
1	0.28701	0.02457
2	0.28668	0.02583
3	0.28782	0.02587
4	0.28946	0.02601

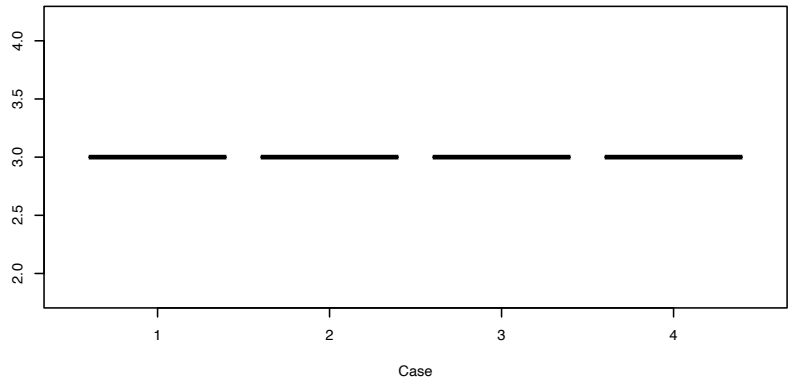

Figure B.7: Scenario 4: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.14. See also Table B.7.

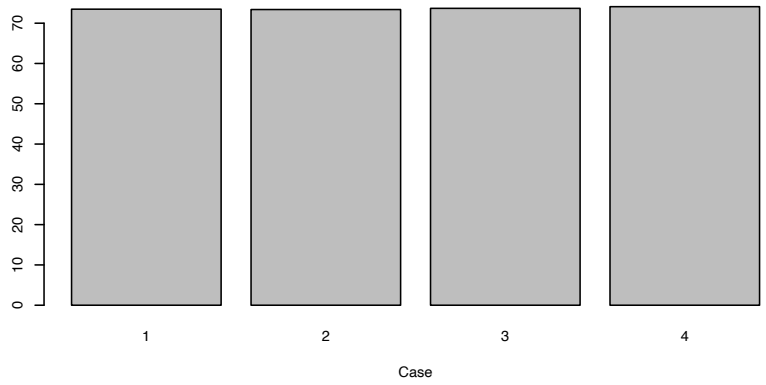

Figure B.8: Scenario 4: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.14. See also Table B.8.

B.1.5 Scenario 5

Table B.9: Scenario 5: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by case, across the datasets simulated from the settings described in Table 4.18.

Case	Mean	SD
1	0.66502	0.06492
2	0.75189	0.07942
3	0.90994	0.11186
4	1.30957	0.18450
5	2.21893	0.38235

Table B.10: Scenario 5: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by case, across the datasets simulated from the settings described in Table 4.18.

Case	Mean	SD
1	8.55078	0.71219
2	9.72266	1.00060
3	11.87500	1.42526
4	16.98828	2.30172
5	30.19922	5.38074

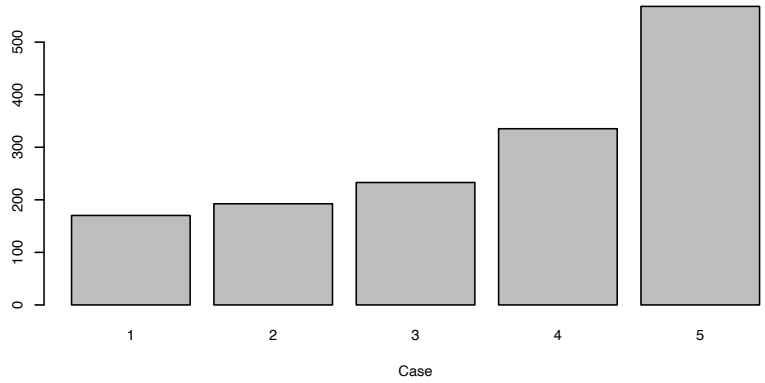

Figure B.9: Scenario 5: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.18. See also Table B.9.

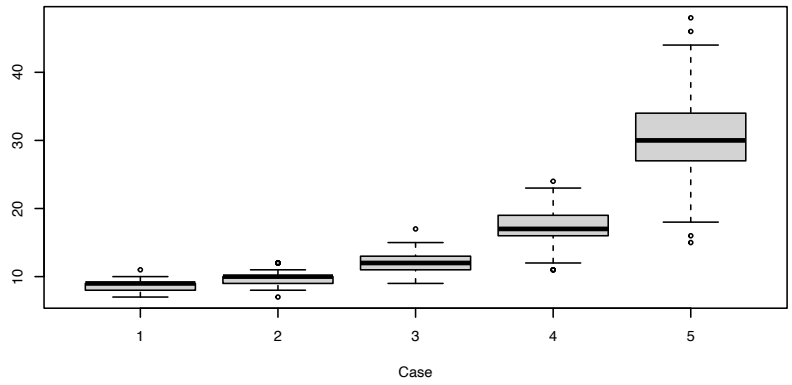

Figure B.10: Scenario 5: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.18. See also Table B.10.

