Fall 2020

Statistical Modeling of High-throughput Sequencing Data and Spatially Resolved Transcriptomic Data

Shen Yin
Southern Methodist University, syin@smu.edu

STATISTICAL MODELING OF HIGH-THROUGHPUT SEQUENCING DATA AND SPATIALLY RESOLVED TRANSCRIPTOMIC DATA

Approved by:

Dr. Xinlei Wang
Professor in Department of Statistical Science, SMU

Dr. Guanghua Xiao
Professor in Department of Population and Data Sciences, UTSW

Dr. Lynne Stokes
Professor in Department of Statistical Science, SMU

Dr. Daniel F. Heitjan
Professor in Department of Statistical Science, SMU & Population and Data Sciences, UTSW

STATISTICAL MODELING OF HIGH-THROUGHPUT SEQUENCING DATA AND SPATIALLY RESOLVED TRANSCRIPTOMIC DATA

A Dissertation Presented to the Graduate Faculty of the

Dedman College

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Biostatistics

by

Shen Yin

B.A., Mathematics and Applied Mathematics, Beijing Institute of Technology
M.S., Applied Statistics and Data Analytics, Southern Methodist University

December 19, 2020

ACKNOWLEDGMENTS

I would like to express my special appreciation and thanks to my advisors Dr. Xinlei Wang and Dr. Guanghua Xiao. Their invaluable support during my Ph.D study is more than I could ever give them credit for here. They have shown me what a biostatistician should be. I wish to thank my dissertation committee members, Dr. Lynne Stokes and Dr. Daniel F. Heitjan. Without their insightful guidance and excellent teaching during my Ph.D study, I would not have completed this thesis. In addition, I would like to acknowledge my colleagues at the Quantitative Biomedical Research Center at UT Southwestern for their patient support and for all of the research opportunities I was given. Last but not least, I would like to thank my family and all of my friends, whose support and guidance are always with me.

Yin, Shen B.A., Mathematics and Applied Mathematics, Beijing Institute of Technology
M.S., Applied Statistics and Data Analytics, Southern Methodist University

Statistical Modeling of High-throughput Sequencing Data
and Spatially Resolved Transcriptomic Data

Advisors: Dr. Xinlei Wang, Dr. Guanghua Xiao

Doctor of Philosophy degree conferred December 19, 2020
Dissertation completed December 8, 2020

Recent studies have shown that RNA sequencing (RNA-seq) can be used to measure mRNA of sufficient quality extracted from Formalin-Fixed Paraffin-Embedded (FFPE) tissues to provide whole-genome transcriptome analysis. However, little attention has been given to the normalization of FFPE RNA-seq data, a key step that adjusts for unwanted biological and technical effects that can bias the signal of interest. Existing methods, developed based on fresh frozen or similar-type samples, may cause suboptimal performance. In Chapter 1, we propose a new normalization method, labeled MIXnorm, for FFPE RNA-seq data. MIXnorm relies on a two-component mixture model, which models non-expressed genes by zero-inflated Poisson distributions and models expressed genes by truncated normal distributions. To obtain maximum likelihood estimates, we develop a nested EM algorithm, in which closed-form updates are available in each iteration. By eliminating the need for numerical optimization in the M-step, the algorithm is easy to implement and computationally efficient. We evaluate MIXnorm through simulations and cancer studies. MIXnorm makes a significant improvement over commonly used methods for RNA-seq expression data.

MIXnorm has been shown to outperform the normalization methods developed based on FF RNA-seq data, but at the cost of a complex mixture model and a high computational burden. It is therefore important to adapt MIXnorm for simplicity and computational efficiency while maintaining superior performance. Furthermore, it is critical to develop

an integrated tool that performs commonly used normalization methods for both FF and FFPE RNA-seq data. In Chapter 2, we develop a new normalization method for FFPE RNA-seq data, named SMIXnorm, based on a simplified two-component mixture model compared to MIXnorm to facilitate computation. The maximum likelihood estimates of the model parameters are obtained by a nested Expectation-Maximization algorithm with a less complicated latent variable structure, and closed-form updates are available within each iteration. Real data applications and simulation studies show that SMIXnorm greatly reduces computing time compared to MIXnorm, without sacrificing the performance. More importantly, we developed a web-based tool, *RNA-seq Normalization* (*RSeqNorm* available at http://lce.biohpc.swmed.edu/rseqnorm), that offers a simple workflow to compute normalized RNA-seq data for both FFPE and FF samples. It includes SMIXnorm and MIXnorm for FFPE RNA-seq data, together with five commonly used normalization methods for FF RNA-seq data. Users can easily upload a raw RNA-seq count matrix and select one of the seven normalization methods to produce a downloadable normalized expression matrix for any downstream analysis.

Recently, spatial molecular profiling technologies have enabled a comprehensive catalog of molecular profiling data together with tissue imaging data with spatial locations and organizations. In the context of spatial profiling, the research interest lies in investigating the association between gene expression levels and their spatial locations, i.e., identifying spatially expressed (SE) genes. However, gene expression data from spatial molecular profiling are subject to severe zero-inflation issues. In Chapter 3, we propose a Bayesian Spatial HEAPing model (SHEAP), which aims to accurately recover major spatial patterns underlying the gene expression levels that are partially observed and subject to heaping at zero. An efficient Markov chain Monte Carlo (MCMC) algorithm is developed for Bayesian inference. We evaluate the proposed method through simulation studies and two real data applications. SHEAP shows significant improvement in detecting SE genes compared to existing methods.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I dedicate this dissertation to my family.

CHAPTER 1

MIXnorm: Normalizing RNA-seq Data from Formalin-Fixed Paraffin-Embedded Samples

1.1. Introduction

Human tissue biospecimens are of two primary types, fresh-frozen (FF) and formalin-fixed paraffin-embedded (FFPE) tissues. As fresh tissues deteriorate rapidly at room temperature, FF samples must be frozen instantly after collection and then stored in freezers. FF tissues are well suited for molecular analysis using gene expression measurements as freezing preserves RNA well. However, they are expensive to store and transport, and difficult to collect for large-scale studies. By contrast, FFPE samples can be stored at room temperature and kept for a long time. Due to the ease of handling and inexpensive storage, numerous FFPE tissue samples have been deposited into tissue banks and pathology laboratories around the world, and are readily available [42, 47, 48]. The ubiquity of FFPE tissue specimens has made them an invaluable resource in biomedical research, with great potential for predictive and prognostic biomarker discovery.

However, the quality of RNA extracted from FFPE tissues is a concern due to chemical modifications and continued degradation over time. The process of using formalin to fix and paraffin embedding to preserve tissues for an extended period of time is designed to well preserve cellular proteins rather than preserving RNA. Consequently, assays using microarray or quantitative polymerase chain reaction (qPCR) often have limited reproducibility and sensitivity when measuring gene expression from such samples. In order to exploit the vast collection of FFPE samples, substantial effort has been devoted to de-

velopment and/or validation of advanced technologies that can reliably probe their gene expression levels. For high-throughput profiling, RNA sequencing (RNA-seq), which uses next-generation sequencing (NGS) to reveal the presence and quantity of RNA in a biological sample, is in common use. Recent studies have shown that for a wide variety of human tumor tissues (e.g., bladder, colon, prostate and renal carcinoma), RNA-seq can be used to measure mRNA of sufficient quality extracted from FFPE tissues to provide biologically relevant transcriptome analysis [19, 20]. Meanwhile, recent FFPE RNA-Seq solutions such as Illumina total RNA-Seq enable researchers to produce high-quality results from degraded samples. As a result, a drastically increasing number of studies have used RNA-seq on FFPE specimens [33, 39].

A critical step when analyzing RNA-seq data is normalization. Normalization removes systematic biases that affect measured gene expression levels (e.g., variability in experimental conditions, sample collection and preparation, and machine parameters, etc.), while preserving the variation in gene expression that occurs because of biologically relevant changes in transcription. A number of normalization methods for RNA-seq data have been developed [11]. A common approach is to normalize the measured expression using (estimated) scaling factors. The most straightforward normalization method, Reads Per Million (RPM) [38], estimates the scaling factor by dividing the total read count of a sample by $1,000,000$. The normalized data are the read counts divided by the scaling factors. The Upper-Quartile (UQ) [4] method estimates the scaling factor by the upper quartile of the read counts within each sample. DESeq [1] works under the assumption that only a small subset of genes are differentially expressed (DE). First, for each gene, the ratio of its read count over its geometric mean across all samples is calculated. Then the scaling factor is estimated by the median ratio within each sample. Thus, it is also referred to as median normalization. Trimmed Mean of M-values (TMM) [49] is also based on the assumption that most of the genes are not DE, where one sample is chosen as the reference sample and the others as test samples. The log ratio of the read count between each test sample and the reference is computed for each gene. Then for each test

sample, TMM estimates the scaling factor by the weighted mean of log ratios after exclusion of the genes with extreme average expression or with largest log ratios. PoissonSeq (PS) [31] models RNA-seq data by a Poisson log-linear model. The normalization is done implicitly by including the scaling factor as a term in the model.

Though a number of normalization methods are available for RNA-seq data, none has been specifically designed for FFPE samples, of which a prominent feature is sparsity (i.e., excessive zero or small counts), caused by RNA degradation in such samples. The quantile-based methods become problematic due to excess zeros that cause ranking ties. For DEseq, the geometric mean is only well defined for genes with at least one read count in every sample. The zero inflation is also a concern for methods that implicitly use scaling factors such as PS since they all rely on Poisson or Negative Binomial distributions for modeling count data.

To illustrate characteristics of RNA-seq data from FFPE samples, we begin by presenting an exploratory analysis in Section 1.2 using a real data example. In Section 1.3, we propose a novel normalization method, called MIXnorm, based on a two-component **mix**ture model for log read counts, to capture the sparsity as well as major mean and variance structures underlying the data. Due to whole-genome sequencing, the number of parameters involved is often very large. We develop an efficient nested expectation–maximization (EM) algorithm to fit the proposed mixture model, where parameters are updated via closed-form solutions iteratively. Section 1.4 briefly summarizes simulation studies and expounds two real data applications to compare the performance of the proposed MIXnorm to five commonly used RNA-seq normalization methods, including UQ, DESeq, RPM, PS and TMM. Section 1.5 concludes the paper with a brief discussion. Technical details, performance evaluation via simulation, and additional analysis results are available in Appendix A.

1.2. An Exploratory Analysis

As mentioned in the introduction, a striking feature of FFPE RNA-seq data is the sparsity, which can be observed in multiple data sets from independent studies. An example is provided here using paired FF and FFPE samples from a published study, RNA sequencing validation of the Complexity INdex in SARComas prognostic signature [29]. Prognosis of metastatic outcomes in soft tissue sarcomas is important because of its high recurring rate (up to 50% of recurrence). Complexity INdex in SARComas (CINSARC), a gene signature that consists of 67 genes, has been identified as a valuable prognostic factor in sarcomas. This signature was originally identified on FF samples assayed by the microarray platform. The study goal of Lesluyes et al. [29] was to evaluate the prognostic performance of CINSARC on both FF and FFPE samples. Thus, the resulting data set contains gene expression levels for $20,242$ protein coding genes, measured by whole-genome next generation sequencing on paired FF and FFPE samples from 41 patients.

We first transformed the raw read counts in this dataset into the natural logarithm scale. In order to deal with zero counts, we define the log count $L \equiv \log(C+1)$, where C is the raw count. Figure 1.1(a) shows that among a total of 20,242 genes, there is a significant portion of genes with more than 50% zero counts in FFPE samples while (b) shows that over 65% genes, represented by the leftmost bar, do not have any zero count in FF samples. Further, Figure 1.1(c) and (d) show that for each sample, regardless of sample types, the commonly used Poisson or Negative Binomial distributions for count data are far from being adequate to capture the bimodal density of gene expression (with one spike at zero). Two other interesting observations from Figure 1.1(c) and (d) are: (1) the locations of the distributions of 41 FFPE samples vary much more than those of FF samples, indicating great heterogeneity in RNA degradation levels among the FFPE tissues; and (2) densities from different FF samples show highly similar variability while those from FFPE samples do not (the spread of the curves varies tremendously).

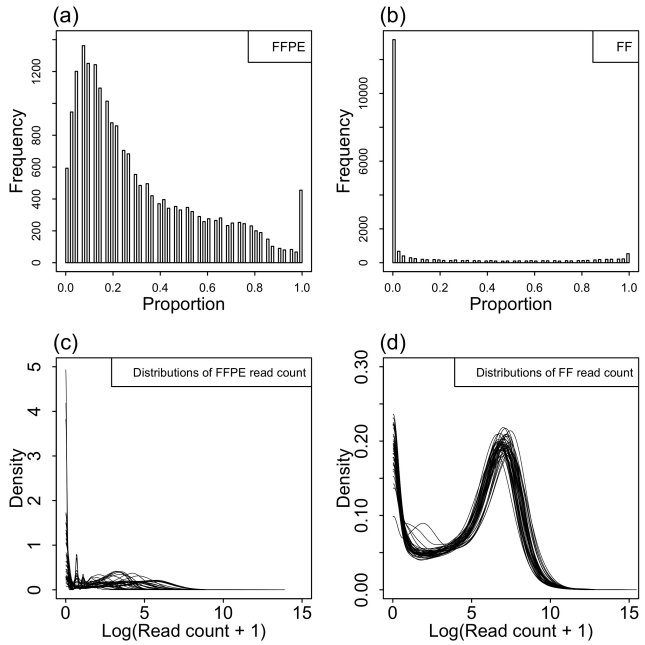


Figure 1.1: An exploratory analysis of RNA-seq data in Lesluyes et al. [29]. Panel (a)/(b) shows the histogram of zero-count proportion among 41 FFPE/FF samples (represented by the horizontal axis) based on a total of 20,242 genes. Panel (c)/(d) shows empirical densities of log read counts for the 41 FFPE/FF samples. Each curve in (c)/(d) represents the density for one sample across all the $20,242$ genes.

The above findings indicate that existing normalization methods for RNA-seq data, all developed based on FF or like samples, are ill-suited for FFPE samples as they cannot cope with the highly complex features of such data. We proceed to develop a robust yet powerful method, MIXnorm, based on a two-component mixture model to capture the distinct bimodality as well as major mean and variance structures underlying the data. The first component is to model non-expressed genes, whose read counts should be zero or relatively small due to non-specific binding. These genes include biologically zero-expression genes that may exist, or those with low expression but cannot be expressed due to various experimental limitations (e.g., drop-outs), or those that should be expressed but cannot because of high-level mRNA degradation. For the non-expressed genes, we use a zero inflated Poisson (ZIP) distribution to capture the spike at zero for each sample, of which the Poisson mean reflects the background noise level. The second component is to model expressed genes, and we use a truncated normal (TN) distribution

for log gene read counts of each sample to approximate the roughly bell-shaped curve centered at the second mode.

1.3. The MIXnorm Method

1.3.1. The statistical model for FFPE data

Let C_{ij} denote the raw count of gene j from sample i and $L_{ij} \equiv \log(C_{ij}+1)$ is the natural logarithm transformed count, for $i = 1, \ldots, I$, $j = 1, \ldots, J$. We define a latent binary variable D_j: $D_j = 0$ indicates gene j is non-expressed in this study, meaning that observed non-zero counts of gene j are due to background noise; $D_j = 1$ indicates gene j is expressed, with mean expression greater than 0. The following mixture model is proposed for FFPE data:

$$C_{ij} \sim \text{ZIP}(\pi_j, \delta_i), \text{ if } D_j = 0, \tag{1.1}$$

$$L_{ij} \sim \text{TN}(\mu_i, \sigma_i^2, 0, +\infty), \text{ if } D_j = 1, \tag{1.2}$$

$$D_j \sim \text{Bernoulli}(\phi),$$

where $0 \leq \pi_j, \phi \leq 1$, $\delta_i, \sigma_i > 0$ for $i = 1, ..., I$, $j = 1, ...J$. Here, ZIP(π_j, δ_i) stands for a zero inflated Poisson distribution, with probability π_j being zero and probability $1 - \pi_j$ being from a Poisson distribution with mean δ_i; TN$(\mu_i, \sigma_i^2; 0, +\infty)$ stands for a normal distribution with mean μ_i and variance σ_i^2, left truncated at zero as $L_{ij} > 0$; and ϕ is the proportion of expressed genes in the study. Figure 1.1(a) clearly shows the zero-count proportion varies across different genes, and so π_j is assumed to be gene-specific instead of being constant. The δ_i reflects sample-specific background noise and should be relatively small. Figure 1.1(c) shows that the location and spread of L_{ij} both vary a lot from sample to sample, meaning that the sample-specific mean μ_i and variance σ_i^2 are necessary for FFPE data. We note that L_{ij} is a discrete random variable with

support $\{0, \log(1), \log(2), ...\}$, but in (1.2), a continuous distribution is used to approximate the discrete distribution of L_{ij}.

Let $\Theta = (\boldsymbol{\pi}, \boldsymbol{\delta}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \phi)$ denote the collection of all the parameters in the mixture model, where $\boldsymbol{\pi} = (\pi_1, ..., \pi_J)$, $\boldsymbol{\delta} = (\delta_1, ..., \delta_I)$, $\boldsymbol{\mu} = (\mu_1, ..., \mu_I)$ and $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_I)$. The (incomplete) likelihood function is

$$\begin{aligned} L(\Theta|\boldsymbol{C}) &= \prod_{j=1}^{J} p(\boldsymbol{C}_j|\Theta) \\ &= \prod_{j=1}^{J} [p(\boldsymbol{C}_j|D_j = 1, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(D_j = 1|\phi) \\ &\quad + p(\boldsymbol{C}_j|D_j = 0, \pi_j, \boldsymbol{\delta}) p(D_j = 0|\phi)] \\ &= \prod_{j=1}^{J} [\prod_{i=1}^{I} p(C_{ij}|D_j = 1, \mu_i, \sigma_i) \cdot \phi \\ &\quad + \prod_{i=1}^{I} p(C_{ij}|D_j = 0, \pi_j, \delta_i) \cdot (1 - \phi)], \end{aligned}$$

where $p(C_{ij}|D_j = 0, \pi_j, \delta_i)$ is the probability mass function (PMF) of C_{ij} of non-expressed genes, i.e., the zero-inflated Poisson distribution in (1.1); $p(C_{ij}|D_j = 1, \mu_i, \sigma_i)$ is the PMF of C_{ij} for expressed genes, which will be approximated by a probability density function (PDF) with $\log(C_{ij} + 1)$ following the TN distribution on $[0, +\infty)$ in (1.2). See Appendix A.1 for a detailed justification about the validity of using the PDF to approximate the PMF.

1.3.2. Model fitting via an EM algorithm

A common method for estimating parameters of a model with a latent variable structure is to employ an EM algorithm [9] to obtain their maximum likelihood estimates (MLEs). The complete-data log-likelihood with the latent variables $\boldsymbol{D}$ is given by

$$\begin{aligned}\ell(\Theta|\boldsymbol{C},\boldsymbol{D}) &= \sum_{j=1}^{J}\log p(\boldsymbol{C}_j, D_j|\Theta)\\ &= \sum_{j=1}^{J} D_j\cdot\{\log(\phi)+\log\left[p(\boldsymbol{C}_j|D_j=1,\boldsymbol{\mu},\boldsymbol{\sigma})\right]\}\\ &+ \sum_{j=1}^{J}(1-D_j)\cdot\{\log(1-\phi)+\log\left[p(\boldsymbol{C}_j|D_j=0,\pi_j,\boldsymbol{\delta})\right]\}.\end{aligned} \tag{1.3}$$

Let $\Theta^{(t)} = (\boldsymbol{\pi}^{(t)}, \boldsymbol{\delta}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)}, \phi^{(t)})$ be the parameter estimates at the tth iteration. The distribution of $\boldsymbol{D}$ given the observed data $\boldsymbol{C}$ and the current parameter estimates $\Theta^{(t)}$ is

$$p(\boldsymbol{D}|\boldsymbol{C},\Theta^{(t)}) = \prod_{j=1}^{J}\frac{p(\boldsymbol{C}_j, D_j|\Theta^{(t)})}{p(\boldsymbol{C}_j|\Theta^{(t)})} = \prod_{j=1}^{J}\left(w_j^{(t)}\right)^{D_j}\left(1-w_j^{(t)}\right)^{1-D_j},$$

where

$$w_j^{(t)} = \frac{\phi^{(t)}p(\boldsymbol{C}_j|D_j=1,\boldsymbol{\mu}^{(t)},\boldsymbol{\sigma}^{(t)})}{\phi^{(t)}p(\boldsymbol{C}_j|D_j=1,\boldsymbol{\mu}^{(t)},\boldsymbol{\sigma}^{(t)})+(1-\phi^{(t)})p(\boldsymbol{C}_j|D_j=0,\pi_j^{(t)},\boldsymbol{\delta}^{(t)})}. \tag{1.4}$$

Each iteration of an EM algorithm consists of two steps, the expectation (E) step and the maximization (M) step. The E step calculates the expected complete-data log likelihood given $\boldsymbol{C}$ and $\Theta^{(t)}$, where the expectation is taken over the latent variables $\boldsymbol{D}$. Since $l(\Theta|\boldsymbol{C},\boldsymbol{D})$ in (1.3) is linear in D_j, and $\mathbf{E}(D_j|\boldsymbol{C},\Theta^{(t)}) = w_j^{(t)}$, we have

$$\begin{aligned}Q(\Theta|\Theta^{(t)}) &= \mathbf{E}_{\boldsymbol{D}|\boldsymbol{C},\Theta^{(t)}}\, l(\Theta|\boldsymbol{C},\boldsymbol{D})\\ &= \sum_{j=1}^{J}(1-w_j^{(t)})\left[\log(1-\phi)+\log p(\boldsymbol{C}_j|D_j=0,\pi_j,\boldsymbol{\delta})\right]\\ &+ \sum_{j=1}^{J} w_j^{(t)}\left[\log(\phi)+\log p(\boldsymbol{C}_j|D_j=1,\boldsymbol{\mu},\boldsymbol{\sigma})\right].\end{aligned} \tag{1.5}$$

In essence, the E step calculates the conditional expectation of $\boldsymbol{D}$ given $\boldsymbol{C}$ and $\Theta^{(t)}$. The M step updates the parameter estimates by maximizing the expected log likelihood

(1.5). Note that (1.5) can be maximized with respect to ϕ, $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ and $(\boldsymbol{\pi}, \boldsymbol{\delta})$ separately. The updated parameter estimates in the $(t+1)$th iteration are given by

$$\phi^{(t+1)} = \frac{\sum_{j=1}^{J} w_j^{(t)}}{J},$$

$$(\mu_i^{(t+1)}, \sigma_i^{(t+1)}) \quad = \quad \underset{\mu_i, \sigma_i}{arg\,max} \sum_{j=1}^{J} \log \mathrm{TN}(L_{ij}|\mu_i, \sigma_i, 0, \infty) \cdot w_j^{(t)}, \quad i = 1, ..., I, \quad (1.6)$$

$$\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}\right) = \underset{\boldsymbol{\pi}, \boldsymbol{\delta}}{arg\,max} \sum_{j=1}^{J} \sum_{i=1}^{I} \left[\log \mathrm{ZIP}(C_{ij}|\pi_j, \delta_i) \cdot \left(1 - w_j^{(t)}\right)\right], \quad (1.7)$$

where the maximization in (1.7) has constraints $\pi_j \in [0, 1]$ and $\delta_i > 0$; TN$(\cdot|\cdot)$ stands for the pdf of the TN distribution, ZIP$(\cdot|\cdot)$ stands for the pmf of the ZIP distribution, both with distributional parameters specified after "|". The update for $\phi^{(t+1)}$ has a closed form. Other parameters can be updated by a Newton-Raphson type method numerically within each iteration t.