B.1.6 Scenario 6

Table B.11: Scenario 6: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by case, across the datasets simulated from the settings described in Table 4.22.

Case	Mean	SD
1	5.01172	0.10783
2	5.00000	0.00000
3	5.00000	0.00000
4	5.00000	0.00000
5	6.86719	1.24850
6	5.00000	0.00000

Table B.12: Scenario 6: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by case, across the datasets simulated from the settings described in Table 4.22.

Case	Mean	SD
1	0.63283	0.04231
2	0.62440	0.03581
3	0.60219	0.03868
4	0.56450	0.04575
5	0.55288	0.10138
6	0.60067	0.03832

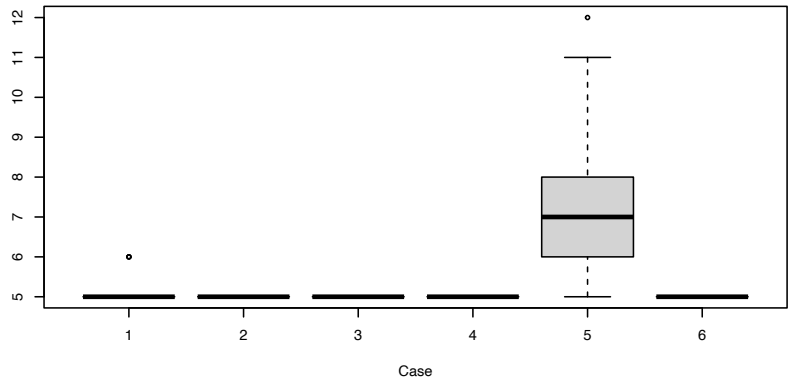

Figure B.11: Scenario 6: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.22. See also Table B.11.

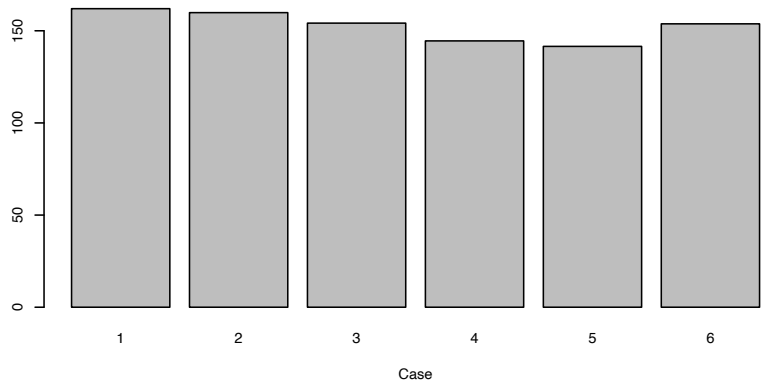

Figure B.12: Scenario 6: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.22. See also Table B.12.

B.2 Simulation Scenarios for the Mixture of ZIP with β_{0g} and ρ_{gk}

B.2.1 Scenario 1

Table B.13: Scenario 1: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by N, across the datasets simulated from the settings described in Table 4.26.

N	Mean	SD
12	0.12676	0.11630
60	0.08136	0.01778
120	0.15825	0.02253
600	0.60486	0.05189
1200	1.26211	0.11048

Table B.14: Scenario 1: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by N, across the datasets simulated from the settings described in Table 4.26.

N	Mean	SD
12	11.19531	3.35964
60	8.64844	0.78305
120	8.27734	0.63031
600	7.88281	0.39845
1200	7.96094	0.21337

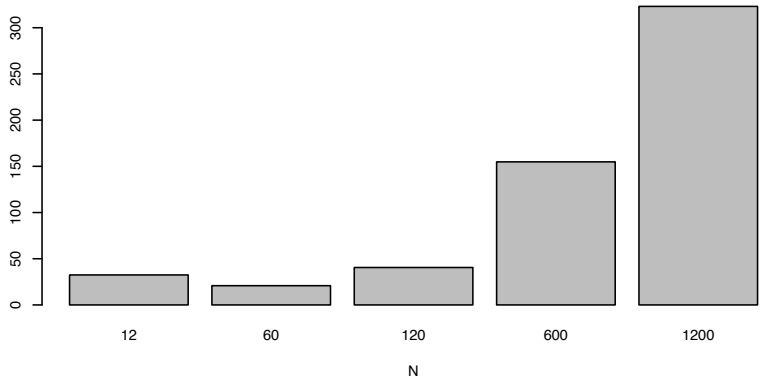

Figure B.13: Scenario 1: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.26. See also Table B.13.

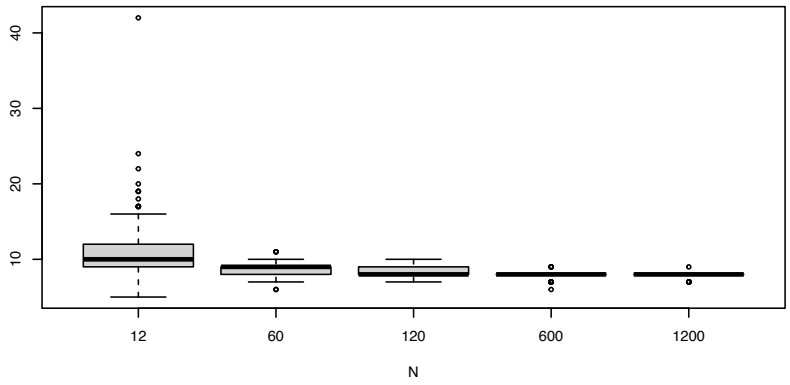

Figure B.14: Scenario 1: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.26. See also Table B.14.

B.2.2 Scenario 2

Table B.15: Scenario 2: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by G, across the datasets simulated from the settings described in Table 4.31.

G	Mean	SD
12	7.63672	0.53583
60	7.89844	0.30266
120	7.94141	0.23532
600	8.69141	0.47946
1200	8.76953	0.44015
6000	8.52734	0.50023

Table B.16: Scenario 2: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by G, across the datasets simulated from the settings described in Table 4.31.

G	Mean	SD
12	0.14569	0.01856
60	0.60723	0.04875
120	1.25957	0.11428
600	7.35106	0.71482
1200	15.25344	1.32889
6000	80.20718	5.61694

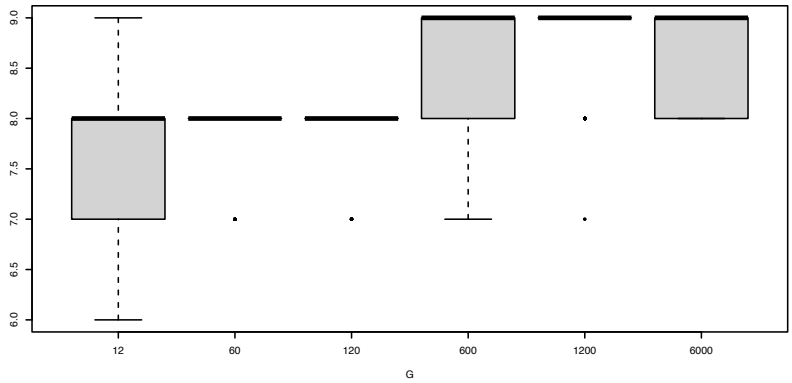

Figure B.15: Scenario 2: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.31. See also Table B.15.

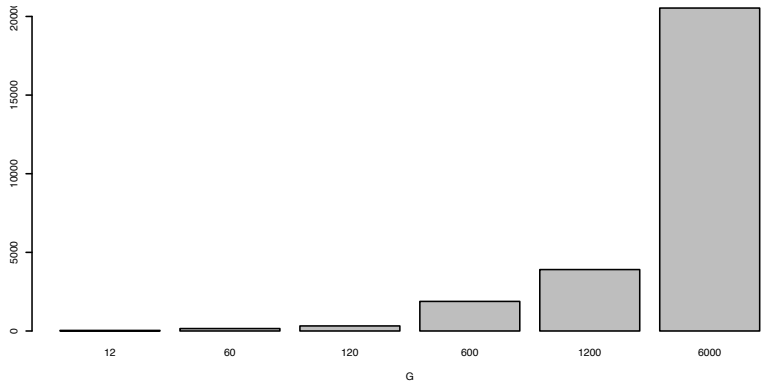

Figure B.16: Scenario 2: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.31. See also Table B.16.

B.2.3 Scenario 3

Table B.17: Scenario 3: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by K, across the datasets simulated from the settings described in Table 4.36.