The pmf $\mathrm{ZIP}(C_{ij}|\pi_j, \delta_i)$ in (1.7) cannot be factored into functions of π_j and δ_i. Therefore, the update of $(\boldsymbol{\pi}, \boldsymbol{\delta})$ involves multi-dimensional optimization, which can be computationally intensive when $I + J$ is large, as is typical for high-throughput profiling such as RNA-seq. Another drawback of the above algorithm is numerical instability due to the use of the Newton-Raphson method for an approximate solution in the M step. Dempster et al. [9] proved that for an EM-type algorithm, the (incomplete) likelihood in every iteration never decreases as t increases. Thus, the incomplete likelihood is typically used to monitor the convergence of the algorithm. However, this monotone convergence property does not necessarily hold if the E or M step is not computed exactly. In such situations, the incomplete log-likelihood may fluctuate around a fixed point for a long time. Due to this instability, when applying the above EM algorithm to real data, we observed that it would not converge, especially when a small tolerance value is selected to terminate the iterative process.

1.3.3. Review of nested EM algorithms

van Dyk [61] described how nesting two or more EM algorithms could take advantage of closed form conditional expectations and lead to algorithms with both ease of implementation and computing efficiency (i.e., fast and stable convergence). Assume the missing data can be split into two (or more) sets $\boldsymbol{Y}_{\text{mis }1}$ and $\boldsymbol{Y}_{\text{mis }2}$ such that the complete data can be expressed by $\boldsymbol{Y}_{\text{com}} = (\boldsymbol{Y}_{\text{obs}}, \boldsymbol{Y}_{\text{mis }1}, \boldsymbol{Y}_{\text{mis }2})$, where $\boldsymbol{Y}_{\text{mis }1}$ and $\boldsymbol{Y}_{\text{mis }2}$ can be introduced under a data augmentation scheme to aid the computation. Let Θ denote the vector of all parameters involved, and $\mathcal{H}$ is the parameter space. Define the nested conditional expectation of log-likelihood by

$$\tilde{Q}\left(\Theta|\Theta_1, \Theta_2\right) = \mathbf{E}\left\{\mathbf{E}\left[l\left(\Theta \mid \boldsymbol{Y}_{\text{obs}}, \boldsymbol{Y}_{\text{mis }1}, \boldsymbol{Y}_{\text{mis }2}\right) \mid \boldsymbol{Y}_{\text{obs}}, \boldsymbol{Y}_{\text{mis }1}, \Theta_1\right] \mid \boldsymbol{Y}_{\text{obs}}, \Theta_2\right\}, \tag{1.8}$$

where Θ_1 and Θ_2 denote different realizations of Θ, and $\tilde{Q}(\Theta|\Theta_1, \Theta_2)$ is a function on $\mathcal{H} \times \mathcal{H} \times \mathcal{H}$. The outer expectation in (1.8) is taken with respect to $\boldsymbol{Y}_{\text{mis }1}$ while the nested inner expectation is taken with respect to $\boldsymbol{Y}_{\text{mis }2}$. According to van Dyk [61], the tth iteration of a nested EM algorithm repeats the following cycle K times.

Cycle k for $k = 1, ..., K$:

E step: compute

$$\tilde{Q}\left(\Theta \mid \Theta^{(t+\frac{k-1}{K})}, \Theta^{(t)}\right) = \mathbf{E}\{\mathbf{E}[\ell\left(\Theta|\boldsymbol{Y}_{\text{com}}\right) \mid \boldsymbol{Y}_{\text{obs}}, \boldsymbol{Y}_{\text{mis }1}, \Theta^{(t+\frac{k-1}{K})}] \mid \boldsymbol{Y}_{\text{obs}}, \Theta^{(t)}\}$$

M step: update the parameter estimates by

$$\Theta^{(t+\frac{k}{K})} = \underset{\Theta}{arg\,max}\ \tilde{Q}\left(\Theta \mid \Theta^{(t+\frac{k-1}{K})}, \Theta^{(t)}\right).$$

Upon completion of the Kth cycle, set $\Theta^{(t+1)} = \Theta^{(t+\frac{K}{K})}$. That is, run K cycles of the inner EM algorithm for each iteration of the outer EM.

When the missing data structure is complex, direct calculation of $\mathbf{E}\left[\ell(\Theta|\boldsymbol{Y}_{\text{com}})|\boldsymbol{Y}_{\text{obs}}, \Theta^{(t)}\right]$ is usually difficult. Moreover, we may not be able to directly sample from $p(\boldsymbol{Y}_{\text{mis }1}, \boldsymbol{Y}_{\text{mis }2}|\boldsymbol{Y}_{\text{obs}}, \Theta)$, and thus a Monte-Carlo EM algorithm is not feasible as well. A nested EM algorithm takes advantages of subdividing the missing data so that $p(\boldsymbol{Y}_{\text{mis }1}|\boldsymbol{Y}_{\text{obs}}, \Theta)$ and $p(\boldsymbol{Y}_{\text{mis }2}|\boldsymbol{Y}_{\text{obs}}, \boldsymbol{Y}_{\text{mis }1}, \Theta)$ are both known distributions or easy to sample directly. Theoretical properties of nested EM algorithms have been well studied. Theorem 1 in [61] guarantees that, like EM algorithms, nested EM algorithms enjoy the monotone convergence property, and so the incomplete-data likelihood $p(\boldsymbol{Y}_{\text{obs}}|\Theta)$ can be used to detect convergence.

1.3.4. Model fitting via a nested EM algorithm

Below we introduce additional latent variables so that a nested EM-type algorithm can be constructed to improve computational efficiency. Based on Lambert [28], a zero inflated Poisson distribution can be thought of as a mixture of two states, the perfect zero state and the Poisson state. Suppose we knew which zeros came from the perfect zero state and which came from the Poisson state. That is, for a non-expressed gene j, we define $Z_{ij} = 1$ when C_{ij} is from the perfect zero state and $Z_{ij} = 0$ when C_{ij} is from the Poisson state, for $i = 1, \ldots I$. Obviously, $Z_{ij}|D_j = 0 \sim$ Bernoulli(π_j). Further, we augment the truncated normal data by (hypothesized) missing observations, which borrows ideas from Tanner and Wong [56] and McLachlan and Jones [37]. That is, the augmented data follow a normal distribution so that the posterior distributions of the parameters or their functions are straightforward to calculate. For sample i, apart from the observed J genes, there are T_i unobserved genes with $D_j = 1$ and their log count $L_{ij} < 0$, $j = J+1, ..., J+T_i$, such that $L_{ij} \sim \text{N}(\mu_i, \sigma_i)$, for $j = 1, ..., J + T_i$. Here, the number of observations T_i falling in $(-\infty, 0)$ is also latent. Note that we now have a quite complex latent variable structure. However, by nesting inner EM algorithms inside an outer EM, we do not need the actual

realizations of the unobservable random variables T_i and L_{ij} for $j = J+1, ..., J+T_i$. To iteratively update the parameter estimates, only the conditional expectations of the corresponding sufficient statistics are required.

A nested EM algorithm is invoked by treating $\boldsymbol{Y}_{\text{com}} = (\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{Z}, \boldsymbol{T}, \boldsymbol{L}_t)$ as the complete data, where $\boldsymbol{T} = (T_1, ..., T_I)$, and $\boldsymbol{L}_t$ is an array with elements L_{ij} for $i = 1, ..., I$ and $j = J+1, ..., J+T_i$. The complete-data log-likelihood is then given by

$$\begin{aligned}\ell(\Theta|\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{Z}, \boldsymbol{T}, \boldsymbol{L}_{\text{t}}) = \sum_{i=1}^{I}\sum_{j=1}^{J}\Big\{ & D_j[\log\phi + \log \text{N}(L_{ij}|\mu_i, \sigma_i) \\ & - \log(C_{ij}+1)] + (1-D_j)[\log(1-\phi) \\ & + Z_{ij}\log\pi_j + (1-Z_{ij})\log(1-\pi_j)] \\ & + (1-D_j)(1-Z_{ij})(C_{ij}\log\delta_i - \delta_i - \log C_{ij}!)\Big\} \\ & + \sum_{i=1}^{I}\sum_{j=J+1}^{J+T_i}[\log \text{N}(L_{ij}|\mu_i, \sigma_i) - \log(C_{ij}+1)].\end{aligned} \tag{1.9}$$

Let $\boldsymbol{Y}_{\text{obs}} = \boldsymbol{C}$ be the observed data. $\boldsymbol{Y}_{\text{mis }1}$ denotes $\boldsymbol{D}$ and $\boldsymbol{Y}_{\text{mis }2}$ denotes the rest of the unobserved data $(\boldsymbol{Z}, \boldsymbol{T}, \boldsymbol{L}_{\text{t}})$. Following the notation used in Dempster et al. [9], denote $\boldsymbol{Y}_{\text{mis }1}^{(t)} = \mathbf{E}(\boldsymbol{Y}_{\text{mis }1}|\boldsymbol{Y}_{\text{obs}}, \Theta^{(t)})$. It is clear from (1.9) that $\mathbf{E}(\ell(\Theta|\boldsymbol{Y}_{\text{com}})|\boldsymbol{Y}_{\text{obs}}, \boldsymbol{Y}_{\text{mis }1}, \Theta^{(t+\frac{k-1}{K})})$ is linear in $\boldsymbol{Y}_{\text{mis }1}$. Therefore, the outer E step can be simplified by computing $\boldsymbol{Y}_{\text{mis }1}^{(t)}$ only once per iteration and then run K inner EM cycles with $(\boldsymbol{Y}_{\text{obs}}, \boldsymbol{Y}_{\text{mis }1}^{(t)})$ treated as observed data. Specifically, the outer E step calculates $w_j^{(t)} = \mathbf{E}(D_j|\boldsymbol{C}, \Theta^{(t)})$, the conditional expectation of $\boldsymbol{D}$. Then the inner EM treats $(\boldsymbol{C}, \boldsymbol{w}^{(t)})$ as observed data, where $\boldsymbol{w}^{(t)} = (w_1^{(t)}, ..., w_J^{(t)})$. Since $\boldsymbol{Z}$ and $\boldsymbol{L}_t$ are independent, we are essentially nesting two inner EM algorithms here. The inner E step involving $\boldsymbol{Z}$ can be simplified to calculate the conditional expectation of Z_{ij} given $(\boldsymbol{C}, \boldsymbol{w}^{(t)}, \Theta^{(t+\frac{k-1}{K})})$ by noting that the complete-data log-likelihood (1.9) is linear in Z_{ij} and $\sum_{i=1}^{I} Z_{ij}$ is the complete-data sufficient statistic for π_j. The inner E step involving $\boldsymbol{L}_t$ and $\boldsymbol{T}$ calculates the expected values of the sufficient statistics $s_i = \sum_{j=1}^{J+T_i} D_j L_{ij}$ and $S_i = \sum_{j=1}^{J+T_i} D_j L_{ij}^2$ for the normal distribution parameters (μ_i, σ_i) conditioning on the

observed data, $\boldsymbol{w}^{(t)}$ and $\Theta^{(t+\frac{k-1}{K})}$. For detailed steps of our nested EM algorithm, see Appendix A.2.

Compared to (1.6) and (1.7), the nested EM algorithm greatly simplifies the process of updating $(\boldsymbol{\pi}, \boldsymbol{\delta}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ by providing closed-form formulas and so avoids the need for high-dimension optimization as well as the issue of numerical instability.

Finally, we need to determine the number of cycles K in each EM iteration. Note that the purpose of the inner EM cycles is not to reach convergence, but rather to move quickly towards the mode of the incomplete-data log-likelihood with a small computational cost. Because EM algorithms usually make a significant progress in the first few iterations, van Dyk [61] suggested to fix K at some small value. We choose $K = 5$ in our implementation.

1.3.5. Normalizing gene expression and identifying expressed genes

Once the mixture model is fitted and the MLE $\hat{\Theta}$ is obtained from the nested EM algorithm, the normalized expression N_{ij} of gene j from sample i can be obtained by

$$N_{ij} = \mathbf{E}\left(D_j | \boldsymbol{C}_j, \hat{\Theta}\right) \times \left\{ L_{ij} - \left[\hat{\mu}_i + \frac{\psi\left(-\frac{\hat{\mu}_i}{\hat{\sigma}_i}\right)}{\Phi\left(-\frac{\hat{\mu}_i}{\hat{\sigma}_i}\right)} \hat{\sigma}_i \right] \right\},$$

where $\mathbf{E}\left(D_j | \boldsymbol{C}_j, \hat{\Theta}\right)$ is calculated by (1.4) from the last E step which estimates the probability of gene j being expressed, and the term in the braces is the estimated expression for an expressed gene after removing the sample-specific effect. Clearly, the normalized expression is in the log scale. It is easy to use MIXnorm for detecting expressed genes. Gene j is identified as expressed if $w_j^{(t)} > c_w$ at convergence, where $c_w \in [0, 1]$ is a cut-off value. As shown in Table A.1 in Appendix A.3.2, the choice of c_w seems not to have a noticeable impact on the classification performance of MIXnorm. In fact, $w_j^{(t)}$ in (1.4) is determined by the ratio of $p(\boldsymbol{C}_j | D_j = 1, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)})$ and $p(\boldsymbol{C}_j | D_j = 0, \pi_j^{(t)}, \boldsymbol{\delta}^{(t)})$, which are the likelihoods of the data modeled by TN and ZIP distributions, respectively. These two

likelihoods are usually separate well. Thus, it is not surprising for us to observe that in our simulations, $w_j^{(t)}$ was either close to zero or close to one when MIXnorm converges, and so different threshold values in a quite wide range would not affect the detection performance much. We mention that MIXnorm is directly applicable to FF or like samples. This is because FF samples may be viewed as a reduced case of FFPE samples (i.e., little degradation in FF samples compared to severe and diverse degradation in FFPE samples). However, it is inappropriate to apply existing methods to FFPE data as they do not have the capacity to deal with the more complex data structure, as mentioned in the introduction.

1.4. Results

1.4.1. Simulation

Simulation studies were conducted to compare MIXnorm with five methods commonly used for normalizing RNA-seq data, including Upper Quartile (UQ), PoissonSeq (PS), DEseq, Reads Per Million (RPM) and trimmed mean of M values (TMM). Here, we used a data-generating model that is modified from the proposed mixture model, in order to better mimic real situations. In our six simulation studies, we examined the impact of the proportion of expressed genes on the normalization performance in study I, the impacts of the sample-specific effects in study II, the impacts of the gene-specific effects in study III, the sensitivity to violations of model assumptions in study IV, the performance of directly and separately applying MIXnorm when differential expressed (DE) genes exist across different conditions in study V, and the relationship between the sample size (number of genes) and computing time of MIXnorm in study VI. For details about the data-generating models, process and simulation settings, see Appendix A.3.1. All the results are reported and discussed in Appendix A.3.2. We find that MIXnorm consistently outperforms the

existing methods in nearly all the settings and is more robust to changes of sample-specific or gene-specific effects as well as violations of model assumptions. We also find that the computing time of MIXnorm has almost a perfect positive linear relationship with the sample size and number of genes, respectively. When DE genes exist, we recommend applying MIXnorm to normalize data from different groups separately instead of applying it to pooled data.

1.4.2. Data application

Soft tissue sarcomas data. The soft tissue sarcomas dataset was used for our exploratory analysis in Section 1.2, which contains expression levels for $20,242$ protein coding genes from paired FF and FFPE samples of 41 patients measured by RNA-seq. Note that the availability of paired FF samples would enable us to quantitatively assess and compare the performance of different RNA-seq normalization methods. Since the true (normalized) gene expression is unknown, it is generally difficult to compare the performance on real data. Nevertheless, such paired FF data, after normalization to remove technical effects, can be used as a surrogate of the truth. This is because FF tissues are known to maintain RNA very well (much lower degradation of RNA and no methylene crosslink between RNA and proteins) and thus are considered as a gold standard for most molecular assays [52]. To be specific, the gene-wise Pearson correlations between normalized FFPE and FF data (in the log scale) were computed and compared among the six different methods (MIXnorm, DEseq, RPM, TMM, PS and UQ). The correlations between original FFPE and FF data (without using any normalization method, also in the log scale) were computed to provide a baseline. We used the same approach as described in Appendix A.3.2 to deal with genes that have zero standard deviations when computing Pearson correlations.

Since the soft tissue sarcomas data were collected primarily for the analysis of the CINSARC gene signature, we evaluated the performance on all the $20,242$ genes as well

Table 1.1: Data applications: the left and middle panels show gene-wise correlations between normalized FFPE and FF expression for soft tissue sarcomas data; the right panel shows gene-wise correlations between normalized FFPE and RNAlater expression for ccRCC data. The Upper-Quartile method (UQ) failed to work for soft tissue sarcomas data due to excess zeros.

Method	Soft tissue sarcomas						ccRCC		
	CINSARC gene signature			20,242 protein coding genes			$18,458$ protein coding genes		
	1st. Qu.	Median	3rd. Qu.	1st. Qu.	Median	3rd. Qu.	1st. Qu.	Median	3rd. Qu.
MIXnorm	**0.333**	**0.455**	**0.517**	**0.098**	**0.235**	**0.384**	**0.304**	**0.524**	**0.789**
DEseq	0.165	0.260	0.354	0.019	0.160	0.298	0.203	0.418	0.609
RPM	0.146	0.243	0.350	0.010	0.156	0.297	0.204	0.422	0.612
TMM	0.010	0.098	0.161	0.021	0.159	0.291	0.110	0.267	0.463
PS	-0.126	0.002	0.154	-0.374	-0.148	0.036	0.071	0.285	0.491
UQ	-	-	-	-	-	-	0.187	0.407	0.610
Original	0.020	0.107	0.181	0.011	0.146	0.277	0.142	0.299	0.485

as the 67 genes in the gene signature. Table 1.1 summarized gene-wise correlations for the CINSARC gene signature in the left panel, and gene-wise correlations for all the genes in the middle panel, where genes in the CINSARC signature show considerably higher correlations than the population of the protein coding genes for the methods MIXnorm, DESeq, and RPM. Among all the methods, MIXnorm results in the highest quartiles. DEseq is the second best in this real data application, which is also one of the recommended normalization methods for high-throughput RNA sequencing data [11]. The most straightforward normalization method RPM gives better result compared to PoissonSeq and TMM for genes in the CINSARC signature. UQ failed to normalize the data. After removing genes with zero raw read counts across all samples, there are still several FFPE samples with more than 75% zero counts, which makes the scaling factors of UQ equal zero. Note that for DEseq, genes with at least one zero read count were removed before calculating the scaling factors, which removed 97% of genes in the FFPE RNA-seq data.

Figure 1.2 plots the normalized FFPE and FF expression levels in the log scale for all 67 genes in the CINSARC signature, where the left panel shows scatterplots for MIXnorm, TMM and DESeq, and the right panel shows scatterplots for RPM, PS and the original data. We observe that all these genes were identified as expressed genes by MIXnorm,

as one may expect. For all methods except MIXnorm, there are genes whose normalized FF expression is high but normalized FFPE expression is low or almost zero, resulting in an obvious horizontal line at $y = 0$. This suggests that the existing methods were not able to handle genes with zero or low expression well in FFPE samples. The Pearson correlation coefficients between normalized FF and FFPE expression levels reported in Figure 1.2 also indicate that MIXnorm has the best overall performance for this gene signature.

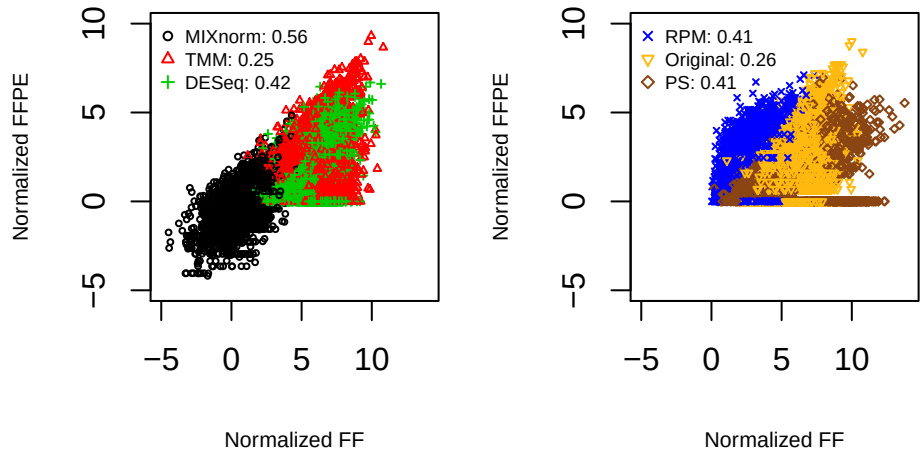


Figure 1.2: Soft tissue sarcomas data example: the normalized FFPE vs. FF expressions in the log scale from all 41 samples for all 67 genes in the CINSARC gene signature. The left panel shows scatterplots for MIXnorm, TMM and DESeq, and the right panel shows scatterplots for RPM, PS and the original data (without any normalization). Pearson correlation coefficients are reported for each method in the legend.

Clear cell renal cell carcinoma (ccRCC) data. Our second application uses the ccRCC dataset from Eikrem et al. [13], of which RNA-seq data from FFPE samples were used to simulate synthetic data in Section 1.4.1. ccRCC is the most common subtype of renal cell carcinoma, and is resistant to conventional chemotherapy and radiotherapy. Therefore, it is only curable by early surgical tumor removal when a surgery is able to eradicate the disease. Reversal of cancer gene expression is predictive of therapeutic potential. Much effort has been made to develop molecular signatures of disease progression for ccRCC. Among many, Eikrem et al. [13] aimed to validate RNA-seq outcomes

from FFPE biopsies with paired RNAlater stored samples for ccRCC patients. The data include 16 adult patients from Haukeland University Hospital. Four core biopsies were obtained from each patient, including two with ccRCC and two from adjacent normal tissues. The two pairs of ccRCC and normal tissues were then stored in FFPE and RNAlater, respectively. The RNA-seq data obtained from these tissues contain genes annotated by Ensembl. We converted the Ensembl ID to the HGNC symbol by Biomart and kept the protein coding genes only. The processed dataset contains $18,458$ protein coding genes and 32 paired FFPE and RNAlater samples.

The proposed MIXnorm model in Section 1.3.1 is quite general, we believe. In practice, however, model assumptions (mainly, zero inflation and truncated normality) may not roughly hold. Thus, before applying MIXnorm to RNA-seq data, we recommend that users conduct an explanatory analysis as we did for the soft tissue sarcomas data using log transformed read counts (i.e., Figures 1.1), and look for clear bimodality with the first spike occurring at zero and approximately Gaussian curves around the second mode for most samples. The empirical densities of log read counts for the 32 paired FFPE and RNAlater samples, shown in Figure A.7 of Appendix A.4, suggest the suitability of the proposed MIXnorm for the ccRCC data. We also suggest conducting a confirmatory analysis after applying MIXnorm, by visually examining Q-Q plots or conducting distributional tests, to check whether the assumption of truncated normality is adequate for expressed genes in most of the samples. Both Q-Q plots (Figure A.8) and p-values from Kullback-Leibler tests (Figure A.9) suggest that there was no gross departure from the assumed TN distributions for ccRCC FFPE data. For detail, see Appendix A.4.

RNAlater is an aqueous, non-toxic tissue storage reagent that rapidly permeates tissues to stabilize and protect cellular RNA in unfrozen specimens. It is considered to be comparable to the fresh frozen procedure. Therefore, the normalized RNAlater data were used as a surrogate of the gold standard in this application. The $18,458$ gene-wise Pearson correlations between normalized FFPE and RNAlater data in the log scale were

computed to evaluate the performance. As suggested in Section 1.4.1, we performed MIXnorm separately on the tumor and normal tissues. The gene-wise correlations were then calculated from all 32 paired samples. The quartiles and median of the correlations are summarized in the right panel of Table 1.1. Compared to the original data, the MIXnorm, DEseq, RPM and UQ normalized data improve the gene-wise Pearson correlations. Clearly, MIXnorm performs the best among all methods. DEseq, RPM and UQ have similar quartiles in this application. We note that the ccRCC FFPE data have better quality compared to the soft tissue sarcomas FFPE data. In fact, DEseq only needs to remove 32% of genes that have zero raw read counts. UQ needs to remove 5% of genes with zero raw read counts across all samples. Obviously, the performance of the quantile-based methods heavily depends on the data quality. Further, in real applications with FFPE samples, none of the existing normalization methods is robust while MIXnorm seems to be superior. After all, only MIXnorm is specifically designed for FFPE RNA-seq data.

As requested by one of the reviewers, we provide additional results in Table A.2 of Appendix A.4 to investigate the impact of removing genes with low expression on the performance of different normalization methods. We find that MIXnorm gives similar results regardless of removal of such genes or not, and maintains its top performance.

This paired design allows us to conduct differential expression analysis between ccRCC and normal conditions using both FFPE and RNAlater samples, and to access the validity of using FFPE samples for such analysis. We identified differentially expressed (DE) genes (Benjamini-Hochberg adjusted p value < 0.05 from paired t-tests and absolute log2 fold change > 2) from each of the two tissue sources based on the different normalization methods and report results in Table 1.2. We find that MIXnorm gives the highest number of common DE genes from the two sources. Furthermore, among the two sets of top 20 DE genes identified from RNAlater and FFPE samples, MIXnorm gives the highest number (13) of common genes while the other methods gives 9 or less. Table 1.3 sum-

marizes the FFPE and RNAlater log2 fold changes (FCs) of 13 shared genes identified by MIXnorm, of which Spearman correlation is 0.88.

Table 1.2: ccRCC data example: summary of differential expression analysis based on different normalization methods. The second column is the number of DE genes identified from the FFPE data; the third column is the number of DE genes identified from the RNAlater data; the fourth column is the number of common genes between the two sets of DE genes; the last column is the number of common genes among the two sets of top 20 DE genes from FFPE and RNAlater.

	FFPE DE genes	RNAlater DE genes	Common DE genes	Common top 20 DE
MIXnorm	1488	1482	1036	13
DEseq	1014	951	680	7
RPM	999	926	676	9
TMM	1073	1067	632	7
PS	1001	1300	652	8
UQ	1002	943	679	8
Original	1041	1096	646	9

Table 1.3: ccRCC data example: the 13 shared genes among the two sets of top 20 DE genes from FFPE and RNAlater, ordered by the absolute value of the RNAlater log2 FC.

	CA9	SLC6A3	NDUFA4L2	UMOD	GP2	CLCNKA	CDCA2	TNFAIP6	SLC4A11	KNG1	SLC12A1	AQP2	NELL1
RNAlater log2 FC	8.04	7.22	6.39	-6.15	-5.51	-5.28	5.23	5.17	-5.08	-5.02	-4.95	-4.92	-4.77
FFPE log2 FC	5.66	6.31	4.89	-5.62	-4.96	-5.69	5.05	5.45	-5.22	-5.03	-4.89	-4.89	-5.02

Table 1.3 confirms strong over-expression of SLC6A3 and CA9 and under-expression of UMOD and SLC12A1 in ccRCC tissues, previously identified by immunohistochemistry studies [13, 51, 65]. The normalized expression levels of the four genes from FFPE samples are plotted in Figure 1.3, which clearly show the up- and down-regulation of these genes. It is interesting to note that the most up-regulated gene SLC6A3 identified by FFPE data is associated with the process of producing dopamine transporter (DAT). The importance of expression changes of DAT has been widely studied in Parkinson's syndrome and attention-deficit/hyperactivity disorder (ADHD) [40, 51]. Recently, Hansson et al. [21] studied fresh frozen samples from The Cancer Genome Atlas (TCGA)

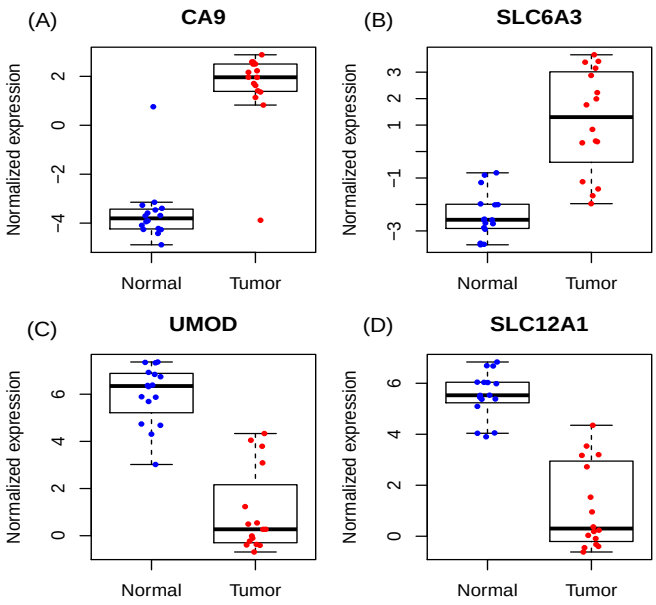


Figure 1.3: ccRCC data example: normalized expressions levels of CA9 (Panel A), SLC6A3 (Panel B), UMOD (Panel C) and SLC12A1 (Panel D) from FFPE samples.

database and identified the DAT SLC6A3 as a specific biomarker for ccRCC. Our application demonstrates that SLC6A3 expression measured from FFPE samples may also serve as a highly specific biomarker for ccRCC. Tostain et al. [57] presented a comprehensive study on the Carbonic anhydrase 9 (CA9) as a marker for diagnosis, prognosis and treatment in ccRCC. It has been shown that CA9 mRNA expression measured by reverse transcription polymerase chain reaction (RT-PCR) and CA9 antigen detected by ELISA are promising molecular markers for diagnosis and prognosis of ccRCC [57]. Our analysis further suggests that CA9 expression measured from FFPE RNA-seq may also serve as a molecular marker for ccRCC. It is worth noting that among the common top 20 DE genes, all normalization methods except MIXnorm failed to identify SLC12A1. SLC12A1 is a protein coding gene that encodes kidney specific sodium-potassium-chloride cotransporter and is known to be associated with Bartter Syndrome and Antenatal Bartter Syndrome. Schrödter et al. [51] found that SLC12A1 expression was decreased in FF ccRCC tissues. Our analysis finds that after MIXnorm normalization, FFPE tissues are also able to detect down-regulation of SLC12A1.

1.5. Discussion

In recent years, many studies have been conducted to evaluate the feasibility of using FFPE specimens with RNA-seq, the dominant high-throughput technology in gene expression profiling. These studies have collectively provided overwhelming evidence of reliable expression profiles obtained from FFPE specimens. However, none of the existing methods was developed for normalizing FFPE RNA-Seq data, a critical step in data analysis. Motivated by real data from FFPE tissues, we developed a two-component mixture model, which intends to capture major characteristics of the FFPE RNA-seq data accurately. Due to the resulting complex likelihood function, direct maximization can be unrealistic and time-consuming. By designing a nested EM-type algorithm that is easy to implement and computationally efficient, we greatly reduced the difficulty of finding the MLE.

We have shown that MIXnorm maintains top performance across various simulation settings and in two real data applications, compared to five existing RNA-seq normalization methods. The advantage of MIXnorm becomes more significant when the proportion of expressed genes becomes small. This may be due to the fact that MIXnorm is able to identify expressed genes from non-expressed genes accurately, and then models the two groups separately by ZIP and TN distributions. Besides the improvement in performance, MIXnorm has two other merits: (i) it handles genes with high-proportion zeros rigorously while existing methods typically require removal of such genes beforehand; (ii) it can output a parameter that represents the proportion of expressed genes, which can serve as an overall quality score for an RNA-seq experiment using FFPE tissues.

In MIXnorm, we employed zero-inflated Poisson (ZIP) instead of zero-inflated Negative Binomial (ZINB) distributions to model non-expressed genes. This is mainly because after sorting out expressed genes, over-dispersion would not be a major issue. Also, NB and Poisson models often give similar parameter estimates, and NB fitting leads to

larger standard error (SE) estimates than Poisson fitting. However, for the purpose of normalization, the SE estimates would not affect the results. Thus, ZIP was used also for simplicity.

We mention that FF data show simpler patterns than FFPE data, which can be modeled by simplifying the FFPE model proposed in Section 1.3, i.e., setting $\sigma_i^2 \equiv \sigma^2$ in (1.2), as Fig 1.1(d) shows a constant variance of L_{ij} across samples. In our soft tissue sarcomas data example, the estimated SDs of the TN distributions are much more consistent for the FF samples (coefficient of variation $CV = 0.05$) than those for the FFPE samples ($CV = 0.40$). That's why one can apply MIXnorm directly to FF or like samples, as discussed in Section 1.3.5. However, for computational efficiency, we can further simplify the nested EM algorithm to accommodate a common variance σ^2, to be able to run faster for FF data.

Single-cell RNA-sequencing (scRNA-seq) has become widely used for transcriptome analysis in many biological studies. Like FFPE RNA-Seq data, scRNA-seq data have the sparsity feature. However, we do not recommend that MIXnorm be applied to such data blindly. scRNA-seq experiments aim to capture the heterogeneity among individual cells, where different cell types or transient states may make a gene expressed only in some cell subpopulations [59]. As discussed in the introduction, commonly used normalization methods for bulk RNA-seq data (eg., DEseq, TMM, PS, UQ, etc.) are typically based on scaling factors, which assume that most of the genes are not differently expressed across different samples in a study [1, 49]. Obviously, this key assumption is not valid for scRNA-seq data. We note that MIXnorm is essentially a scaling factor based method, too. The scaling factor for each sample is estimated by the mean of the sample-specific TN distribution. In particular, MIXnorm assumes that a gene is either expressed or not across all samples in a study, which is invalid for scRNA-seq data. Thus, we believe that MIXnorm is not suitable for normalizing scRNA-seq data.

CHAPTER 2

SMIXnorm: Fast and Accurate RNA-seq Data Normalization for Formalin-Fixed Paraffin-Embedded Samples

2.1. Introduction

The application of next-generation sequencing (NGS) on measuring transcript abundance is widely known as RNA-seq. RNA-seq works by sequencing a library of cDNA fragments in a high-throughput manner in order to provide a comprehensive quantification of transcriptomic activities in biological samples [45]. In practice, normalization is an important step in RNA-seq data analysis since raw counts are often not directly comparable between samples [11]. Normalization brings out the biologically relevant information in gene expression by removing the systematic noise that arises from various experimental reasons (such as batch effect, lane effect, sequencing bias, etc.). Recent studies have shown that the raw sequencing data without normalization could cause invalid inference from many conventional statistical analyses and measurements [45].

Fresh-frozen (FF) tissue biospecimens are considered the gold standard in molecular analysis using gene expression, as freezing preserves RNA well. A number of normalization methods have been well studied on FF bulk RNA-seq data [11]. Most existing methods, including Reads Per Million (RPM), Upper-Quartile (UQ), DESeq, Trimmed Mean of M-values (TMM), etc., are based on scaling factor estimation, where the normalized expression is obtained by dividing the raw count by an estimate of the sample-specific scaling factor. For example, RPM [38] estimates the scaling factor for each sample by the total number of reads divided by $1,000,000$. Similarly, UQ [4] estimates the scaling

factor by the upper quartile of counts across all genes for each sample. DESeq [1] normalization first calculates the geometric means of the counts for all genes as an average reference library. Then the ratio of the count in each sample to that in the reference library is computed. The scaling factor for each sample is estimated as the median of this ratio across all genes. TMM [49] normalization selects one sample as a reference, and the M values are calculated as the log ratios of the read count between each test sample and the reference for all genes. Then for each test sample, the scaling factor is estimated by the weighted mean of M values after removing the genes with extreme average expression or M values. On the other hand, normalization can be done implicitly by accounting for the size factor as a term in the RNA-seq data model. PoissonSeq (PS) [31] models RNA-seq data by a Poisson log-linear model, where a set of sample-specific parameters is included as offset parameters in the linear predictor to account for different sequencing depths.

Recently, there has been increasing interest in performing transcriptome profiling on Formalin-Fixed Paraffin-Embedded (FFPE) tissues, as they are widely available from routine diagnostic sample preparation [39]. Being able to successfully measure mRNA abundance from FFPE samples could greatly facilitate biomarker discoveries and genomic studies of clinical samples [19, 20]. The major challenge of adapting FFPE biospecimens in molecular analysis is that the chemical process is designed to preserve cellular proteins rather than preserving RNA. As a result, RNA from FFPE tissues is usually degraded, which could limit gene expression analysis. Studies have shown that the fixation process, storage time, specimen size and conditions play important roles in the RNA quality from FFPE samples [63]. Although such samples may suffer from the chemical modifications and continued degradation over time, recent studies have shown that RNA-seq can measure the RNA expression from FFPE samples in sufficient quality [32]. Due to chemical modifications and the RNA degradation, the RNA quality and abundance extracted from the FFPE samples vary a lot. Therefore, the normalization step is even more important for RNA-seq data measured from FFPE samples than those measured from

FF samples. However, none of the existing methods were specifically designed and validated on FFPE samples. The mRNA expression measured from FFPE can have lower quality and higher sparsity (i.e. many zero counts occur), compared with FF samples. Consequently, the zero-inflation makes the assumption of model-based FF RNA-seq normalization invalid. For most scaling factor-based RNA-seq normalization, practitioners need to discard genes with many zeros beforehand, which may be a significant portion of the data when applied to FFPE samples. To the best of our knowledge, MIXnorm [67] is the only method that is designed to normalize FFPE RNA-seq data but that is applicable to FF RNA-seq data as well. MIXnorm addresses the zero inflation by separately modeling the expressed genes and non-expressed genes in a mixture model. It consistently outperforms commonly used FF RNA-seq normalization methods. However, MIXnorm relies on a complex latent variable structure for model fitting, which causes a heavy computational burden for large data sets. We show in this paper that the statistical model of MIXnorm can be properly simplified to still capture the main characteristics of the FFPE RNA-seq data. We propose a simplified version of MIXnorm, labeled SMIXnorm, for FFPE RNA-seq data normalization. The fitting of SMIXnorm requires a less complicated latent variable structure. We show through simulation studies and real data applications that SMIXnorm retains almost the same performance as MIXnorm, while greatly reducing the computing time.

More importantly, there is a lack of platforms that integrate existing methods and produce normalized data by different methods. Evans et al. [15] mentioned that the selection of normalization methods played an important role in downstream analysis due to the different assumptions those methods made. Furthermore, it is important to raise the awareness of the separate normalization methods for FFPE samples, as more and more applications involve RNA-seq data from such samples. We developed a web portal, *RNA-seq Normalization (RSeqNorm)* (http://lce.biohpc.swmed.edu/rseqnorm/), to conduct normalization for both FF and FFPE RNA-seq data. It offers seven normalization methods, with accompanying diagnostic plots for users to visually examine the RNA-seq data quality.

Based on this platform, we compared different normalization methods using both comprehensive simulation studies and real data applications. These results, together with the *RSeqNorm* web portal, will facilitate users to select the best normalization method for their application.

The paper is structured as follows. In Section 2.2.1, we present the SMIXnorm method. The statistical model of SMIXnorm is simplified from that of MIXnorm. An efficient nested Expectation-Maximization (EM) algorithm is designed for model fitting. We further justify the simplifications by comparing SMIXnorm to MIXnorm from a technical point of view. An introduction of the web portal *RSeqNorm* is given in Section 2.2.2. Section 2.3 reports simulation studies and real data analyses. Finally, a brief concluding discussion is made in Section 2.4.

2.2. Materials and methods

2.2.1. The SMIXnorm method

The simplified statistical model. Assume the RNA-seq count data from FFPE samples can be summarized by a matrix $\boldsymbol{C}_{I\times J}$, where C_{ij} is the number of reads in sample i for gene j. We adopt a similar latent variable framework as in MIXnorm to address the zero inflation of such data. That is, the binary latent variable $D_j = 1$ indicates gene j is expressed and $D_j = 0$ indicates gene j is not expressed in the study for $j = 1, ..., J$. Then we model the count data as a mixture of zero-inflated Poisson (ZIP) and normal distributions,

$$C_{ij} \sim \text{ZIP}(\pi_j, \delta), \text{if } D_j = 0, \tag{2.1}$$

$$L_{ij} \sim \mathrm{N}\ (\mu_i, \sigma_i^2), \text{if } D_j = 1, \tag{2.2}$$

$$D_j \sim \mathrm{Ber}(\phi),$$

where $L_{ij} = \log(C_{ij} + 1)$ denotes the log transformed count, $0 \leq \pi_j, \phi \leq 1$, $\delta, \mu_i, \sigma_i \geq 0$ for $i = 1, ..., I$ and $j = 1, ...J$. The model assumes an unobserved variable D_j which follows a Bernoulli distribution with parameter ϕ. Genes with $D_j = 0$ are considered not expressed. These include low-expression genes that have abundance below detection limit, or biologically non-expressed genes that are absent from the biological sample of interest, or genes that should have been expressed but suffer from high-level mRNA degradation. The observed counts from non-expressed genes are due to background noise and are modeled by a zero-inflated Poisson distribution with gene-specific probability of extra zeros π_j and a common expected Poisson count δ. Genes with $D_j = 1$ are expressed genes and we model the log counts of those genes by a normal distribution with sample-specific location and scale parameters. Compared to the model of MIXnorm, we use the normal distribution in (2.2) instead of a truncated normal distribution and a common Poisson mean δ rather than the sample-specific mean δ_i. As will be discussed in the following section, these would greatly facilitate the computation while not hurting much the performance. Note that the normal distribution assigns positive densities to negative log counts L_{ij}, which never occur in real data. However, it is reasonable to assume that the negative values only take a negligible portion of the density in modeling expressed genes as the mean (log) counts of such genes are usually well above zero. Further, SMIXnorm is directly applicable to FF or like samples based on the same argument in Yin et al. [67] that FF samples may be considered a reduced case of FFPE samples.

Model fitting. Let $\Theta = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi}, \delta, \phi)$ denote the set of all parameters in the simplified model. The observed data likelihood as a function of Θ is defined as follow:

$$\begin{aligned} L(\Theta|\boldsymbol{C}) &= \prod_{j=1}^{J} p(\boldsymbol{C}_j|\Theta) \\ &= \prod_{j=1}^{J} [p(\boldsymbol{C}_j|D_j = 1, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(D_j = 1|\phi) \\ &+ p(\boldsymbol{C}_j|D_j = 0, \pi_j, \delta) p(D_j = 0|\phi)] \\ &= \prod_{j=1}^{J} [\prod_{i=1}^{I} p(C_{ij}|D_j = 1, \mu_i, \sigma_i) \cdot \phi \\ &+ \prod_{i=1}^{I} p(C_{ij}|D_j = 0, \pi_j, \delta) \cdot (1 - \phi)], \end{aligned} \tag{2.3}$$

where $p(C_{ij}|D_j = 1, \mu_i, \sigma_i)$ is the probability density function (pdf) of C_{ij} for expressed genes such that $\log(C_{ij}+1)$ follows the normal distribution in (2.2), and $p(C_{ij}|D_j = 0, \pi_j, \delta)$ is the zero-inflated Poisson probability mass function (pmf) for non-expressed genes as described in (2.1). Note that a continuous distribution is used to approximately model the discrete random variable $\log(C_{ij} + 1)$. See Yin et al. [67] for a detailed justification for the approximation.