K	Mean	SD
1	4.95312	0.21179
2	8.87500	0.35425
3	7.99609	0.30030
5	9.79297	0.42485

Table B.18: Scenario 3: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by K, across the datasets simulated from the settings described in Table 4.36.

K	Mean	SD
1	0.25657	0.02467
2	0.90463	0.06559
3	1.27514	0.09175
5	2.68266	0.19326

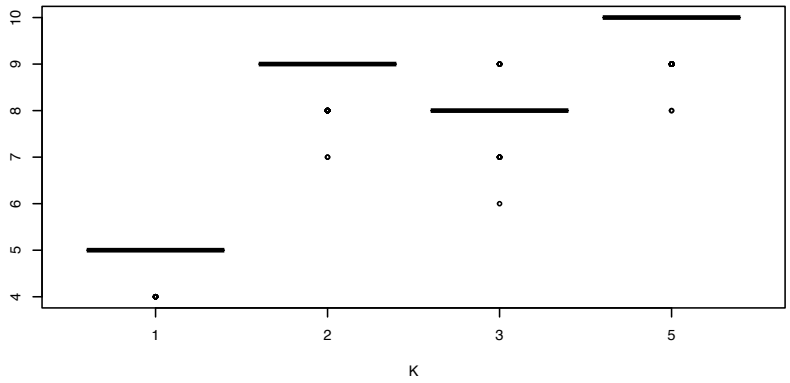

Figure B.17: Scenario 3: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.36. See also Table B.17.

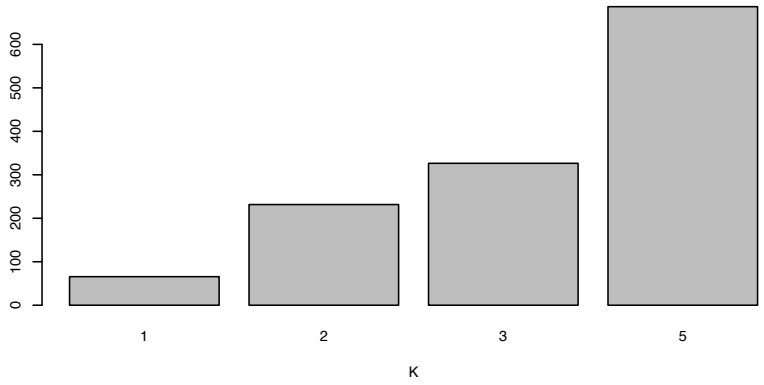

Figure B.18: Scenario 3: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.36. See also Table B.18.

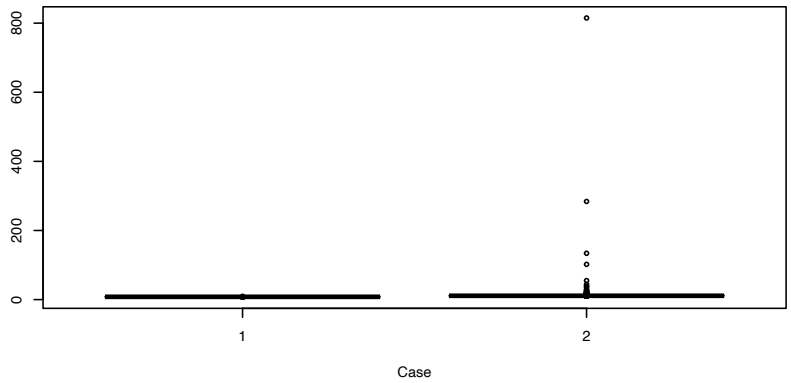

Figure B.19: Scenario 4: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.41. See also Table B.19.

B.2.4 Scenario 4

Table B.19: Scenario 4: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by case, across the datasets simulated from the settings described in Table 4.41.

Case	Mean	SD
1	7.96875	0.21466
2	17.14844	53.90989

Table B.20: Scenario 4: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by case, across the datasets simulated from the settings described in Table 4.41.