A common approach to obtain the maximum likelihood estimate (MLE) of Θ is to treat $(\boldsymbol{C}, \boldsymbol{D})$ as the complete data and update the parameter estimates iteratively by an EM algorithm. However, direct implementation of the EM algorithm requires Newton-Raphson type maximization to update the parameters within the ZIP component, which may cause the EM algorithm fail to converge due to numerical instability. By a similar approach in Yin et al. [67], we update the parameter estimates from the ZIP component by nesting another EM algorithm and avoid the Newton-Raphson type optimization. Specifically, we construct another latent variable Z_{ij} for genes not expressed so that the ZIP distribution can be treated as a mixture of two states, the perfect zero state and the Poisson state. Assume $Z_{ij} = 1$ when C_{ij} is from the perfect zero state and $Z_{ij} = 0$ when C_{ij} is from

the Poisson state, satisfying $Z_{ij}|D_j = 0 \sim \text{Ber}(\pi_j)$. Let $\boldsymbol{Y}_{\text{com}} = (\boldsymbol{C}, \boldsymbol{D}, \boldsymbol{Z})$ denotes the complete data. The complete-data log-likelihood with latent variables $\boldsymbol{D}$ and $\mathbf{Z}$ is given by

$$\begin{aligned}\ell(\Theta|\mathbf{C}, \boldsymbol{D}, \mathbf{Z}) = \sum_{j=1}^{J}\sum_{i=1}^{I}\Big\{ & D_j \left[\log\phi + \log \mathrm{N}(L_{ij}|\mu_i, \sigma_i) - \log(C_{ij}+1)\right] \\ & + (1-D_j)\left[\log(1-\phi) + Z_{ij}\log\pi_j + (1-Z_{ij})\log(1-\pi_j)\right] \\ & + (1-D_j)(1-Z_{ij})\left[C_{ij}\log\delta - \delta - \log C_{ij}!\right]\Big\}. \end{aligned} \tag{2.4}$$

The outer EM treats $\boldsymbol{C}$ as observed data and $\boldsymbol{D}$ as missing data. Let $\Theta^{(t)} = (\boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)}, \boldsymbol{\pi}^{(t)}, \delta^{(t)}, \phi^{(t)})$ denote the set of current parameter estimates after t iterations of the algorithm. The distribution of $\boldsymbol{D}$ given the observed data and $\Theta^{(t)}$ is

$$p(\boldsymbol{D}|\boldsymbol{C}, \Theta^{(t)}) = \prod_{j=1}^{J}\frac{p(\boldsymbol{C}_j, D_j|\Theta^{(t)})}{p(\boldsymbol{C}_j|\Theta^{(t)})} = \prod_{j=1}^{J}\left(w_j^{(t)}\right)^{D_j}\left(1-w_j^{(t)}\right)^{1-D_j},$$

where

$$w_j^{(t)} = \frac{\phi^{(t)}p(\boldsymbol{C}_j|D_j=1, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)})}{\phi^{(t)}p(\boldsymbol{C}_j|D_j=1, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)}) + (1-\phi^{(t)})p(\boldsymbol{C}_j|D_j=0, \pi_j^{(t)}, \delta^{(t)})}. \tag{2.5}$$

Note that in (2.4), the complete-data log-likelihood is linear in $\boldsymbol{D}$. Therefore, the outer E step, which calculates the conditional expectation of the complete-data log-likelihood with respect to the missing data $\boldsymbol{D}$, is reduced to computing $w_j^{(t)} = \mathbf{E}(D_j|\boldsymbol{C}, \Theta^{(t)})$ only once per iteration. Within each iteration of the outer EM, the inner EM repeats K cycles and treats $\boldsymbol{Z}$ as missing data and $(\boldsymbol{C}, \boldsymbol{w}^{(t)})$ as observed data, where $\boldsymbol{w}^{(t)} = (w_1^{(t)}, ..., w_J^{(t)})$. By the similar argument and that $\sum_{i=1}^{I} Z_{ij}$ is the complete-data sufficient statistic for π_j, the kth cycle of the inner E step involving $\boldsymbol{Z}$ is reduced to calculating the conditional expectation of Z_{ij} given $(\boldsymbol{C}, \boldsymbol{w}^{(t)}, \Theta^{(t+\frac{k-1}{K})})$.

Details of the nested EM algorithm are summarized in Appendix B.1. The convergence can be detected using the change of the observed-data log-likelihood $\ell(\Theta|\boldsymbol{C})$ in

two consecutive iterations which is trivial to obtain via (2.3). We set the number of inner EM cycles $K = 5$ within each iteration in our implementation as in Yin et al. [67].

Similar to MIXnorm, SMIXnorm normalization relies on the accurate classification of the genes that are expressed or not. This is indicated by the latent variable D_j. We estimate D_j by its conditional expectation given the observed data at the last E-step. Assume that the majority of genes are not differentially expressed across samples (Robinson, Oshlack 2010). Then the means of the normal distributions μ_i's can be treated as the sample-specific noises. The normalized expression for sample i and gene j can be given by

$$N_{ij} = \mathbf{E}\left(D_j|\boldsymbol{C}_j, \hat{\Theta}\right) \times (L_{ij} - \hat{\mu}_i), \tag{2.6}$$

where $\hat{\Theta}$ denotes the MLE of Θ. In our numerical experiments, when gene j is not expressed, it is often the case that the conditional expectation $\mathbf{E}\left(D_j|\boldsymbol{C}_j, \hat{\Theta}\right) \approx 0$, which makes $N_{ij} \approx 0$; when gene j is expressed, $\mathbf{E}\left(D_j|\boldsymbol{C}_j, \hat{\Theta}\right)$ is often close to 1. Thus, we may simply output zero for genes with $\hat{D}_j = 0$, and $L_{ij} - \hat{\mu}_i$ for genes with $\hat{D}_j = 1$ in the actual implementation. The proposed method normalizes the data by subtracting the estimated sample-specific noise from the log count. Therefore, the normalized data are in the log scale.

SMIXnorm vs. MIXnorm. There are two major simplifications when comparing SMIXnorm to MIXnorm. First, SMIXnorm assumes a common Poisson mean δ for the non-expressed genes, where MIXnorm allows the sample-specific Poisson means δ_i. Note that δ appears in the normalization step (2.6) through the conditional expectation of D_j that can be computed by equation (2.5). We observe in practice that after the nested EM algorithm converges, the conditional expectation $\mathbf{E}(D_j|\boldsymbol{C}, \Theta^{(t)})$ (i.e., $w_i^{(t)}$ in equation (2.5)) is not sensitive to the choice of a common or sample specific Poisson mean. With the parameters set to their MLE, this conditional expectation mainly depends on the ratio between

$p(\boldsymbol{C}_j|D_j = 0, \hat{\pi}_j, \hat{\delta})$ (or $p(\boldsymbol{C}_j|D_j = 0, \hat{\pi}_j, \hat{\delta}_i)$ in MIXnorm) and $p(\boldsymbol{C}_j|D_j = 1, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}})$. The Poisson mean essentially reflects the background noise level and is supposed to be a small positive number. Thus, the former distribution, regardless of using $\hat{\delta}$ or $\hat{\delta}_i$, puts most of its probability mass near 0, whereas the latter distribution in practice has negligible density around small values near 0. Consequently, the conditional expectation in equation (2.5) is close to either 1 or 0 depending on whether the gene is expressed or not and reducing δ_i to δ has little effect in the normalization step.

We further simplify the model by ignoring the truncation on the normal distributions. Compared to MIXnorm, the sample-specific noise among expressed genes is captured by the mean of a normal distribution in SMIXnorm instead of the mean of a truncated normal distribution. However, we argue that these two estimates are asymptotically identical. Assume we model a set of continuous positive real numbers $\boldsymbol{X} = (X_1, ..., X_J)$ by a truncated normal distribution TN$(\theta, \tau^2, 0, +\infty)$, where θ and τ are the mean and standard deviation of the corresponding normal distribution before truncation. To estimate the mean of the truncated normal distribution, which is a function of θ and τ, say $m(\theta, \tau)$, one common approach is to first obtain the maximum likelihood estimates $(\hat{\theta}, \hat{\tau})$, then the maximum likelihood estimate of the mean is $m(\hat{\theta}, \hat{\tau})$ based on the invariance property of MLE. Another approach for a point estimate is the method of moments. The method of moments estimate of $m(\theta, \tau)$ is simply the sample mean $\bar{X}$, which is essentially the MLE of μ if we ignore the truncation and model the data with $\mathrm{N}(\mu, \sigma^2)$. Under the large sample situation, as is often the case in RNA-seq data involving many genes from whole-genome experiments, both $m(\hat{\theta}, \hat{\tau})$ and $\bar{X}$ are asymptotically consistent estimators for $m(\theta, \tau)$. The use of a normal distribution without truncation is appealing since it has closed-form parameter updates within the nested EM algorithm, while a truncated normal distribution requires additional latent variable structures and data augmentation to obtain closed-form parameter updates [67]. Overall, SMIXnorm produces similar normalized expression compared to MIXnorm whereas greatly simplifies the model fitting process, as will be confirmed in Section 2.3 by numerical evidence.

2.2.2. *RSeqNorm* web portal

The *RSeqNorm* web portal (http://lce.biohpc.swmed.edu/rseqnorm) provides a set of analysis routines for normalization of RNA-seq data from either FF or FFPE samples. The workflow is illustrated in Figure 2.1. Users provide raw sequence read count data (e.g., from RNA-seq experiments) in the form of an integer-valued matrix C. The web portal can compute the normalized expression as well as diagnostic information for download. We implement SMIXnorm, MIXnorm and five commonly used normalization methods, including Reads Per Million (RPM), Upper-Quartile (UQ), DESeq, Trimmed Mean of M-values (TMM) and PoissonSeq (PS). Though developed for FFPE data, SMIXnorm and MIXnorm are directly applicable to FF data normalization. However, other methods may cause suboptimal performance if applied to FFPE data [67]. For FF data normalization, there seems to have no unanimously best normalization method. We suggest using the methods offered by *RSeqNorm* and evaluating the normalization performance using prior information or known biological knowledge. For example, users may conduct a differential expression (DE) test following the normalization step and select the method that detects more genes that are known to be differentially expressed in the literature.

RSeqNorm accepts the raw read count matrix in a comma-separated values (CSV) file. The (i, j) element of the count matrix records how many reads have been assigned to gene j in sample i. An example input file is downloadable from the *RSeqNorm* website. Detailed file requirements are shown in Figure 2.2. SMIXnorm and MIXnorm require additional input arguments including the maximum number of iterations (range $(10, 50)$, default value 15) and the convergence threshold (recommend range $(1\mathrm{e}-5, 1)$, default value 0.01) for the nested EM algorithm. We note that the observed-data log-likelihood as a function of all parameters may have a large curvature near the MLE. However, the SMIXnorm and MIXnorm normalized expression values are not sensitive to small variations of the parameter estimates. Therefore, the convergence criterion here is defined

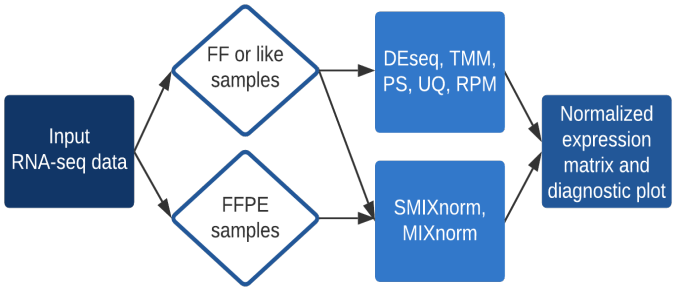
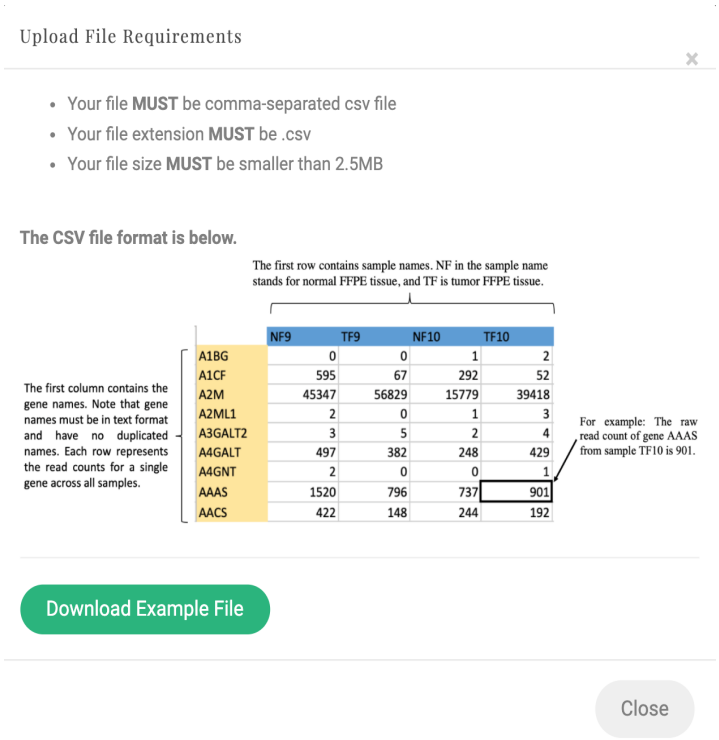


Figure 2.1: Summary of *RSeqNorm* web-portal process.

as the maximum absolute change in the parameter updates between the previous and current iterations, instead of using the change in the observed-data log-likelihood. The algorithm stops when the absolute change is smaller than the predetermined threshold value or the maximum number of iterations is reached. We mentioned in Section 2.2.1 that the conditional probability of being expressed is often close to either 0 or 1. As a consequence, the normalized expression level for a non-expressed gene is usually a trivial number (<10e-10 in our implementation to real data). Therefore, an approximate set of normalized expression is also available for SMIXnorm and MIXnorm, which normalizes genes with $\hat{D}_j = 0$ directly to 0 across all samples, where $\hat{D}_j = I(w_j^{(t)} > c_w)$ at convergence, $I(\cdot)$ is the indicator function and c_w is set to 0.5 in *RSeqNorm*. In practice, we observe that the choice of c_w is not sensitive in identifying expressed genes. The conditional probability of being expressed ($w_j^{(t)}$) and the proportion of expressed genes identified ($\hat{\phi}$) are returned in the R package *RSEQNORM*, which is downloadable from our website. Note that $\hat{\phi}$ may reflect the overall data quality for an RNA-seq experiment (i.e., a value close to 1 indicates high quality).

RSeqNorm returns the normalized expression in a CSV file with the same dimension as the user input file. User may leave the web portal after successfully submitting the job and an email notification will be sent to the user with a download link when the normalization is finished. A histogram of zero-count proportions is also returned as shown in Figure 2.3, where SMIXnorm is used on the example data provided by *RSeqNorm*.



	NF9	TF9	NF10	TF10
A1BG	0	0	1	2
A1CF	595	67	292	52
A2M	45347	56829	15779	39418
A2ML1	2	0	1	3
A3GALT2	3	5	2	4
A4GALT	497	382	248	429
A4GNT	2	0	0	1
AAAS	1520	796	737	901
AACS	422	148	244	192

Figure 2.2: *RSeqNorm* upload file requirements.

The histogram shows the distribution of zero-count proportions among all samples (represented by the horizontal axis) over all input genes. High frequencies near 0 indicate that most biologically expressed genes are actually expressed in all samples in the experiment and so the data are of high quality. Ideally, one would expect that a gene is either expressed among all samples or not expressed in any of the samples. In this ideal case, the histogram shows frequencies only at 0 and 1 on the horizontal axis.

2.3. Results

2.3.1. Simulation

We conducted simulation study to show that SMIXnorm greatly reduces computing time and maintains performance comparable to MIXnorm that is better than all FF RNA-

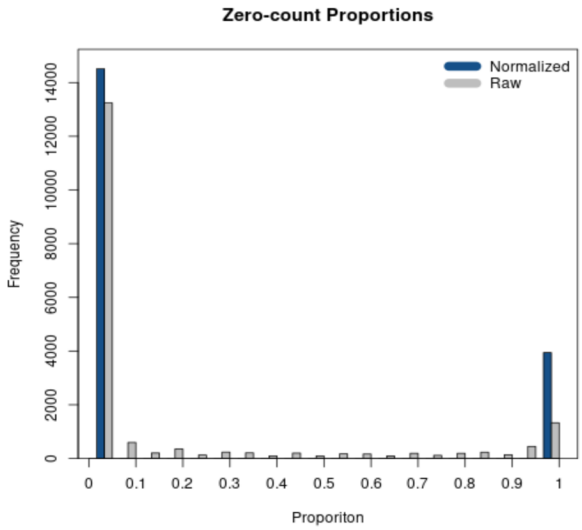


Figure 2.3: Diagnostic plot returned by *RSeqNorm* using SMIXnorm.

seq normalization methods. Our simulation study evaluates the computing time of SMIXnorm and MIXnorm with different sample sizes. Note that the SMIXnorm model cannot be used to simulate the data since it permits negative log transformed counts that do not exist in practice. Here, a modified MIXnorm model was used to generate synthetic data sets (under the same settings as in Simulation VI in Yin et al. [67]). The model parameters were set to their MLEs estimated from a public RNA-seq dataset for FFPE soft tissue sarcomas samples [30], which contains expression levels for $20,242$ protein-coding genes from 41 patients.

The sample size was set to a multiple of 41 from 41 to $2,050$. The average computing time for SMIXnorm and MIXnorm over 50 replicates is plotted against sample size in Figure 2.4. Both SMIXnorm and MIXnorm show a linear relationship between the average computing time and the sample size with Pearson correlations greater than 0.99. The linear regression fit of the SMIXnorm computing time against sample size results in a slope estimate of 0.051, compared to 0.297 from MIXnorm. To evaluate the performance of SMIXnorm and MIXnorm together with five commonly used FF RNA-seq normalization methods (PS, UQ, DESeq, RPM and TMM), we calculated the gene-wise Pearson correlations for the $20,242$ genes between the normalized and true expression for each

of the 50 simulated data set under the setting of 41 samples. The results are reported in Figure B.1 of Appendix B.2, showing that SMIXnorm and MIXnorm have almost identical performance and they consistently beat the other methods.

2.3.2. Data application

When dealing with real data where true expression is unknown, researchers often consider frozen sections as the gold standard for most molecular assays. As mentioned in the introduction, the frozen process maintains RNA well, compared to the FFPE process. Therefore, we used the paired FF RNA-seq expression after normalization as a surrogate to the true gene expression in our three data examples, in order to evaluate the performance of different normalization methods.

Colorectal cancer data. The colorectal cancer data [41] contains 54 selected FFPE tumor specimens from a larger multi-center cohort with paired FF samples. Gene expression levels were measured by whole genome RNA-seq (RNA-Acc) assay and Affymetrix GeneChip (Affy) platform on the FFPE samples and FF samples, respectively.

The activation of RAS signaling pathway is frequent in human cancer. Recent studies have shown that RAS mutations account for approximately 40% of colorectal cancers and lung cancers [41]. A number of RAS pathway activation gene expression signatures have been identified using multiple types of cancer cell lines and human FF samples. Omolo et al. [41] evaluated an 18-gene RAS pathway signature on FFPE samples in five technology platforms. We focus on the Illumina whole genome RNA-seq of FFPE samples and the gold standard FF samples measured by Affymetrix GeneChip. The Affy_FF samples were normalized using the RMA method [25, 41].

To assess the performance of the translation of the gene signature from FF to FFPE samples, we considered the same metric as in Omolo et al. [41]. The RAS pathway activation score is defined as the mean normalized expression levels of the 18 RAS genes

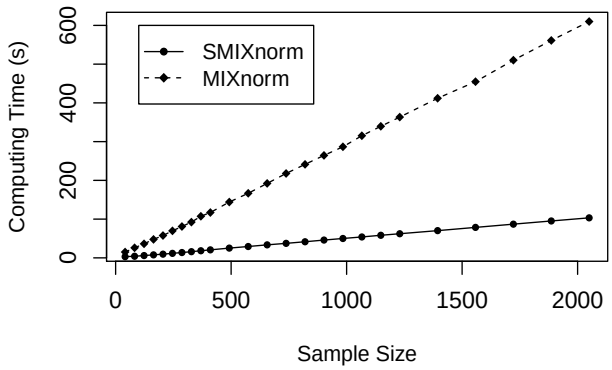


Figure 2.4: Simulation study. Average computing time of SMIXnorm and MIXnorm vs. sample size

for each sample. We calculated the Spearman correlation using the 54 pairs of FF and FFPE RAS pathway activation scores for each normalization method and summarized the results in Table 2.1. The raw data give the lowest correlation as expected. SMIXnorm and MIXnorm give almost the same results and the highest correlations. The p-value based on a two-sided permutation test for the hypothesis $H_0 : \rho = 0$ is reported in the parenthesis. SMIXnorm, MIXnorm, PS, DESeq, RPM and UQ report significant correlations at the significance level of 0.05. MIXnorm takes about 42 seconds to normalize the colorectal cancer data, while SMIXnorm only takes about 7.5 seconds.

Table 2.1: Correlations ρ between normalized FF and FFPE RAS pathway activation scores. p-values in the parenthesis are based on a two-sided permutation test for the hypothesis $H_0 : \rho = 0$.

	SMIXnorm	MIXnorm	PS	DESeq
ρ	0.343 (0.012)	0.343 (0.011)	0.324 (0.017)	0.298 (0.029)
	RPM	UQ	TMM	Raw
ρ	0.286 (0.036)	0.270 (0.049)	0.153 (0.269)	0.125 (0.366)

Soft tissue sarcomas data. The soft tissue sarcomas data [30] measure the expression levels of $20,242$ protein-coding genes from 41 patients with paired FF and FFPE

samples. To evaluate the normalization performance, the gene-wise Pearson correlations between normalized FFPE and FF expression levels (in the log scale) were computed and compared among the seven different methods (SMIXnorm, MIXnorm, DESeq, RPM, TMM, PS and UQ). A gene expression signature, Complexity INdex in SARComas (CINSARC), which contains 67 genes, has been identified as an important prognostic factor in sarcomas using fresh frozen samples. Lesluyes et al. [30] showed that CINSARC remains a potential prognostic factor using the FFPE RNA-seq data. Therefore, we evaluated the performance of different normalization methods on all the $20,242$ genes as well as the 67 genes in the CINSARC gene signature.

Figure 2.5 shows the gene-wise Pearson correlations for all the $20,242$ genes using violin plots. The dot and line in each violin plot represent the mean and standard deviation of the gene-wise Pearson correlations. The gene-wise correlations from the 67 genes in the CINSARC signature are summarized in Table 2.2 using the first, second and third quartiles. Note that UQ failed to normalize the data as the scaling factor estimates equal 0 for some samples due to the excess zero counts. The genes in the gene signature have higher correlations than those calculated from the population of all protein-coding genes for all the normalization methods. The shapes of the violin plots suggest that SMIXnorm and MIXnorm have almost identical results and SMIXnorm gives the highest mean correlation while PS gives the lowest among all the methods. The three commonly used FF RNA-seq normalization methods, TMM, DESeq and RPM, give similar results on all protein-coding genes and there is no clear improvement compared to the original correlations calculated without any normalization. However, DESeq and RPM show much higher correlations on the CINSARC gene signature compared to TMM. Overall, SMIXnorm and MIXnorm show similar results and are consistently better than the other methods. We further note that in this example, a poor choice of normalization method (e.g., UQ, TMM or PS) may yield results worse than those from original unnormalized data. MIXnorm takes about 5 minutes to normalize the soft tissue sarcomas data and we note that the algorithm reaches the default maximum number of iterations before convergence. On the

other hand, SMIXnorm converges in 6 iterations and takes about 10 seconds to normalize the data.

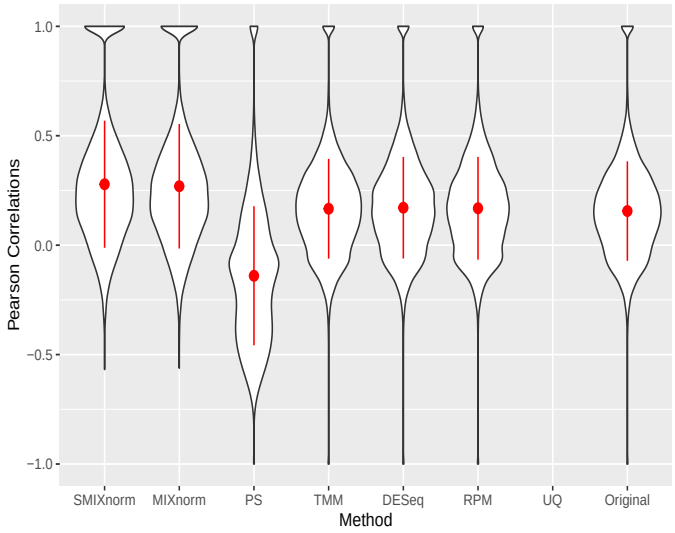


Figure 2.5: Gene-wise correlations between normalized FFPE and FF expression for soft tissue sarcomas data on all $20,242$ protein coding genes. The UQ method failed to normalize the data due to excess zero counts.

Clear cell renal cell carcinoma data. The clear cell renal cell carcinoma (ccRCC) data are available in the repository Gene Expression Omnibus from a published study [13]. ccRCC is the most common and aggressive histological type among the primary renal neoplasms. Metastasis is a major cause of ccRCC patient death due to the resistance to standard chemotherapy and radiotherapy. Hence, much effort has been made to unravel the underlying molecular mechanisms of ccRCC, for example, by applying gene expression analysis to develop molecular signatures of disease progression, which plays an important role in assessing the carcinogenesis and development of disease as well as guiding clinical decisions. The ccRCC RNA-seq data contain 32 pairs of FFPE and RNAlater samples with $18,458$ protein-coding genes converted from $64,253$ Ensembl annotated genes. Following Yin et al. [67], the RNAlater samples were considered the gold standard in this study and gene-wise Pearson correlations were computed to compare the performance of the seven normalization methods. The results are shown via violin plots

Table 2.2: Gene-wise correlations between normalized FFPE and FF expression for soft tissue sarcomas data on the CINSARC gene signature. The UQ method failed to normalize the data due to excess zero counts.

	First Qu.	Median	Third Qu.
SMIXnorm	0.344	0.465	0.529
MIXnorm	0.333	0.455	0.517
DESeq	0.165	0.260	0.354
RPM	0.146	0.243	0.350
TMM	0.010	0.098	0.161
PS	-0.126	0.002	0.154
UQ	-	-	-
Original	0.020	0.107	0.181

in Figure 2.6. Again, we observe that SMIXnorm and MIXnorm give almost the same results and consistently perform the best among all methods. It is interesting to note that TMM, which performs the best among existing FF normalization methods in soft tissue sarcomas data on all protein-coding genes, gives worse results than DESeq, RPM and UQ in this application. In fact, TMM and PS show no advantage compared to the original correlations without any normalization. MIXnorm takes about 10.5 seconds to normalize the ccRCC FFPE data, while SMIXnorm only takes about 3.3 seconds.

2.4. Discussions

We have developed an efficient normalization method, named SMIXnorm, for FFPE RNA-seq data normalization. Modified from the MIXnorm statistical model, we use a similar two-component mixture model to separately model the expressed and non-expressed genes. The simplifications of the statistical model avoid the complex likelihood function and the need of a complicated latent variable structure to invoke the nested EM algorithm. We have shown through real data applications and simulation studies that SMIXnorm greatly reduces the complexity of the likelihood function and the computing time without

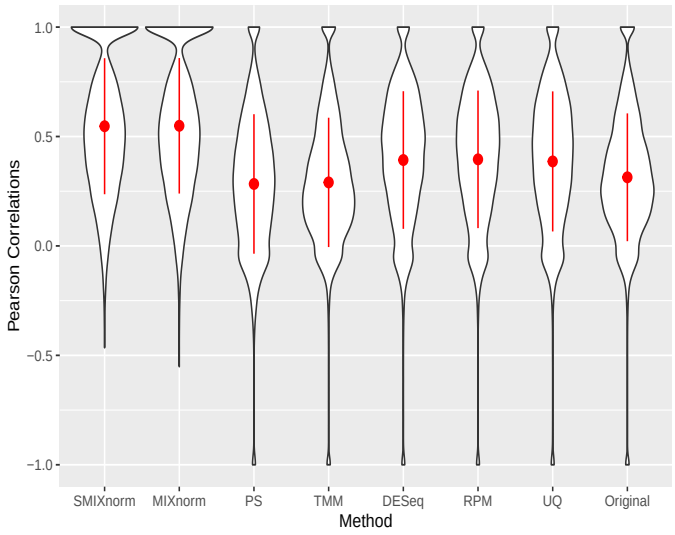


Figure 2.6: Gene-wise correlations between normalized FFPE and RNAlater for ccRCC data on $18,458$ protein coding genes.

sacrificing the performance. Though other FF normalization methods take only about 1 second to normalize each dataset in our three data applications, their performance is not comparable to that of SMIXnorm and MIXnorm for FFPE samples. Some FF normalization methods perform even worse than the use of original data without any normalization.

We mentioned in the soft tissue sarcomas application that MIXnorm failed to converge at the default maximum number of iterations and tolerance level. Due to the severely different RNA degradation levels among the 41 soft tissue sarcomas FFPE samples, 10 FFPE samples have more than $10,000$ zero counts and 9 FFPE samples have less than $3,000$ zero counts. Consequently, MIXnorm gives negative estimated sample-specific location parameters of the truncated normal distributions for those samples with a higher proportion of zero counts, which blurs the distinction between expressed and non-expressed genes. SMIXnorm models the expressed genes by normal distributions without truncation, which naturally constrains the location parameters to be non-negative in all the EM iterations as sample means are used for estimation. Thus, SMIXnorm seems to be more robust and converges faster than MIXnorm.

Recently, single-cell RNA sequencing (scRNA-seq) becomes an important technology in molecular analysis. While bulk RNA-seq measures the expression in the population level across cells, scRNA-seq allows for the cell level resolution and therefore, reveals heterogeneity of cell subpopulations. Similar to FFPE RNA-seq data, a prominent feature of scRNA-seq data is the sparsity. The high proportion of zero count arises for both biological reasons and technical reasons [35, 60]. The most commonly used scRNA-seq normalization method is SCnorm [2], which uses quantile regression to group genes by estimated count-depth relationships, and then estimates different scaling factors within each group via a second quantile regression. Other popular scRNA-seq normalization methods, including BASiCS [58], SAMstrt [26] and GRM [60], rely on spike-ins. Therefore, most scRNA-seq normalization methods are not directly applicable to the FFPE RNA-seq data that typically do not have spike-ins.

With the rapid adaption of FFPE samples in RNA-seq analysis, it is important for users to realize that different normalization procedures should be used for FFPE vs. FF data. We offer *RSeqNorm*, a comprehensive and user-friendly normalization toolkit for RNA-seq data. To the best of our knowledge, *RSeqNorm* is the only available web-based tool that integrates different normalization methods for both FFPE and FF RNA-seq data. It includes seven normalization methods, among which five are commonly used (RPM, UQ, DESeq, TMM and PS) for FF or like RNA-seq data. Though MIXnorm and SMIXnorm are specifically designed for FFPE RNA-seq normalization, they can be applied to FF data directly. However, it is generally inappropriate to implement other existing methods on FFPE data. The input for all methods is a read count matrix at the gene level. The output is an expression matrix after normalization with the same dimension as the input data. The R package, RSEQNORM, which implements SMIXnorm and MIXnorm is downloadable from the website (http://lce.biohpc.swmed.edu/rseqnorm).

CHAPTER 3

SHEAP: Detecting Spatially Expressed Genes via a Bayesian Spatial Heaping Model for Zero-Inflated Spatial Transcriptomics Data

3.1. Introduction

RNA sequencing (RNA-seq) and single-cell RNA sequencing (scRNA-seq) have been essential technologies to study organism development, normal and cancer tissues in biomedical research [10]. One key step in most molecular profiling technologies is tissue dissociation. Tissue dissociation is the process of using enzymes to digest tissue pieces and release cells from the tissue to the best possible quality for further sequencing experiments [10, 36]. However, this dissociation process leads to a loss of tissue morphological and spatial information. As a consequence, the association between sequencing data and tissue morphology has been overlooked for a long time. In order to explore such association, much effort has been devoted to combine molecular profiling experiments and imaging experiments. Recent advances in spatial molecular profiling technologies have enabled a comprehensive catalog of cell spatial organizations and their mRNA.

Spatial molecular profiling technologies are mainly of two branches: imaging-based and sequencing-based. The imaging-based technologies, including single-molecule fluorescence in situ hybridization (smFISH) [16, 46], seqFISH [14, 34] and multiplexed error-robust FISH (MERFISH) [7], first hybridize RNAs with fluorescent probes in situ. Then the fluorescent sequences are quantified by mass spectrometry or microscopy. Sequencing-based technologies, such as spatial transcriptomics technology (ST) [53], high-definition

spatial transcriptomics (HDST) [62] and slide-seq [50], incorporate additional spatial barcode in the sample preparation phase to trace back the original location of molecules. Then the expression and spatial information is obtained by complementary DNA (cDNA) sequencing. Though the two branches of technologies have different features, studies have shown that both are able to extract expression and spatial information of sufficient quality from varies types of tissues [68].

Traditional profiling data analysis mainly focuses on comparison of gene expression levels between groups of samples with different phenotypes, that is, identifying differently expressed (DE) genes. In the context of spatial profiling, however, a new question of interest is to study the association between gene expression levels and their spatial locations, i.e., identifying spatially expressed (SE) genes. Figure 3.1 illustrates an SE gene vs. a non-SE gene in spatially resolved expression profiles of a mouse olfactory bulb tissue [53], where a lighter color indicates higher expression. Edsgärd et al. [12] proposed Trendsceek, which detects SE genes by a permutation test. Specifically, each gene is tested independently, and four summary statistics (Stoyan's mark-correlation ρ, mean-mark function E, variance-mark V, and mark-variogram γ) are calculated to capture different aspects of the spatial distribution. Reference distributions of these statistics are then obtained from random permutations of gene expression levels at different locations. Finally, for each summary statistic, the p-value for detecting an SE gene can be computed as the probability of observing the statistic as extreme or more extreme than its observed value using the corresponding reference distribution. Note that the four statistics may have different power depending on the actual spatial distribution under consideration. SpatialDE [55] and SPARK [54] are both model-based methods that employ Gaussian processes to characterize spatial dependency. SpatialDE models the log transformed expression profiles for a given gene across spatial locations by a multivariate normal distribution, whose covariance matrix can be decomposed into a covariance matrix of a Gaussian process incorporating spatial correlations and an independent non-spatial variation in gene expression. Testing SE genes is equivalent to a likelihood ratio test with

an asymptotic χ^2 approximation, which compares the model with and without the spatial correlations. SPARK models the raw count data by Poisson distributions with mean rates across spatial locations depending on a Gaussian Process. Benchmark comparisons by Sun et al. [54] and Zhang et al. [68] suggest that SPARK gives the best performance in most situations, followed by SpatialDE. Trendsceek has less power in identifying SE genes, and also requires intensive computation.

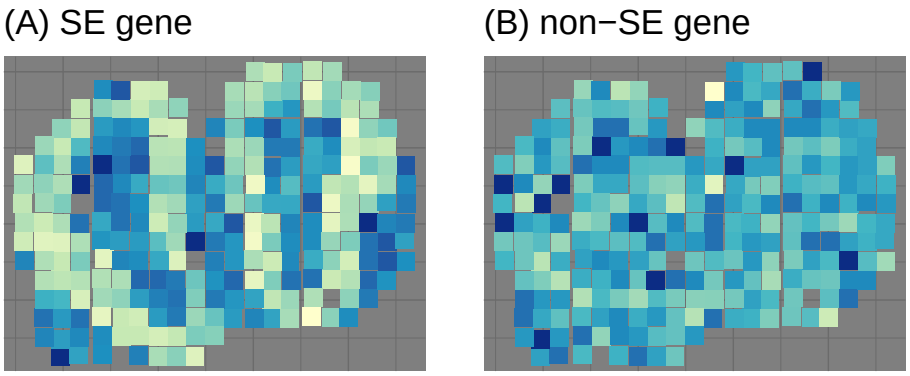


Figure 3.1: Spatially resolved expression profiles. (A) example of spatially expressed (SE) gene. (B) example of non-spatially expressed gene.

Gene expression data from spatial profiling are characterized by conspicuous zero inflation due to either technical issues, such as a limited sequencing depth or fluorescent level, or biological issues that some genes are indeed silent in a particular condition. For example, the mouse olfactory bulb data and breast cancer data [53–55] analyzed by both SpatialDE and SPARK contain about 60% and 90% zero counts, respectively. Nevertheless, none of the existing methods address the zero inflation issue formally. We propose a Bayesian **S**patial **HEAP**ing model (SHEAP), which aims to accurately recover major spatial patterns underlying the gene expression levels that are partially observed and subject to heaping at zero.

Heaping is a special case of coarse data [24] and arises naturally in many applications, especially those with self-reported data. For example, if age is recorded in months, people may report ages rounded to the nearest year (full-year heaper) or half-year (half-year

heaper) or the nearest month (exact reporter), and a reported age of 24 months can be due to full-year heaping, half-year heaping as well as an exact report. Heitjan and Rubin [23] modeled these heaping behaviors and studied the effect of age heaping on inference about the nutritional status of Tanzanian children. Other authors such as Pickering [43] and Wright and Bray [66] also modeled data with certain heaping mechanisms to improve statistical inference in medical studies. In the context of spatial profiling, gene expression count data with excess zeros can be treated as a special case of heaped data. An observed zero count may arise from either a truly non-expressed gene or an expressed gene, but the experiment fails to detect its expression because of various technical issues. Further, a zero count, due to a limited sequencing depth or fluorescent level, is more likely to occur among lowly expressed genes.

Among methods developed for detecting SE genes, our proposed method makes the first attempt to accurately model the heaping mechanism underlying zero-inflated count data. SHEAP also extends the existing heaping models to account for spatially correlated count data, which also allows the probability of heaping to be spatially dependent. Further, a hurdle model is employed to incorporate model selection seamlessly into the Bayesian framework for detecting SE genes. Through elaborate Bayesian hierarchical modeling that accounts for heaping, spatial dependency, and SE testing under one nutshell, SHEAP is powerful in detecting SE genes and uncovering spatial patterns, as will be confirmed by our numerical results.

The paper is structured as follows. In Section 3.2, a heaping data model is proposed to model the observed count data and multi-layer Bayesian hierarchical structures are introduced to model the heaping mechanism and the latent spatial process; Bayesian computation and inference is further elaborated on the proposed hierarchical model. Section 3.3 evaluates the performance of the proposed method by simulation studies and two data applications in mouse olfactory bulb data and breast cancer data. Concluding comments and future research directions are briefly outlined in Section 3.4.

3.2. Methods

Let the integer-valued matrix $\boldsymbol{Y}$ denote the gene expression data measured on I locations and J genes from a spatially resolved transcriptomic study. Each element Y_{ij} denotes the raw read count of gene j from location i, for $i = 1, ..., I$ and $j = 1, ..., J$. Further, let $\boldsymbol{T}$ be an $I \times 2$ matrix with the i-th row (t_{i1}, t_{i2}) denoting the coordinates of location i in a two-dimensional Euclidean space.

3.2.1. A spatial heaping model for count data

Consider modeling one gene at a time and so we ignore the subscript of the gene. Let Y_i denote the observed gene expression at location i for $i = 1, ..., I$. Let Z_i be a latent variable denoting the underlying expression at location i, and $Z_i|\lambda_i \sim \text{Poi}(s_i\lambda_i)$ where s_i is the size factor of location i reflecting systematic biases that affect measured gene expression levels and λ_i denotes the normalized expression level from location i. To ensure identifiability of these two sets of parameters, we use a plug-in estimate for $\boldsymbol{s} = (s_1, ..., s_I)$ [54], where each s_i is estimated by the total sum of counts at location i and subject to the constraint $\prod_{i=1}^{I} s_i = 1$. Next, define a latent binary variable G_i to indicate whether Z_i can be directly observed or heaped on zero such that the observed expression Y_i is a function of Z_i and G_i, namely

$$Y_i = Y(Z_i, G_i) = \begin{cases} Z_i & \text{if } G_i = 1, \\ 0 & \text{if } G_i = 0. \end{cases}$$

An observed nonzero expression $Y_i = y$ indicates that the corresponding $(Z_i, G_i) = (y, 1)$ for $y \neq 0$. However, any observed zero expression may arise from more than one possible (z, g) pairs; that is, g can be either zero or one: if $g = 0$, then z can be any count from the Poisson distribution; if $g = 1$, then z must equal zero. For each possible y value, we denote the set of such pairs as $ZG(y) = \{(z, g) : y = Y(z, g)\}$ so that

$$p(Y_i = y|g_i, z_i) = \begin{cases} 1 & \text{if } (z_i, g_i) \in ZG(y), \\ 0 & \text{otherwise.} \end{cases}$$

Further, we assume that $G_i|Z_i \sim \text{Ber}(\phi_i)$, where $\text{logit}(\phi_i) = \beta_0 + \beta_1 Z_i$. This is because lowly expressed genes are more likely to have their expression heaped on 0 due to the low amounts of mRNA and insufficient mRNA capture, and can be approximately captured by the linear relationship in the logit scale.Thus, the marginal probability of $Y_i = y$, given parameters λ_i, β_0, and β_1, can be written into

$$p(Y_i = y|\lambda_i, \beta_0, \beta_1) = \sum_{ZG(y)} p(y|g_i, z_i)p(g_i|z_i, \beta_0, \beta_1)p(z_i|\lambda_i),$$

which, combined with a logistic regression model for G_i, can be further written as

$$\begin{aligned} &p(Y_i = y_i|\lambda_i, \beta_0, \beta_1) \\ &= I(y_i = 0) \sum_{z=0}^{\infty} p(g = 0|z, \beta_0, \beta_1)p(z|\lambda_i) + p(g = 1|z = y_i, \beta_0, \beta_1)p(z = y_i|\lambda_i) \\ &= I(y_i = 0) \sum_{z=0}^{\infty} \frac{1}{1 + \exp(\beta_0 + \beta_1 z)} p(z|\lambda_i) + \frac{\exp(\beta_0 + \beta_1 y_i)}{1 + \exp(\beta_0 + \beta_1 y_i)} p(z = y_i|\lambda_i), \end{aligned}$$

where $I(\cdot)$ is the indicator function.

Similarly as in [23, 64], the above model contains three main elements: the underlying expression levels $\boldsymbol{Z}$ at the sample locations and their distributions; the zero inflation mechanism $\boldsymbol{G}$ given $\boldsymbol{Z}$ via a classical logistic regression setup; and a mapping from $(\boldsymbol{Z}, \boldsymbol{G})$ to the observed heaped counts $\boldsymbol{Y}$, where $\boldsymbol{Z} \equiv (Z_i)_{i=1}^I$, and $\boldsymbol{G}$ and $\boldsymbol{Y}$ are defined similarly. Note that Heitjan and Rubin [23] assumed independence across sampling units given parameters. We generalize the heaping model by incorporating potential spatial dependence into both $\boldsymbol{Z}$ and $\boldsymbol{G}$ by modeling the (log) mean structure of $\boldsymbol{Z}$ via a Gaussian Process, namely

$$\log(\boldsymbol{\lambda}) \sim \text{MVN}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{K}(l)), \tag{3.1}$$

where $\boldsymbol{\lambda} \equiv (\lambda_i)_{i=1}^I$, and $\boldsymbol{\mu} = \mathbf{1} \cdot \mu$ (i.e., all locations have a common mean μ). The correlation matrix $\boldsymbol{K}(l)$ (we write $\boldsymbol{K}(l) = \boldsymbol{K}$ thereafter) incorporates potential spatial patterns through a kernel function such as the squared exponential kernel, the Cauchy kernel or the Ornstein-Uhlenbeck kernel, and l is the characteristic length parameter. For example, a squared exponential kernel, used in our numerical examples in Section 3.3, gives the (i, j) element $K_{ij} = \exp(-\frac{d_{ij}^2}{2l^2})$ for $i \neq j$, and $K_{ii} = 1$, where $d_{ij} = \sqrt{(t_{i1} - t_{j1})^2 + (t_{i2} - t_{j2})^2}$ is the Euclidean distance between locations i and j. We denote the above model by M_1. A reduced case of M_1 is the model assuming that $\boldsymbol{K}$ is an identity matrix (i.e., $l = 0$), denoted M_0, corresponding to a non-SE gene.

3.2.2. Prior specification

We specify the following prior distributions for the parameters μ and σ^2 introduced by the Gaussian Process (3.1),

$$\mu|\sigma^2 \sim \mathrm{N}(\mu_0, h\sigma^2), \quad \sigma^2 \sim \text{Inv-Gamma}(a_\sigma, b_\sigma),$$

where $h = 10$, $a_\sigma = 3$, $b_\sigma = 10$, and μ_0 is set to the mean of $\log(Y_i)$'s for $Y_i > 0$. Note that h is set to a number larger than 3 so that where the center μ_0 is located is not so important and thus we use a rough estimate here. We specify a bivariate normal prior for the logistic regression parameters $\boldsymbol{\beta} = (\beta_0, \beta_1)^T$,

$$\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{b}, \boldsymbol{B}),$$

where $\boldsymbol{b} = (0, 0)^T$and $\boldsymbol{B} = 100 \cdot \boldsymbol{I}$ indicate a diffuse and vague prior. We further use a hurdle model to specify the prior distribution for l, which introduces a binary latent variable

D for model selection between M_0 and M_1,

$$l|D \sim \begin{cases} I(l=0) & D=0, \\ \mathsf{U}[L_l, U_l] & D=1, \end{cases} \quad D \sim \mathsf{Bern}(0.5). \tag{3.2}$$

For the case of $D = 0$, $l = 0$ with probability 1 so that $\boldsymbol{K}$ reduces to an identity matrix, which corresponds to the null model M_0. The hurdle is cleared if $D = 1$ with a prior probability of 0.5. Therefore, (3.2) assumes that M_1 and M_0 are equally likely *a priori*. Note that the value of the lower bound L_l determines the minimum detectable spatial correlation. We set L_l to half of the 1-percentile of all pairwise Euclidean distances d_{ij}'s, which leads to the top 1% pair-wise correlation in $\boldsymbol{K}$ greater than $\exp(-2) \approx 0.135$. We also note that the inverse of $\boldsymbol{K}$ becomes numerically unstable as l increases, due to the singularity of the resulting covariance matrix. Thus, we set U_l to the median of the pairwise Euclidean distances in our algorithm implementation.