Case	Mean	SD
1	1.31225	0.07914
2	3.86476	8.52394

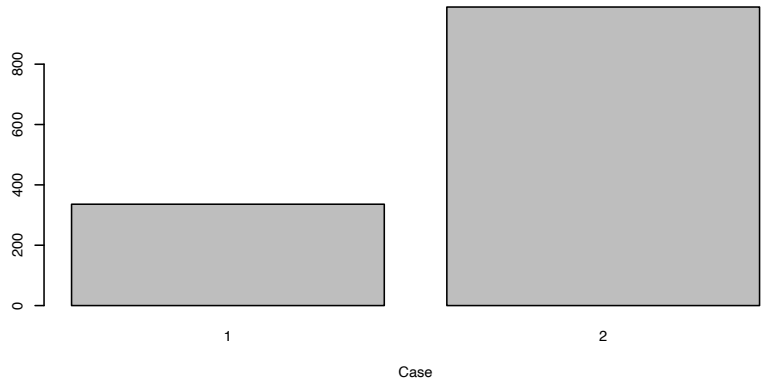

Figure B.20: Scenario 4: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.41. See also Table B.20.

B.2.5 Scenario 5

Table B.21: Scenario 5: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by case, across the datasets simulated from the settings described in Table 4.46.

Case	Mean	SD
1	8.89062	0.32502
2	8.89844	0.31535
3	8.86328	0.37685
4	8.95703	0.38870

Table B.22: Scenario 5: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by case, across the datasets simulated from the settings described in Table 4.46.

Case	Mean	SD
1	0.90346	0.06764
2	0.89992	0.06433
3	0.89883	0.06407
4	0.92589	0.06959

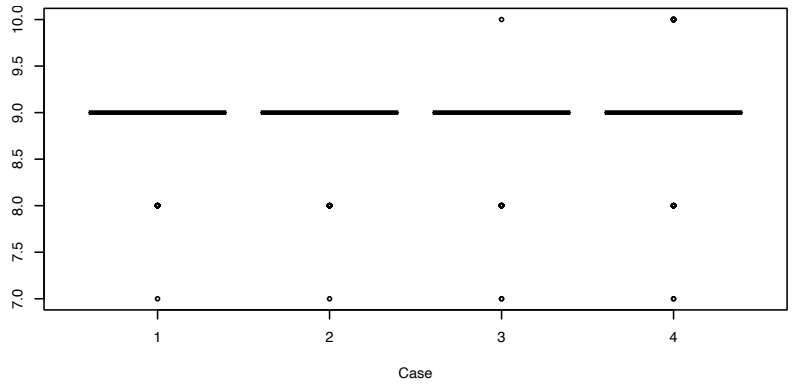

Figure B.21: Scenario 5: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.46. See also Table B.21.

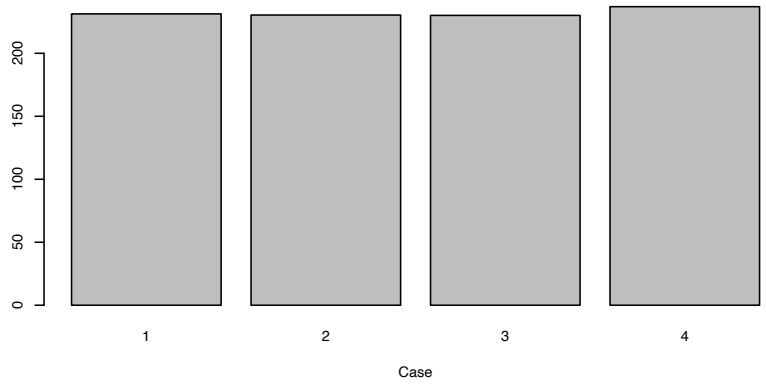

Figure B.22: Scenario 5: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.46. See also Table B.22.

B.2.6 Scenario 6

Table B.23: Scenario 6: Mean and standard deviation (SD) for the number of iterations until the EM algorithm converged, by case, across the datasets simulated from the settings described in Table 4.51.

Case	Mean	SD
1	8.51172	0.55999
2	7.96875	0.27830
3	7.60547	0.50547
4	7.01562	0.27962
5	9.84375	2.05393
6	8.10547	0.55361

Table B.24: Scenario 6: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by case, across the datasets simulated from the settings described in Table 4.51.

Case	Mean	SD
1	1.32499	0.11500
2	1.26188	0.09447
3	1.17087	0.10087
4	1.02929	0.08939
5	1.27236	0.30302
6	1.24007	0.12507