3.2.3. Posterior computation and inference

Figure 3.2 shows the hierarchical structure of the proposed SHEAP model via a diagram. The conjugate N-Inv-Gamma prior setting for μ and σ^2 that takes the form of $p(\mu, \sigma^2) = p(\mu|\sigma^2)p(\sigma^2)$ allows us to integrate the Gaussian Process mean and the covariance scaling factor out of $p(\log \boldsymbol{\lambda}|\mu, \sigma^2, l)$, yielding a generalized multivariate Student's t-distribution on $\log \boldsymbol{\lambda}$:

$$p(\log \boldsymbol{\lambda}|l) \propto |\boldsymbol{K}|^{-\frac{1}{2}} H^{-\frac{1}{2}} \left\{ b_\sigma + \frac{1}{2}\left[\log \boldsymbol{\lambda}^T \boldsymbol{K}^{-1} \log \boldsymbol{\lambda} - \frac{1}{H}(\mathbf{1}^T \boldsymbol{K}^{-1} \log \boldsymbol{\lambda} - \frac{\mu_0}{h})^T (\mathbf{1}^T \boldsymbol{K}^{-1} \log \boldsymbol{\lambda} - \frac{\mu_0}{h}) \right] \right\}^{-a_\sigma - \frac{I}{2}},$$

where $H = K_{\cdot\cdot}^{-1} + \frac{1}{h}$, $K_{\cdot\cdot}^{-1}$ is the grand sum of the matrix $\boldsymbol{K}^{-1}$; $\mathbf{1}$ is an $I \times 1$ vector of 1; μ_0, h, a_σ and b_σ are pre-specified hyper-parameters as mentioned in Section 3.2.2.

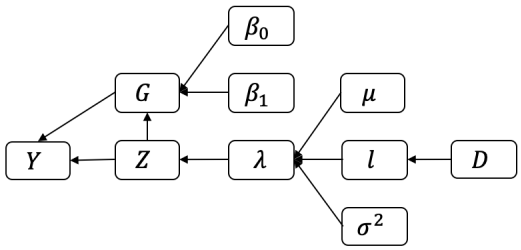


Figure 3.2: The hierarchical structure of the SHEAP model.

Let $\boldsymbol{\Theta} = (l, \beta_0, \beta_1)$. The full probability model, after marginalization over μ and σ^2, is given by

$$\begin{aligned} p(\boldsymbol{Y}, \boldsymbol{G}, \boldsymbol{Z}, \log \boldsymbol{\lambda}, \boldsymbol{\Theta}) &= p(\boldsymbol{Y}|\boldsymbol{G}, \boldsymbol{Z}, \boldsymbol{\Theta}) p(\boldsymbol{G}|\boldsymbol{Z}, \boldsymbol{\Theta}) p(\boldsymbol{Z}|\boldsymbol{\lambda}, \boldsymbol{\Theta}) p(\log \boldsymbol{\lambda}|\boldsymbol{\Theta}) p(\boldsymbol{\Theta}|D) p(D) \\ &= \prod_{i=1}^{I} \left\{ I[(z_i, g_i) \in ZG(y_i)] \frac{[\exp(\beta_0 + \beta_1 z_i)]^{g_i}}{1 + \exp(\beta_0 + \beta_1 z_i)} \frac{(s_i \lambda_i)^{z_i}}{z_i!} \exp(-s_i \lambda_i) \right\} \\ & \quad p(\log \boldsymbol{\lambda}|l) p(\boldsymbol{\Theta}|D) p(D). \end{aligned}$$

To employ a Gibbs sampler for posterior computation using Markov chain Monte Carlo (MCMC), we first derive the complete set of full conditionals. First note that

$$\begin{cases} p(G_i = 1|Y_i > 0) = 1, \\ p(G_i = 0|Y_i = 0, Z_i > 0) = 1, \\ p(G_i = 0|Y_i = 0, Z_i = 0, \beta_0) = \frac{1}{1+e^{\beta_0}}, \\ p(G_i = 1|Y_i = 0, Z_i = 0, \beta_0) = \frac{e^{\beta_0}}{1+e^{\beta_0}}. \end{cases}$$

Thus, in each MCMC iteration, we set $G_i \equiv 1$ if $Y_i > 0$, and set $G_i = 0$ if $Y_i = 0$ and $Z_i > 0$; we update each G_i sequentially only if $Y_i = Z_i = 0$ by sampling from a Bernoulli(ϕ) distribution with $\phi = \frac{1}{1+e^{\beta_0}}$, where Z_i and β_0 take values from the most recent updates.

The full conditionals for $\boldsymbol{\lambda}$ and $\boldsymbol{Z}$ are given by

$$\begin{cases} p(Z_i = Y_i | G_i = 1) = 1, \\ p(Z_i = z | G_i = 0, \lambda_i, \beta_0, \beta_1) \propto \frac{(s_i\lambda_i)^z}{(1+e^{\beta_0+\beta_1 z})z!} \quad \text{for } z \geq 0, \end{cases} \tag{3.3}$$

$$p(\log \lambda_i | Z_i, l, \log \boldsymbol{\lambda}_{-i}) \propto (s_i\lambda_i)^{Z_i} e^{-s_i\lambda_i} \left\{ b_\sigma + \frac{1}{2} \left[\log \boldsymbol{\lambda}^T \left(\boldsymbol{K}^{-1} - H\boldsymbol{K}^{-1}\boldsymbol{J}\boldsymbol{K}^{-1} \right) \log \boldsymbol{\lambda}^T \right] \right\}^{-a_\sigma - \frac{I}{2}},$$

where $\boldsymbol{\lambda}_{-i} = (\lambda_1, ..., \lambda_{i-1}, \lambda_{i+1}, ..., \lambda_I)$. Since these are not known distributions and cannot be sampled directly, we use the Metropolis-Hastings algorithm to sample $\boldsymbol{\lambda}$ and $\boldsymbol{Z}$ [22]. See Appendix C.1 for the detail.

The Polya-Gamma method [44] is employed to sample from the full conditional distribution of $\boldsymbol{\beta}$ efficiently. We first sample a latent variable conditional on $\boldsymbol{\beta}$ and $\boldsymbol{Z}$ by $\omega_i | \boldsymbol{\beta}, Z_i \sim \text{Polya-Gamma}(1, \beta_0 + \beta_1 Z_i)$. Then sample the full conditional of $\boldsymbol{\beta}$ by

$$\boldsymbol{\beta} | \boldsymbol{Z}, \boldsymbol{G}, \boldsymbol{\omega} \sim \text{N}(m_\omega, V_\omega),$$

where $V_\omega = (\boldsymbol{Z}_1^T \boldsymbol{\Omega} \boldsymbol{Z}_1 + \boldsymbol{B}^{-1})^{-1}$ and $m_\omega = V_\omega(\boldsymbol{Z}_1^T \kappa + \boldsymbol{B}^{-1} b)$, where $\boldsymbol{Z}_1$ is an $I \times 2$ matrix with its first column being 1's and second column being $Z_i's$, $\kappa = (g_1 - \frac{1}{2}, \ldots, g_I - \frac{1}{2})$, and $\boldsymbol{\Omega}$ is the diagonal matrix of ω_i's.

Since l depends on D, we perform a joint update of the two parameters within each Gibbs sampling iteration using another M-H step. To do so, we first propose $D^* = 1 - D$. If $D^* = 1$, we should propose l^* from $U[L_l, U_l]$. Otherwise, we set $l^* = 0$. The proposed move is accepted with probability $\min(1, m_D)$, where the hastings ratio m_D is

$$m_D = \frac{p(\log \boldsymbol{\lambda} | l^*) p(D^*) p(l^* | D^*) J(D | D^*) J(l | l^*)}{p(\log \boldsymbol{\lambda} | l) p(D) p(l | D) J(D^* | D) J(l^* | l)},$$

where $\frac{p(D^*)}{p(D)} = \frac{J(D|D^*)}{J(D^*|D)} = 1$ and $\frac{p(l^*|D^*)J(l|l^*)}{p(l|D)J(l^*|l)} = 1$. To further speed up the MCMC algorithm, we replace the continuous uniform component in the prior distribution of l by a discrete uniform distribution on T values (say $T = 10$). Specifically, we choose T equally-spaced points from $\log L_l$ to $\log U_l$. Then transform the points to the original scale by taking anti-log to form the discrete uniform distribution.

Finally, to detect spatially expressed (SE) genes, we calculate p_0 as the proportion of $l = 0$ from its posterior samples after a burn-in period. A gene is identified as an SE gene at the significance level of α if $p_0 < \alpha$.

3.3. Results

3.3.1. Simulation

We conduct simulation studies to evaluate and compare the performance of the proposed SHEAP method and existing methods including SPARK, SpatialDE, and Trendsceek. The spatial heaping model specified in Section 3.2.1 is considered as the data generating model in our simulation study I, and we control the zero-inflation proportion by varying the value of β_0. Specifically, we generate 100 random location coordinates by a two-dimensional uniform distribution on $[5, 25] \times [5, 25]$, which is approximately the range of coordinates from a published spatial profiling data [53]. We set $\mu = 2.5$, $\beta_1 = 0.12$ and $\beta_0 = -1.5, -2.5, -3.5$ to reflect low, median and high levels of zero inflation. The 3 settings of β_0 correspond to approximately $45\%, 60\%$ and 75% of zeros within each gene. We also consider a no-extra zero setting that corresponds to a data generating model with $G \equiv 1$ (i.e., heaping on zero does not occur). The size factor $\boldsymbol{s} = (s_1, ..., s_I)$ is generated from a N$(1, 0.2)$ distribution on each location independently. For SE genes, we set the scaling parameter $\sigma^2 = 1$ in the Gaussian Process covariance matrix and the characteristic length parameter l to the 5-percentile of all pair-wise Euclidean distances. For genes with

no spatial pattern, we simply set $\sigma^2 \boldsymbol{K}$ to an identity matrix. Ten SE genes and ten non-SE genes are generated in each of the four settings for different levels of zero inflation.

To detect SE genes, we first obtain the p-values from each method. Then we control for false discovery rate (FDR) at the level of 0.05 using the Benjamini-Hochberg (BH) procedure [3]. The results are shown in Table 3.1. All methods are able to control the false discovery rate (FDR) well in the four settings considered. As mentioned in the introduction, Trendsceek has four different versions, each using a different summary statistic, including Stoyan's mark-correlation ρ, mean-mark function E, variance-mark V, and mark-variogram γ, Among them, the Stoyan's mark-correlation ρ appears to be the best. Nevertheless, all the four Trendsceek variants perform poorly, yielding no/low power, regardless of the level of zero inflation. The other methods perform well in the no-extra zero setting: the proposed SHEAP and SpatialDE detect all the SE genes,and SPARK detects 9 out of the 10 SE genes. As the degree of zero inflation increases from low to high, SHEAP retains great performance by identifying all the SE genes meanwhile maintaining the zero FDR. SpatialDE has an increasing power as the proportion of extra zeros increases, and so it is better than SPARK whose power is steadily low.

Figure 3.3 shows receiver operating characteristic (ROC) curves based on false discovery rates under each of the four settings. In the high proportion of extra zeros setting, all methods tend to have a high area under the ROC curve (AUC). We note that for SE genes, the heaping phenomenon amplifies the spatial correlations among lowly expressed locations, since heaped zero expressions help make (nearly) perfect correlations. In addition, the zero inflation causes over-dispersion on the count data. SpatialDE models the log-transformed expression directly by a Gaussian Process, which has an additional parameter to model the variance. However, SPARK models the count data by a Poisson distribution, which is known to be inadequate for over-dispersed data. Overall, SHEAP is the only method that models the zero inflation rigorously and thus has consistently good performance regardless of the zero-inflation level.

Extra zeros	Zero			Low			Median			High		
	TP	FP	AUC	TP	FP	AUC	TP	FP	AUC	TP	FP	AUC
SHEAP	10	0	1	10	0	1	10	0	1	10	0	1
SPARK	9	0	1	3	0	0.98	4	0	0.95	3	0	1
SpatialDE	10	0	1	3	0	0.96	7	0	0.935	9	0	0.99
Trendsceek.E	0	0	0.75	0	0	0.34	0	0	0.435	0	0	0.67
Trendsceek.ρ	3	1	0.83	0	0	0.77	3	1	0.81	3	1	0.84
Trendsceek.γ	0	0	1	0	0	0.91	0	0	0.80	4	0	1
Trendsceek.V	0	0	0.67	0	0	0.70	0	0	0.615	0	0	0.70

Table 3.1: True positive (TP), false positive (FP) and area under the ROC curve (AUC) results for Simulation I. Ten spatially expressed genes and ten non-spatially expressed genes are generated in each setting. Zero, low, median and high proportion of extra zeros correspond to approximately 45%, 60% and 75% of zeros within each gene. TP (FP) is the number of correctly (falsely) identified SE genes.

In Simulation II, we exam the performance of the methods under different levels of spatial correlations. We set l to the 2.5- and 10-percentile of all pair-wise Euclidean distances to reflect low and high levels of spatial correlations. All other parameters are set to the same values as in simulation I, and we fix β_0 at -2.5, which produces approximately 60% zeros within each gene to mimic the real scenario [53]. Again, ten SE genes and ten non-SE genes are generated in each of the two settings.

Table 3.2 summarizes the results from simulation II. SHEAP is able to identify almost all SE genes and makes no false positives in both settings. The second best method, SpatialDE, detects only one SE gene in the low spatial correlation setting and seven in the high spatial correlation setting. SPARK and Trendsceek.ρ show no power in detecting SE genes in the low correlation setting, and detect about half of the SE genes in the high correlation setting. Trendsceek.E, Trendsceek.γ and Trendsceek.V have no power in either setting. The ROC curves based on false discovery rates are reported in Figure 3.4. Clearly, the proposed SHEAP offers the best performance with the AUC equal to one in both settings.

Spatial correlation	Low			High		
	TP	FP	AUC	TP	FP	AUC
SHEAP	9	0	1	10	0	1
SPARK	0	0	0.3	5	0	0.95
SpatialDE	1	0	0.79	7	0	0.96
Trendsceek.E	0	0	0.63	0	0	0.53
Trendsceek.ρ	0	0	0.68	4	1	0.74
Trendsceek.γ	0	0	0.46	0	0	0.82
Trendsceek.V	0	0	0.57	0	0	0.51

Table 3.2: True positive (TP) and false positive (FP) results for Simulation II. Ten spatially expressed genes and ten non-spatially expressed genes are generated in each setting. l is set to 2.5- and 10-percentile of all pair-wise Euclidean distances in the low and high spatial correlation setting, respectively.

3.3.2. Data application

We apply the proposed SHEAP method to analyze two published datasets and compare its performance with the existing methods SpatialDE, SPARK, and Trendscreek.

Mouse olfactory bulb data. The mouse olfactory bulb data [53] contains $11,274$ genes measured on 260 locations by spatial transcriptomics sequencing. In the original study, Ståhl et al. [53] presented a list of 10 marker genes in the olfactory system. These genes are known to have enriched expression in the mitral cell layer (MCL) but low expression or even no expression in the adjacent granular cell layer (GCL), and so they should exhibit spatial patterns. The proposed SHEAP detects 9 of the 10 marker genes as SE genes (Figure 3.5) and SPARK detects 8, SpatialDE detects 3 and all the four statistic tests of Trendsceek detect none of them. We summarize the 95% credible interval of l in Table 3.3.

It is worth noting that SPARK performs almost as good as the proposed SHEAP. We further examined the data and found that these 10 genes have only 3.4% zeros on average, while the whole mouse olfactory bulb dataset contains 59.6% zeros.

	Doc2g	Uchl1	Nmb	Plcxd2	Shisa3	Sv2b	Cdhr1	Reln	Slc17a7	Rcan2
LL	0.82	0	0.51	0.65	0.65	0.51	0.65	0.65	0.65	0.51
UL	0.82	0.51	0.82	0.82	0.82	0.65	0.65	0.65	0.65	0.65
p_0	0	0.71	0	0	0	0	0	0	0	0

Table 3.3: Mouse olfactory bulb data. Analysis of the 10 marker genes in mouse olfactory bulb data. LL and UL are the lower and upper limits of the 95% credible interval of l. p_0 is the proportion of $l = 0$ from its posterior samples after burn-in.

Breast cancer data. We consider a human breast cancer dataset [53] in our second data application. The data consist of spatial transcriptomics of human breast cancer biopsies measured on $5,262$ genes and 250 locations. Ståhl et al. [53] identified, on the basis of morphological criteria, one area with invasive ductal cancer and six separate areas of ductal cancer in situ. In the original study, 14 extracellular matrix-associated genes showed high expression in the invasive ductal cancer area. In addition, their expression levels in the ductal cancer in situ areas suggested a high degree of heterogeneity. We summarize the results of extracellular matrix-associated genes in Table 3.4. SHEAP identifies all the 14 extracellular matrix-associated genes as spatially expressed genes (Figure 3.6), whereas SPARK detects only 10, SpatialDE detects 7, and Trendsceek detects only 2 (false discovery rate < 0.05 for at least one of the four statistic tests). Further, seven of the extracellular matrix-associated genes are highly zero-inflated with the zero proportion greater than 67%. SHEAP and SpatialDE are able to detect all of the 7 genes while SPARK detects 3. For example, GAS6, which is related to the epithelial-to-mesenchymal transition [17, 18], was detected by SHEAP and SpatialDE only. The expression of GAS6 is high only in two of the ductal cancer in situ area, which may reflect different subclones.

3.4. Discussion

We have developed a new Bayesian method, SHEAP, for identifying SE genes in spatial transcriptomics studies. SHEAP is based on the assumption that the observed ex-

	POSTN	MMP14	SPARC	IGFBP5	DCN	SCGB2A2	FN1
LL	0.81	0.51	0.64	0.81	0.64	0.51	0.81
UL	0.81	0.64	0.81	0.81	0.81	0.81	0.81
p_0	0	0.0005	0	0	0	0.0005	0
	VIM	GAS6	KRT17	AREG	PEG10	MUCL1	PIP
LL	0.64	0.64	0.64	0.81	0.51	0.51	0.51
UL	0.81	0.81	0.81	0.81	0.81	0.81	0.81
p_0	0	0	0	0	0	0.0005	0.0075

Table 3.4: Breast cancer data. Analysis of the 14 extracellular matrix-associated genes. LL and UL are the lower and upper limit of the 95% credible interval of l. p_0 is the proportion of $l = 0$ from its posterior samples after burn-in.

pression level is a function of an underlying expression level, which incorporates potential spatial patterns and is subject to heaping at zero. By modeling the heaping behavior, whose distribution depends on the underlying expression level, we rigorously model the zero inflation issue that is often observed in spatial transcriptomics data. An efficient MCMC algorithm is designed for valid bayesian inference. Simulation studies and real data applications show that the proposed SHEAP is more powerful than the existing competitors, especially in situations of a high proportion of excess zeros.

A common approach to characterize the excess zeros in sequencing data is using a zero-inflated distribution [8, 69]. A zero-inflated distribution accounts for the proportion of extra zeros by a single parameter, and the presence of an extra zero is random in the sense that it is not related to the potentially unobserved expression count. In practice, however, the zero count, due to a limited sequencing depth, is more likely to present among lowly expressed genes. To the best of our knowledge, the proposed SHEAP is the only method that accounts for this dependence between the underlying count and the probability of observing an extra zero.

In our numerical experiments and data applications, we restrict our attention to the squared exponential covariance kernel when modeling the spatial dependency. However,

our method can be directly applied to other popular kernel functions such as the Cauchy kernel and the Ornstein-Uhlenbeck kernel. Our Bayesian framework has the capacity of further incorporating kernel selection and model averaging into the proposed SHEAP method, which is an interesting future research direction.

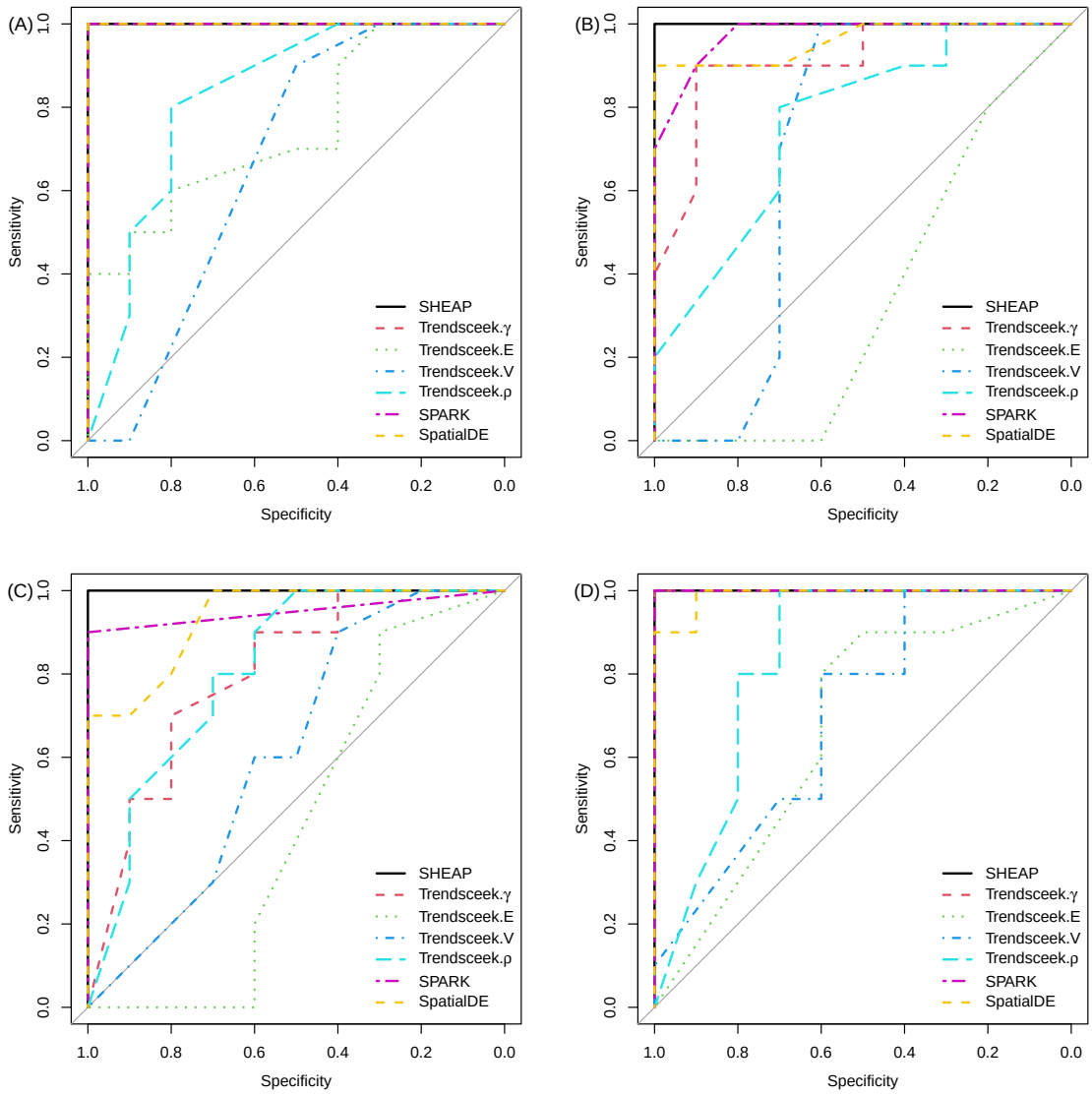


Figure 3.3: ROC curves at the false discovery rates from 0 to 1 for Simulation I. (A) no extra zeros. (B) low proportion of extra zeros. (C) median proportion of extra zeros. (D) high proportion of extra zeros.