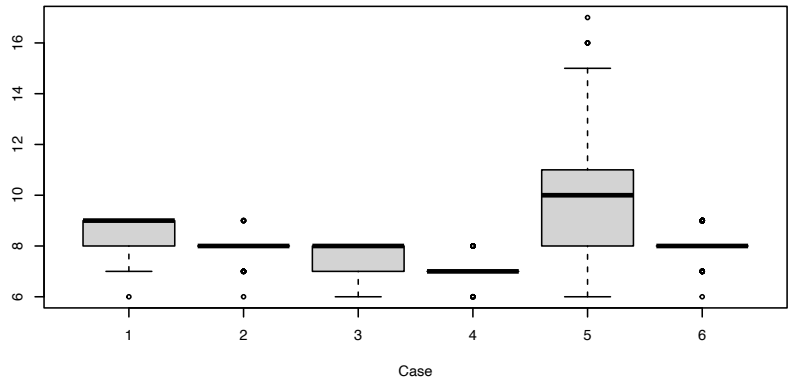

Figure B.23: Scenario 6: Boxplots for the number of iterations until the EM algorithm converged across the datasets simulated from the settings described in Table 4.51. See also Table B.23.

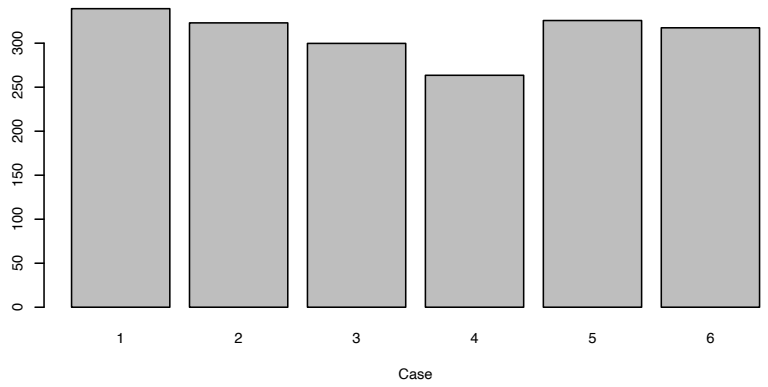

Figure B.24: Scenario 6: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.51. See also Table B.24.

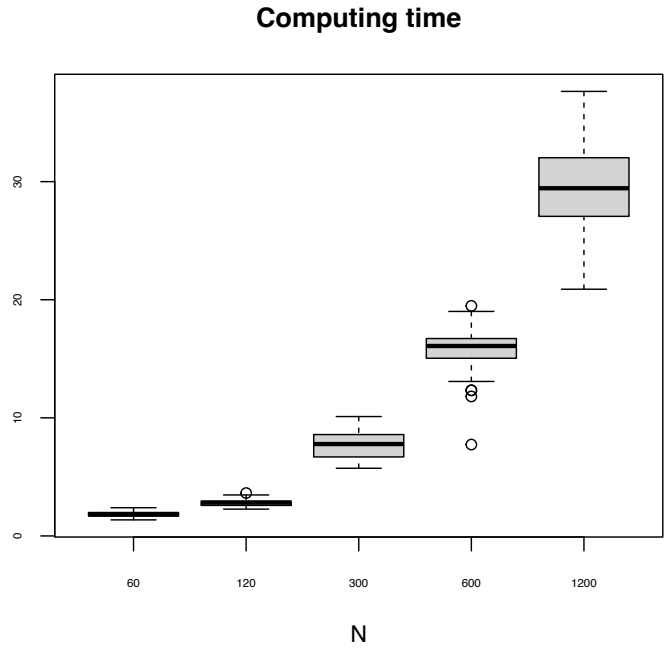

Figure B.25: Scenario 1: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.56. See also Table B.25.

B.3 Simulation Scenarios for the Mixture of ZIP with β_{0g}, ρ_{gk}, and β_{pg}

B.3.1 Scenario 1

Table B.25: Scenario 1: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by K, across the datasets simulated from the settings described in Table 4.56.

N	Mean	SD
60	1.84	0.21
120	2.80	0.27
300	7.72	1.15
600	15.80	1.64
1200	29.46	3.03