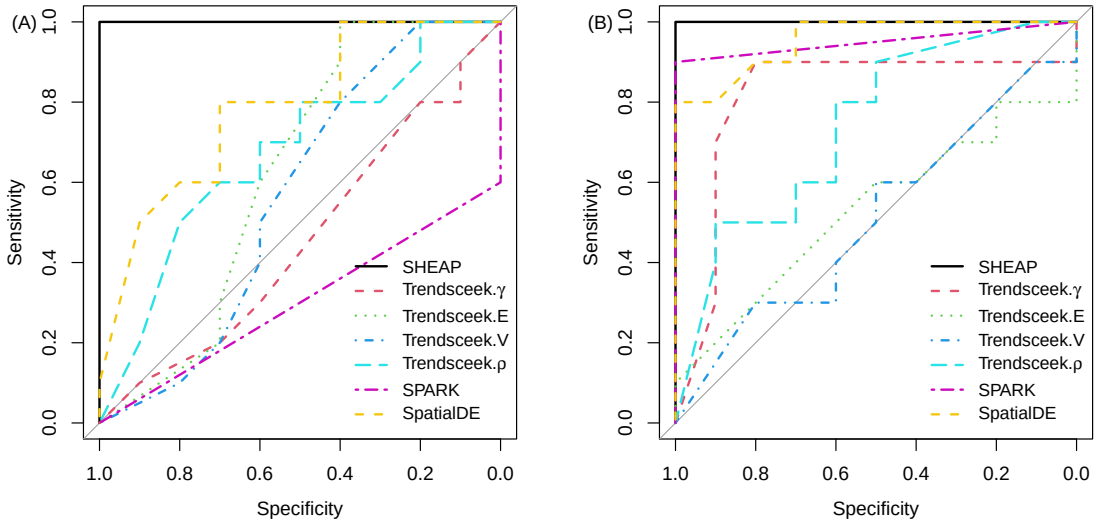


Figure 3.4: ROC curves at the false discovery rates from 0 to 1 for Simulation II. (A) low spatial correlations. (B) high spatial correlations.

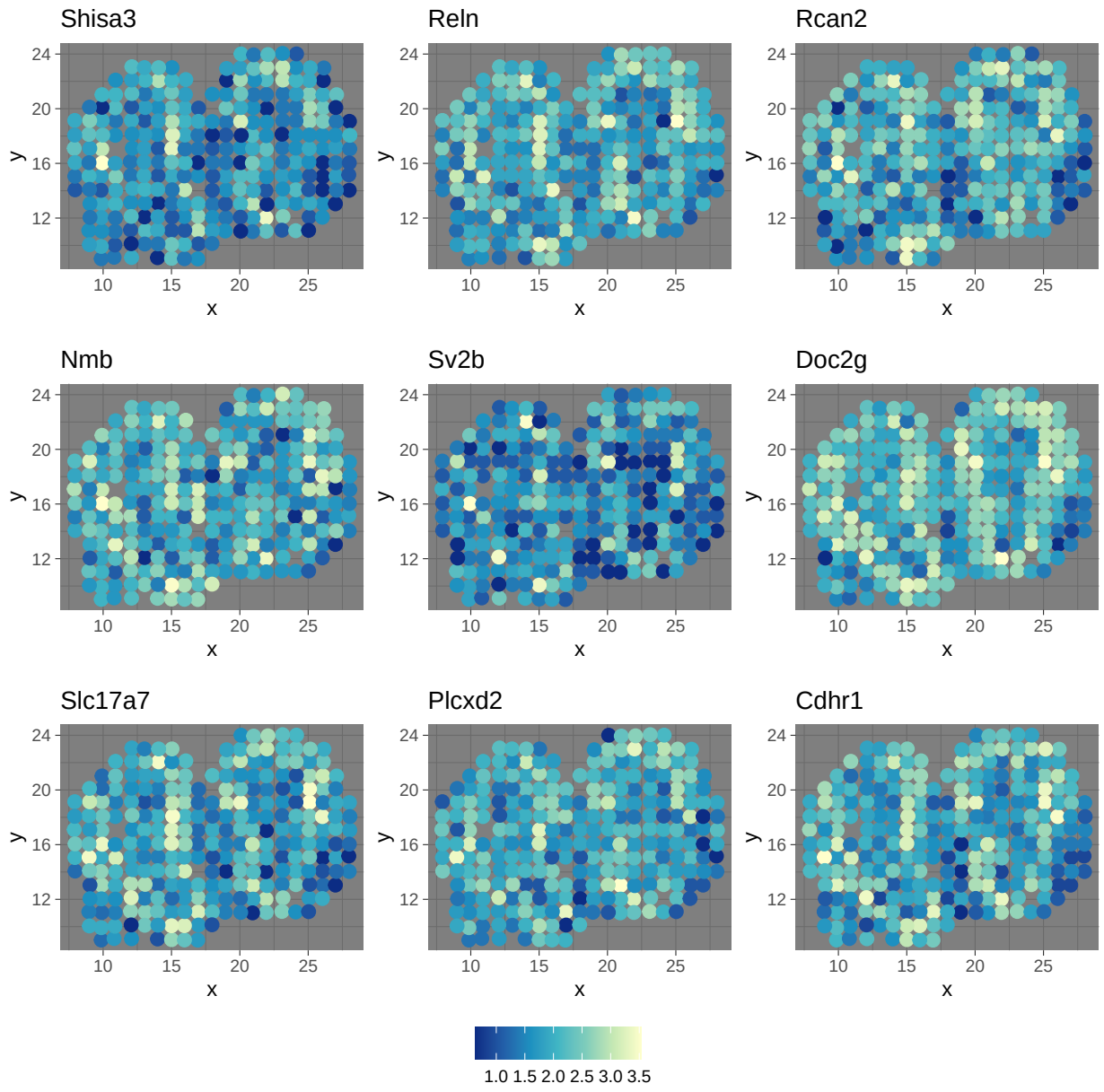


Figure 3.5: Mouse olfactory bulb data. Spatial expression patterns for 9 SE genes identified by SHEAP. Expression levels are in the natural log scale. These genes are known to have enriched expression in the mitral cell layer (MCL) but low expression or even no expression in the adjacent granular cell layer (GCL).

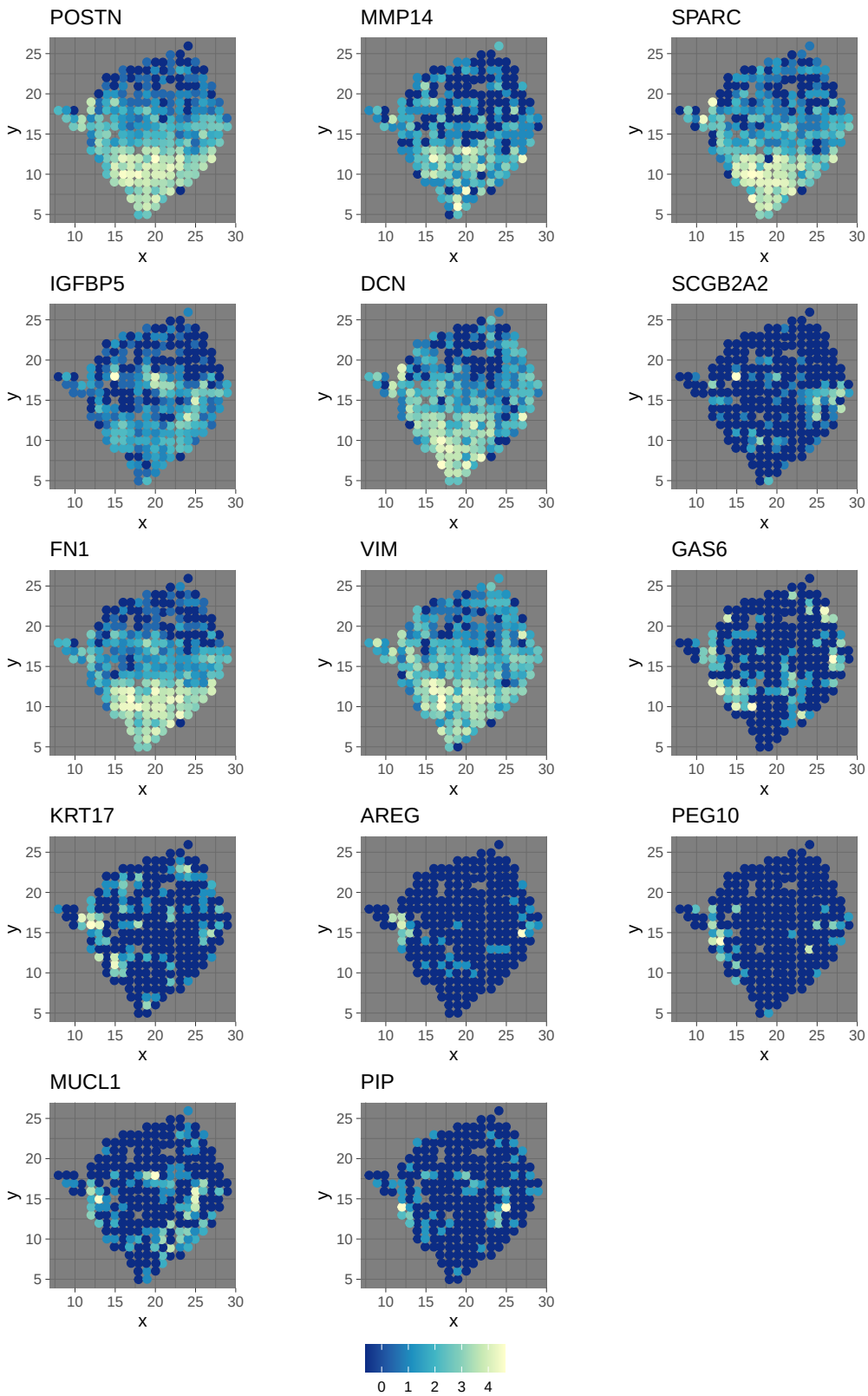


Figure 3.6: Breast cancer data. Spatial expression patterns for 14 SE genes identified by SHEAP. Expression levels are in the natural log scale. These 14 extracellular matrix-associated genes are cancer relevant and characterized as spatially expressed genes in the original study [53].

APPENDIX A

APPENDIX of CHAPTER 1

A.1. Approximating a discrete distribution by a continuous distribution

For a specific gene and sample, let C denote the raw count and $L \equiv \log(C+1)$ is the natural logarithm transformed count. Here, we drop the subscripts i and j for simplicity. We note that L is a discrete random variable with support $\{0, \log(1), \log(2), ...\}$, but in Section 3 of the main paper, a continuous distribution (i.e., a truncated normal distribution) is used to approximate the discrete distribution of L for expressed genes (when $D = 1$). Below is a justification for the validity of this approximation.

Let Θ denote the set of all parameters, we have

$$p(C|\Theta) = p(C|D=1,\Theta)p(D=1|\Theta) + p(C|D=0,\Theta)p(D=0|\Theta),$$

where $p(C|D=0,\Theta)$ is the probability mass function (PMF) of the ZIP distribution and $p(C|D=1,\Theta)$ is the PMF of some discrete distribution, both with support $\{0,1,2,...\}$.

Suppose a random variable $X \sim \text{TN}(\mu, \sigma^2, 0, \infty)$ is used to approximate L and f_X is the probability density function (PDF) of X. Let $Y = g(X) \equiv \exp(X) - 1$ so that $X = g^{-1}(Y) = \log(Y+1)$. According to the change-of-variable theorem (i.e., Theorem 2.1.5 in Casella and Berger [5]), Y has a PDF given by $f_Y = f_X \cdot (g^{-1})' = f_X \cdot \frac{1}{y+1}$. Clearly, Y is a continuous random variable on $[0, \infty)$. We now use Y to approximate C, namely

$$
\begin{aligned}
P(C = c) &\approx P(c - 0.5 \leq Y < c + 0.5) \\
&= \int_{c-0.5}^{c+0.5} f_Y dy \\
&\approx \frac{1}{2} \left[f_Y(c - 0.5) + f_Y(c + 0.5) \right] \left[c + 0.5 - (c - 0.5) \right] \\
&\approx f_Y(c),
\end{aligned}
$$

where the third line uses the area of a trapezoid to approximate the integral. Therefore, the PMF of the discrete random variable C can be approximated by the PDF of Y directly. The above approximation works well for expressed genes, whose observed counts are not close to zero.

A.2. Technical Details of the Nested EM Algorithm

Let t denote the current iteration of the outer EM and k denote the current cycle of the inner EM; let K denote the total number of cycles in the inner EM; let Θ denote the set of all parameters. Given $\sum_{j=1}^{J} D_j$ positive observations, the number of negative observations from $\mathsf{N}(\mu_i, \sigma_i^2)$, denoted by T_i, is modeled by a negative binomial (NB) distribution,

$$
T_i \mid \boldsymbol{D}, \Theta^{(t+\frac{k-1}{K})} \sim \mathrm{NB}\left(\sum_{j=1}^{J} \boldsymbol{D}_j, \Phi\left(-\frac{\mu_i^{(t+\frac{k-1}{K})}}{\sigma_i^{(t+\frac{k-1}{K})}} \right) \right),
$$

where $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution $\mathsf{N}(0, 1)$. For $j = J + 1, ..., J + T_i$,

$$
L_{ij} \mid \boldsymbol{Y}_{\mathrm{obs}}, \Theta^{(t+\frac{k-1}{K})} \sim \mathrm{TN}\Big(\mu_i^{(t+\frac{k-1}{K})}, \sigma_i^{(t+\frac{k-1}{K})}, -\infty, 0 \Big).
$$

Let $t_k = t + \frac{k-1}{K}$. Then the conditional expectations of the sufficient statistics are

$$s_i^{(t_k)} = \mathbf{E}\left(s_i \mid \boldsymbol{C}, \boldsymbol{w}^{(t)}, \Theta^{(t_k)}\right) = \sum_{j=1}^{J} w_j^{(t)} L_{ij} + T_i^{(t_k)} m_i^{(t_k)},$$

$$S_i^{(t_k)} = \mathbf{E}\left(S_i \mid \boldsymbol{C}, \boldsymbol{w}^{(t)}, \Theta^{(t_k)}\right) = \sum_{j=1}^{J} w_j^{(t)} L_{ij}^2 + T_i^{(t_k)} M_i^{(t_k)},$$

where

$$T_i^{(t_k)} = \mathbf{E}\left(T_i \mid \boldsymbol{w}^{(t)}, \Theta^{(t_k)}\right) = \frac{\sum_{j=1}^{J} w_j^{(t)} \Phi\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)}{1 - \Phi\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)},$$

$$m_i^{(t_k)} = \mathbf{E}\left(L_{ij} \mid \boldsymbol{Y}_{\text{obs}}, \Theta^{(t_k)}\right) = \mu_i^{(t_k)} - \frac{\psi\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)}{\Phi\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)} \sigma_i^{(t_k)}, \tag{A.1}$$

$$M_i^{(t_k)} = \mathbf{E}\left(L_{ij}^2 \mid \boldsymbol{Y}_{\text{obs}}, \Theta^{(t_k)}\right)$$

$$= \left(\sigma_i^{(t_k)}\right)^2 + \left(\mu_i^{(t_k)}\right)^2 + \left(\sigma_i^{(t_k)}\right)^2 \frac{\psi'\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)}{\Phi\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)} - 2\mu_i^{(t_k)} \sigma_i^{(t_k)} \frac{\psi\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)}{\Phi\left(-\frac{\mu_i^{(t_k)}}{\sigma_i^{(t_k)}}\right)}, \tag{A.2}$$

and $\psi(\cdot)$ is the pdf of a standard normal distribution and ψ' denote its first derivative. (A.1) and (A.2) are the first and second moment of the truncated normal distribution. The tth iteration of our nested EM algorithm takes the form:

E1-step: Compute $w_j^{(t)} = \mathbf{E}\left(D_j|\boldsymbol{C}_j, \Theta^{(t)}\right)$ as the E step in an ordinary EM algorithm.

E2-step:

$$\begin{aligned} Z_{ij}^{\left(t+\frac{k-1}{K}\right)} &= \mathbf{E}\left(Z_{ij}|C_{ij}, w_j^{(t)}, \Theta^{\left(t+\frac{k-1}{K}\right)}\right) = \Pr\left(Z_{ij}=1|C_{ij}, w_j^{(t)}, \Theta^{\left(t+\frac{k-1}{K}\right)}\right) \\ &= \frac{f\left(Z_{ij}=1, C_{ij}, w_j^{(t)}|\Theta^{\left(t+\frac{k-1}{K}\right)}\right)}{f\left(Z_{ij}=1, C_{ij}, w_j^{(t)}|\Theta^{\left(t+\frac{k-1}{K}\right)}\right) + f\left(Z_{ij}=0, C_{ij}, w_j^{(t)}|\Theta^{\left(t+\frac{k-1}{K}\right)}\right)} \\ &= \begin{cases} \dfrac{\left(\pi_j^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t)}}}{\left(\pi_j^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t)}} + \left(1-\pi_j^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t)}} \exp\left(-\delta_i^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t)}}} & \text{if } C_{ij}=0; \\ 0 & \text{if } C_{ij}\neq 0, \end{cases} \end{aligned}$$

$$s_i^{(t+\frac{k-1}{K})} = \mathbf{E}\left(s_i|\boldsymbol{C}, \boldsymbol{w}^{(t)}, \Theta^{(t+\frac{k-1}{K})}\right) = \sum_{j=1}^{J} w_j^{(t)} L_{ij} + T_i^{(t+\frac{k-1}{K})} m_i^{(t+\frac{k-1}{K})},$$

$$S_i^{(t+\frac{k-1}{K})} = \mathbf{E}\left(S_i|\boldsymbol{C}, \boldsymbol{w}^{(t)}, \Theta^{(t+\frac{k-1}{K})}\right) = \sum_{j=1}^{J} w_j^{(t)} L_{ij}^2 + T_i^{(t+\frac{k-1}{K})} M_i^{(t+\frac{k-1}{K})}.$$

M1-step: $\phi^{(t+1)} = \sum_{j=1}^{J} w_j^{(t)}/J$**.**

M2-step:

$$\pi_j^{\left(t+\frac{k}{K}\right)} = \frac{\sum_{i=1}^{I} Z_{ij}^{\left(t+\frac{k-1}{K}\right)}}{I}, \quad \delta_i^{\left(t+\frac{k}{K}\right)} = \frac{\sum_{j=1}^{J}\left(1-w_j^{(t)}\right)\left(1-Z_{ij}^{\left(t+\frac{k-1}{K}\right)}\right)C_{ij}}{\sum_{j=1}^{J}\left(1-w_j^{(t)}\right)\left(1-Z_{ij}^{\left(t+\frac{k-1}{K}\right)}\right)},$$

$$\mu_i^{(t+\frac{k}{K})} = \frac{s_i^{(t+\frac{k-1}{K})}}{\sum_{j=1}^{J} w_j^{(t)} + T_i^{(t+\frac{k-1}{K})}}, \quad \sigma_i^{(t+\frac{k}{K})} = \sqrt{\frac{S_i^{(t+\frac{k-1}{K})}}{\sum_{j=1}^{J} w_j^{(t)} + T_i^{(t+\frac{k-1}{K})}} - \left(\mu_i^{(t+\frac{k}{K})}\right)^2}.$$

For $k = 1, \ldots, K$, the tth iteration repeats E2-step and M2-step K times. Upon completion of the K cycles, we set $\left(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\delta}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\sigma}^{(t+1)}\right) = \left(\boldsymbol{\pi}^{(t+\frac{K}{K})}, \boldsymbol{\delta}^{(t+\frac{K}{K})}, \boldsymbol{\mu}^{(t+\frac{K}{K})}, \boldsymbol{\sigma}^{(t+\frac{K}{K})}\right)$.

A.3. Performance Evaluation via Simulation

A.3.1. Settings

Our mixture model assumes a common sample-specific effect for all expressed genes, represented by the mean of the truncated normal distribution for a given sample. In reality, the RNA degradation rates are not the same among genes in a given sample due to a variety of factors. For example, Chen et al. [6] studied genome-wide mRNA degradation and showed that RNA degradation is dependent on the distance relative to the mRNA's 5' end. Environmental conditions also have effects on gene-wise degradation rates. However, without using covariate information about such factors that may be hard to obtain, none of the six normalization methods can separate the gene-wise degradation effects from the true expression levels. To avoid overly optimistic results, we considered the gene-wise degradation effects in our simulation by generating the gene-wise noise g_j from N$(0, \sigma_g^2)$ with $\sigma_g = 1.5$ for all expressed genes. Specifically, the following data generating model was used: $C_{ij} \sim \mathrm{ZIP}(\pi_j, \delta_i)$ for $j > \phi J$ and $L_{ij} \sim \mathrm{TN}(\mu_i + g_j, \sigma_i^2, 0, +\infty)$ for $j \leq \phi J$, where ϕ indicates the proportion of expressed genes, and ϕJ is the number of expressed genes rounded to the nearest integer. We then generated data under various scenarios using the above model based on a real dataset for clear cell renal cell carcinoma (ccRCC) in Eikrem et al. [13], which contains $18,458$ genes and 32 FFPE samples. The proposed mixture model was fit on the ccRCC data and the MLEs of the parameters were obtained. Then for each setting considered in the first five simulation studies described below, we generated 100 data sets independently, all with $I = 32$ and $J = 18,458$, to examine the impact of one parameter at a time by varying its value from the MLE while fixing other parameters at their respective MLEs.

The first simulation study is to evaluate the performance of the six methods by varying the proportion of expressed genes. In settings I-1 to I-5, the data sets were generated

with $(\boldsymbol{\pi}, \boldsymbol{\delta}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ fixed, but $\phi = 0.59, 0.69, 0.79, 0.89$ and 0.99, respectively. Note that the MLE $\hat{\phi} = 0.79$. Therefore, setting I-3 can be treated as the reference setting in which we used the MLEs for all the parameters.