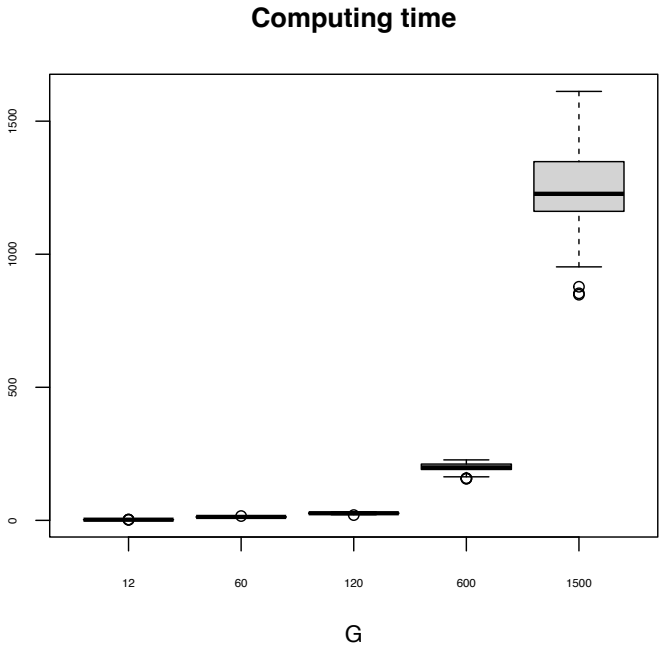

Figure B.26: Scenario 2: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.61. See also Table B.26.

B.3.2 Scenario 2

Table B.26: Scenario 2: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by K, across the datasets simulated from the settings described in Table 4.61.

G	Mean	SD
12	2.43	0.23
60	12.78	1.14
120	26.62	2.33
600	198.17	16.87
1500	1241.09	150.62

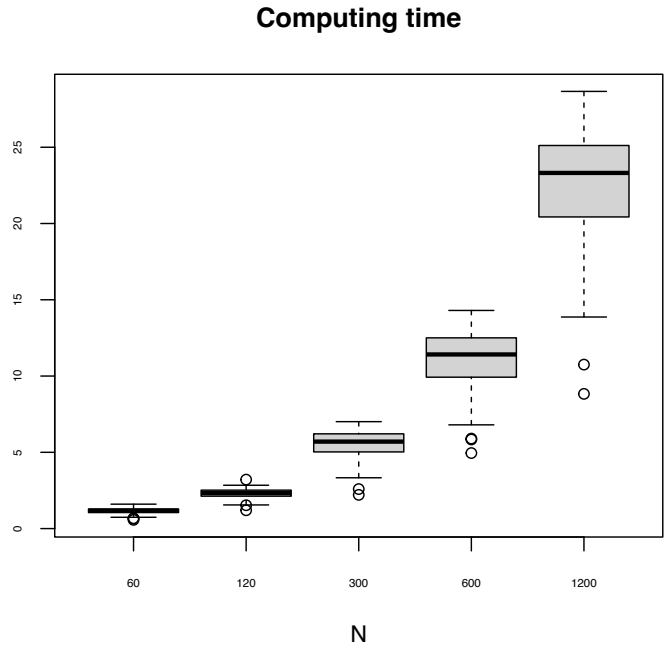

Figure B.27: Scenario 1: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.66. See also Table B.27.

B.4 Simulation Scenarios for the Mixture of ZINB without covariates

B.4.1 Scenario 1

Table B.27: Scenario 1: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by K, across the datasets simulated from the settings described in Table 4.66.

N	Mean	SD
60	1.15	0.19
120	2.29	0.31
300	5.52	0.95
600	11.05	1.89
1200	22.61	3.57

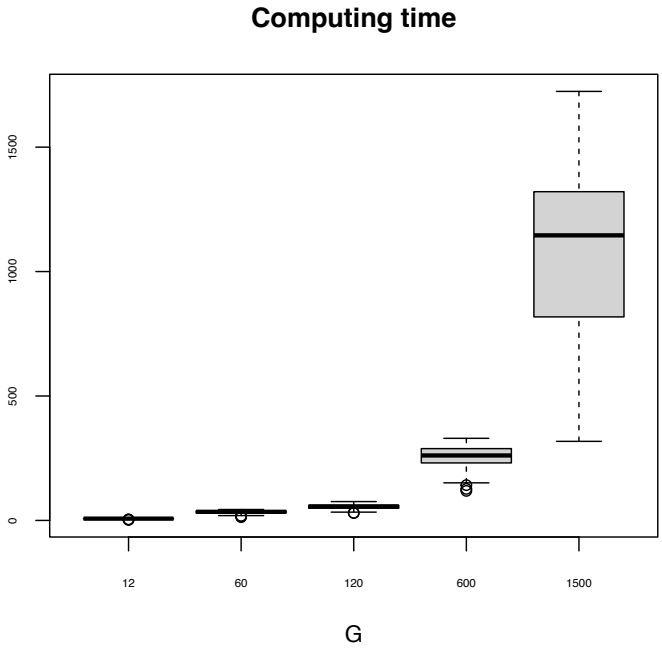

Figure B.28: Scenario 2: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.71. See also Table B.28.

B.4.2 Scenario 2

Table B.28: Scenario 2: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by K, across the datasets simulated from the settings described in Table 4.71.

G	Mean	SD
12	6.96	1.26
60	33.52	6.32
120	54.62	9.28
600	256.19	46.64
1500	1099.00	302.13

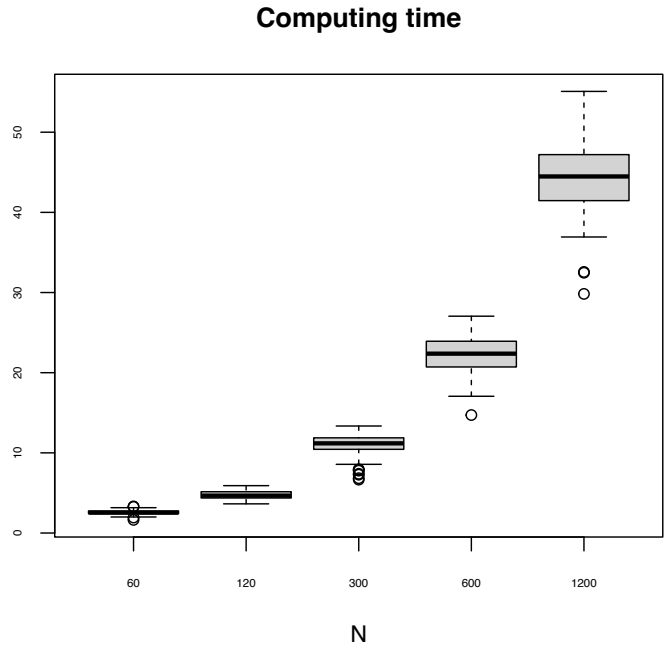

Figure B.29: Scenario 1: Bar plot for the total computing times, in seconds, taken for the EM algorithm to converge across the datasets simulated from the settings described in Table 4.76. See also Table B.29.

B.5 Simulation Scenarios for the Mixture of ZINB with covariates

B.5.1 Scenario 1

Table B.29: Scenario 1: Mean and standard deviation (SD) for the EM algorithm computing times, in seconds, by K, across the datasets simulated from the settings described in Table 4.76.

N	Mean	SD
60	2.58	0.29
120	4.72	0.52
300	10.95	1.37
600	22.14	2.38
1200	44.42	4.34

Curriculum Vitae

Name:	Zahra A.Shirazi
Post-Secondary Education and Degrees:	Western University, London, ON, Canada Ph.D. in Statistics 2017-2023
	York University, Toronto, ON, Canada Master in Statistics 2016-2017
	Shahid Beheshti University, Tehran, Iran M.Sc in Applied Statistics 2008-2011
	Shahid Beheshti University, Tehran, Iran B.Sc in Statistics 2002-2006
Honours and Awards:	Western Graduate Scholarships (WGRS), Western University 2017-2021
	York Graduate Fellowship, York University 2016-2017
Related Work Experience:	Research and Teaching Assistant Western University (2017-2023)
	Statistical Consultant Western Data Science Solution (2018-2021)
	Teaching Assistant York University (2016-2017)
	Teaching Assistant Shahid Beheshti University (2006-2008)

Publications:

1. Aghahosseinalishirazi, Z., da Silva, J. P., de Souza, C. P. E., Parameter Estimation for Grouped Data Using EM and MCEM Algorithms.(Aug 2022) Communication in Statistics-Simulation and Computation. https://doi.org/10.1080/03610918.2022.2108843.

2. Shirazi, Z., de Souza, C. P. E., Kashef, R. and Rodrigues, F. F., (2020) Deep Learning in the Healthcare Industry: Theory and Applications, Book Chapter. In Computational Intelligence and Soft Computing Applications in Healthcare Management Science (pp. 220-245). IGI Global.

Conference Presentations:

1. Aghahosseinalishirazi, Z., Assuncao Rangel,P., de Souza, C. P. E., Clustering Single-Cell RNA Sequencing Data Via The Expectation-Maximization Algorithm. (2023) Poster presentation in the Joint Statistical Meetings (JSM).

2. Aghahosseinalishirazi, Z., Assuncao Rangel,P., de Souza, C. P. E., Clustering Single-Cell RNA Sequencing Data Via The Expectation-Maximization Algorithm. (2022) Presented as a talk in Annual meeting of the Statistical Society of Canada (SSC).

3. Aghahosseinalishirazi, Z., da Silva, J. P., de Souza, C. P. E., Parameter Estimation for Grouped Data Using EM and MCEM Algorithms. (2021) Presented as a talk in Annual meeting of the Statistical Society of Canada (SSC).