Study II is to examine the performance under different scenarios of the sample-specific effects. In setting II-1, μ_is were increased to 2 times of their MLEs, reflecting the situation where expressed genes have larger sample-specific effects. In setting II-2, to reflect the situation where expressed genes have larger variability in their expression levels, we increased σ_is by 2 times. In setting II-3, δ_is were increased to 2 times of their MLEs to reflect the situation where non-expressed genes have larger sample-wise background noise.

Study III is to examine the performance of the methods under different scenarios of the gene-specific effects. In setting III-1, σ_g was increased from 1.5 to 3, reflecting larger variability of the gene-wise noise g_js for expressed genes. In setting III-2, π_js were set to a half of their MLEs, reflecting reduced probabilities of the perfect zero state for non-expressed genes.

Study IV is to evaluate the robustness of MIXnorm when the model assumptions are violated. In setting IV-1, with all the parameter values set to their MLEs and $\sigma_g = 1.5$, we replaced $\mathrm{TN}(\mu_i, \sigma_i^2, 0, +\infty)$ by a generalized Student's t-distribution truncated to $[0, +\infty)$, namely $t_{[0,+\infty)}(\mu_i^t, \sigma_i^t, \nu)$, for expressed genes. We set the number of degrees of freedom ν to 3, creating the heaviest tails when the second moment of $t_{[0,+\infty)}(\mu_i^t, \sigma_i^t, \nu)$ exists, and set μ_i^t and σ_i^t to the values such that the mean and variance of $t_{[0,+\infty)}(\mu_i^t, \sigma_i^t, \nu)$ were equal to the mean and variance of $\mathrm{TN}(\mu_i, \sigma_i^2, 0, +\infty)$ [27]. In setting IV-2, we replaced $\mathrm{TN}(\mu_i, \sigma_i^2, 0, +\infty)$ by Gamma(α_i, β_i) for expressed genes. We set $\alpha_i = 2$ so that the gamma distribution has a positive skewness equal to $\sqrt{2}/2$ and β_i was chosen so that the mean of the gamma distribution was matched to that of TN$(\mu_i, \sigma_i^2, 0, +\infty)$, and all other parameters were kept the same as in setting IV-1. In setting IV-3, we used the PoissonSeq model (i.e. model 3.1 from [31]) to generate data, with modification that intends to better

mimic real FFPE samples. In their model 3.1, the raw count $C_{ij} \sim \text{Poisson}(u_{ij})$, and $\log u_{ij} = \log(d_i) + \log(\beta_j) + \alpha_i + \gamma_j y_i$, where d_i is the sequencing depth of sample i, β_j is the expression level of gene j, γ_j is the slope for the association of gene j with the quantitative outcome y_i, and α_i is added to reflect different degradation levels for FFPE samples. To account for excess zeros in FFPE data, $(1-\phi) \times 18,458$ genes were set to be not expressed with $\beta_j = 1 \times 10^{-5}$, where ϕ is set to its MLE estimated from the ccRCC data. All the other model parameters were generated based on the setting in Section 6 of the Supplementary Material in [31].

Study V is to compare two normalization strategies when differentially expressed (DE) genes exist between two experimental conditions: separate normalization for each condition vs. pooled normalization for both conditions. Here, we randomly assigned the 32 samples into two equal sized groups. Then $d\%$ of the expressed genes were randomly selected and set to be non-expressed genes in one group, whose read counts were generated from ZIP distributions. We set d to 5, 10 and 15 for settings V-1, V-2 and V-3. All the other parameters were set to their MLEs as in setting I-3.

The last study VI is to examine the the computing time of MIXnorm when applied to data sets with different sample sizes or numbers of genes. In setting VI-1, data were generated with parameters fixed at their MLEs, $J = 18,458$, and $\sigma_g = 1.5$, but the sample size I was set to a multiple of 32 (i.e., $I = 32m$ for $m = 1, 2, \ldots, 48$), where sample-specific parameters μ_i, σ_i and δ_i were duplicated m times correspondingly. The average computing time for MIXnorm over 50 replicate data sets was recorded for each m. When the sample size is large, monitoring convergence by the change of likelihood between consecutive iterations becomes numerically unreliable. Thus, the maximum change among parameter updates for all parameters were used to detect convergence in this simulation study. In setting VI-2, the sample size was fixed at 32 but the number of genes was raised from $20,000$ to $50,000$ with $1,000$ increment. The proportion of expressed genes was set to the MLE of ϕ.

A.3.2. Results

For a specific sample i, the true expression in the log scale for any expressed gene j was calculated by subtracting the gene-wise noise g_j and the sample-specific effect (i.e., the mean of the TN distribution given by $\mu_i + \psi\left(-\frac{\mu_i}{\sigma_i}\right)\sigma_i/\Phi\left(-\frac{\mu_i}{\sigma_i}\right)$) from the generated log expression L_{ij}. The true expression for any non-expressed gene is zero across all samples. The gene-wise Pearson correlations for the $18,458$ genes between the normalized and true expression were calculated for each simulated data set and the combined data set based on all the 100 replicates to evaluate the performance of the six methods on normalization. Since MIXnorm works on the log transformed counts directly, for the existing methods, we transformed the (normalized and true) expression levels into the log scale before calculating the correlations to ensure a fair comparison. There exist cases where the correlations are ill defined. For genes that have zero standard deviation (SD) in both normalized and true expression, the correlations are simply set to 1 because they are actually the genes with true and normalized expression both equal to zero across all samples. For genes that have zero SD in true (normalized) expression, but not in normalized (true) expression, we added a small amount of disturbance generated from $N(0, 1 \times 10^{-4})$ to the true (normalized) expression when the corresponding normalized (true) expression is not zero, in order to compute the correlations. Within each data set, the first, second (i.e., median) and third quartiles of the $18,458$ gene-wise Pearson correlations were recorded as summary statistics. In each simulation setting, box-plots were used to summarize the three quartiles from the 100 replicates, and the box-plot of the $18,458$ gene-wise Pearson correlations calculated from the combined data, which contains 100×32 samples, were also generated.

Figure A.1 shows that when the proportion of expressed genes ϕ varies from 0.59 to 0.99, the third quartile appears to be more affected than the first two quartiles for MIXnorm. In contrast, for the other five methods, the first quartile appears to be more

affected than the last two quartiles. MIXnorm performs the best in settings I-1 to I-3 with regard to all the three quartiles. In settings I-4 and I-5 with high proportions of expressed genes 0.89 and 0.99, MIXnorm, PS and DEseq have comparable top performance, followed by RPM and TMM, and last by UQ. The all-correlation plots in Figure A.1 confirm that overall, MIXnorm has a large advantage over the other methods when ϕ is relatively low, which is often true for FFPE samples due to the RNA degradation. This advantage may come from the precise identification of expressed genes, which is a unique feature of MIXnorm compared to the other normalization methods. Table A.1 reports the average proportion of genes identified as expressed (i.e., genes with $w_j^{(T)} > 0.5$) by MIXnorm and the average Area Under Curve (AUC) of receiver operating characteristic (ROC) curves under each of the five settings. The proportions identified are very close to true proportions and the AUCs are all close to 1, which clearly shows that MIXnorm is highly accurate in identifying expressed genes.

The results from Study II are shown in Figure A.2, where the different sample-specific effects μ_i, σ_i and δ_i are examined. The boxplots show that among the six methods, MIXnorm gives the highest gene-wise correlations in all three settings. Further, the all-correlation plots show that MIXnorm has the shortest whiskers on the left, meaning that MIXnorm consistently produces fewer negative or low correlations. Among the existing methods, PS and DEseq appear to perform the best, and can offer performance close or comparable to MIXnorm in terms of the three quartiles.

Figure A.3 presents results from Study III, which investigates the impact of gene-wise effects on the performance of the methods. In setting III-1 where the variability of gene-wise noise σ_g was increased to 3 for expressed genes, the performance of all the six methods becomes worse, compared to the reference setting I-3. However, MIXnorm still shows advantage over the other methods with higher correlation quantiles. In setting III-2 where π_js were set to a half of their MLEs to increase the gene-wise noise for non-expressed genes, all the methods show lower correlation quartiles compared to those of

setting I-3. But MIXnorm is less sensitive and yields much higher first quartiles and higher other quartiles than the other methods.

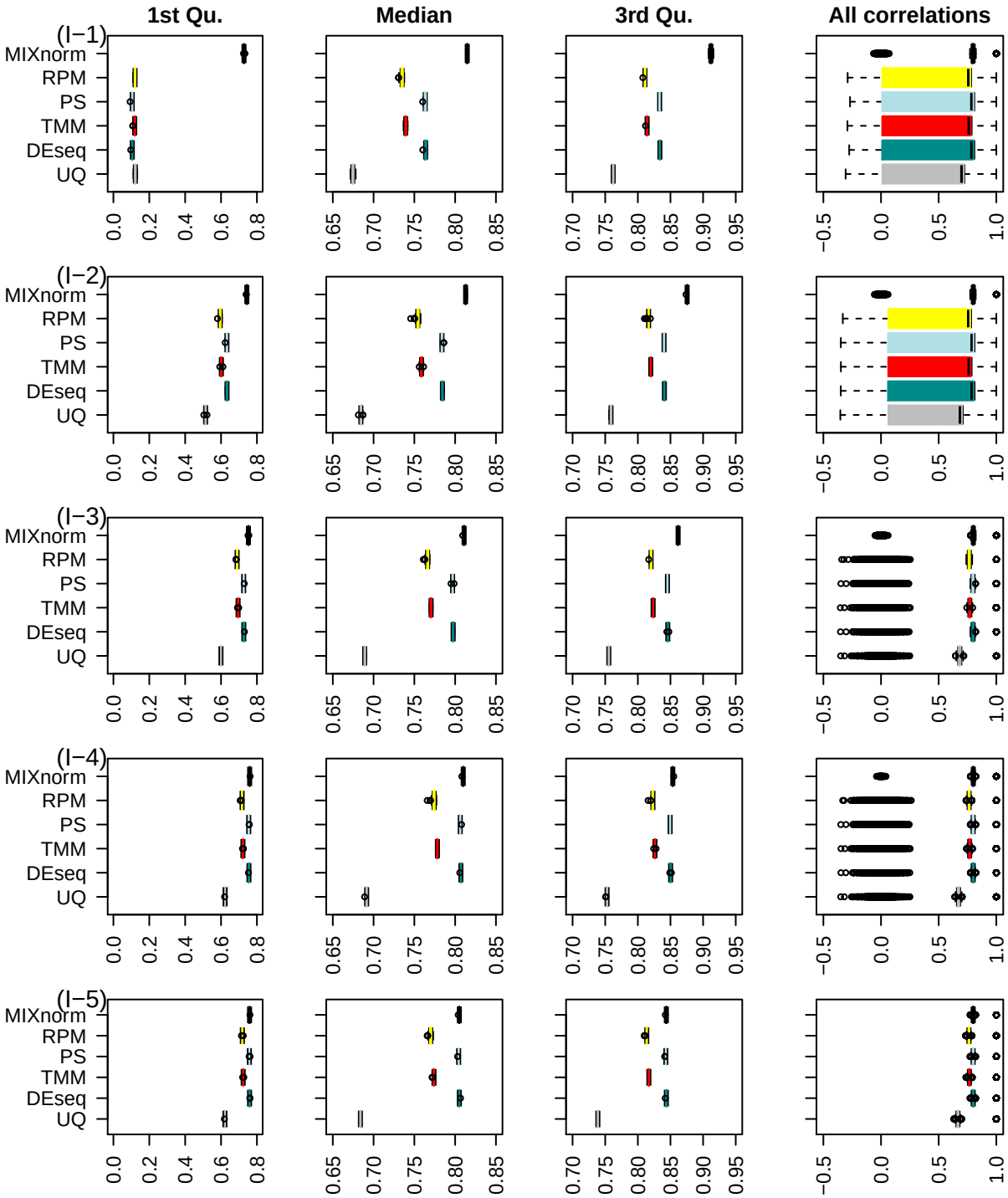


Figure A.1: Simulation study I: the first three columns show box-plots for the 1st to 3rd quartiles of 18, 458 gene-wise correlations between normalized and true expression levels based on 100 replicates for the five settings I1 - I5 that vary the proportion of expressed genes ϕ from 0.59 to 0.99 by 0.1; the last column shows the box-plot of the $18,458$ gene-wise Pearson correlations calculated from the combined data, which contains 100×32 samples in each setting. The MLE $\hat{\phi}$ estimated from the ccRCC data is 0.79, which is the value of ϕ in Setting I-3 that can be treated as the reference setting.

Setting	Proportion of expressed genes	Avg. proportion detected	Avg. AUC
I-1	0.59	0.595	0.994
I-2	0.69	0.691	0.998
I-3	0.79	0.790	1.000
I-4	0.89	0.890	1.000
I-5	0.99	0.990	1.000

Table A.1: Simulation study I: the performance of MIXnorm in identifying expressed genes, measured by the average proportion of genes detected as expressed (column 3) and the average AUC (column 4) in each of the five settings I1 – I5. Gene j is identified as expressed if $w_j^{(t)} > 0.5$ in the last iteration.

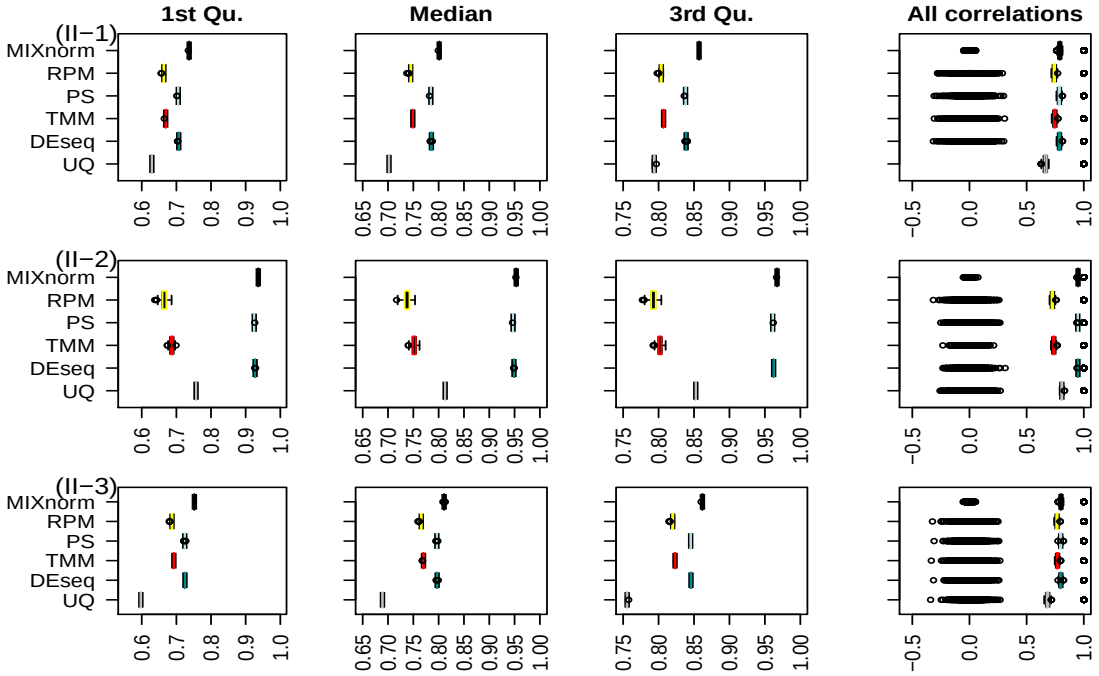


Figure A.2: Simulation study II: the first three columns show box-plots for the 1st to 3rd quartiles of $18,458$ gene-wise correlations between normalized and true expression based on 100 replicates for the three settings II1 – II3; the last column shows the box-plot of the $18,458$ gene-wise Pearson correlations calculated from the combined data, which contains 100×32 samples in each setting. In II-1, the sample specific location parameters μ_is were increased to 2 times of their MLEs obtained from the ccRCC data. In II-2, the sample specific scale parameters σ_is were increased to 2 times of their MLEs. In II-3, δ_is, which control the sample specific background noise for non-expressed genes, were increased to 2 times of their MLEs.

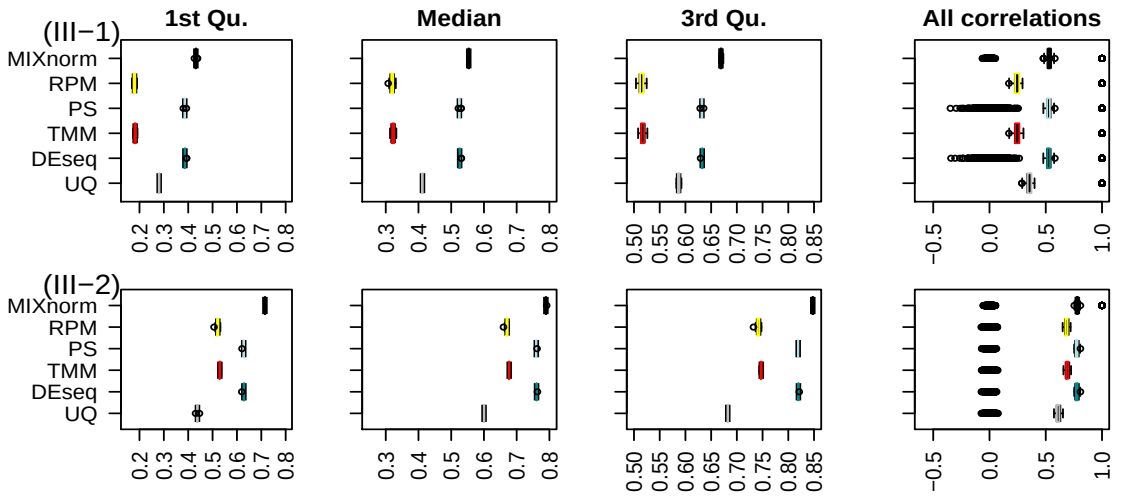


Figure A.3: Simulation study III: the first three columns show box-plots for the 1st to 3rd quartiles of $18,458$ gene-wise correlations between normalized and true expression levels based on 100 replicates for the two settings III1 – III2; the last column shows the box-plot of the $18,458$ gene-wise Pearson correlations calculated from the combined data, which contains 100×32 samples in each setting. In setting III-1, the variability of gene-wise noise σ_g was increased from 1.5 to 3 for expressed genes. In setting III-2, π_js, the probabilities of extra zero for non-expressed genes, were set to a half of their MLEs.

Figure A.4 shows results from Study IV, in which deviations from model assumptions were considered. In settings IV-1 and IV-2, where heavy-tailed and skewed data were generated for expressed genes, respectively, we observe that MIXnorm, PS and DEseq offer better performance than the other methods, among which MIXnorm is better than or at least comparable to PS and DEseq. In setting IV-3, where the (modified) Poisson log-linear model was used to generate data, all the methods produce wider box-plots, indicating larger variability, and MIXnorm beats the other methods in all three quartiles.

Figure A.5 shows that when there exist DE genes, separate normalization yields better performance. The difference between separate and pooled normalization is small when $d = 5$, but it increases as d increases. Thus, we recommend using MIXnorm to normalize data from different conditions separately, especially when the proportion of DE genes is not small.

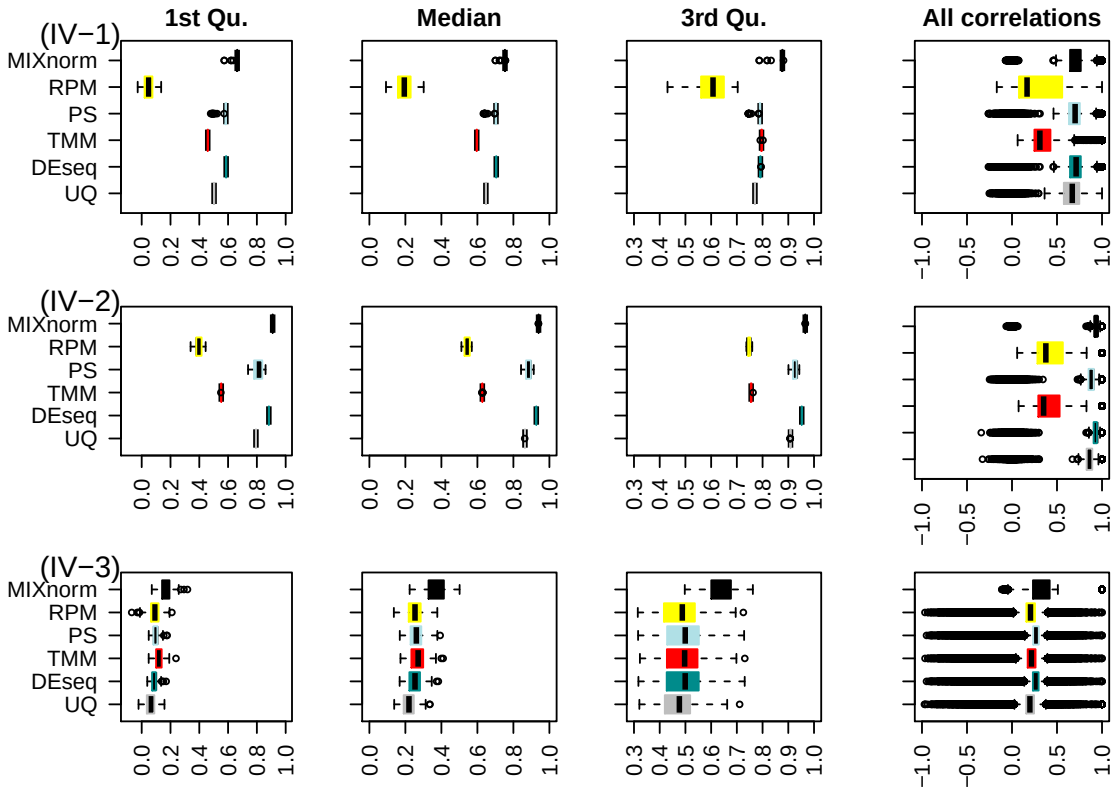


Figure A.4: Simulation study IV: the first three columns show box-plots for the 1st to 3rd quartiles of 18, 458 gene-wise correlations between normalized and true expression based on 100 replicates for the three settings IV1 - IV3; the last column shows the box-plot of the $18,458$ gene-wise Pearson correlations calculated from the combined data, which contains 100×32 samples in each setting. In IV-1, a truncated t-distribution was used to generate heavy-tailed data for expressed genes.In IV-2, a gamma distribution was used to generate skewed data for expressed genes. In IV-3, the Poisson log-linear model (model 3.1 from [31]), with modification to better mimic RNA-seq data from FFPE samples, was used.

Figure A.6 shows the average computing time (in second) of MIXnorm against the sample size I (the left panel) and against the number of genes J (the right panel) while holding the other constant. Clearly, the average computing time increases linearly as I or J increases, with Pearson correlation greater than 0.99, indicating a virtually perfect linear relationship in each scatterplot. MIXnorm takes about 7 minutes for a data set with $\sim 1,500$ samples and $\sim 18,000$ genes.

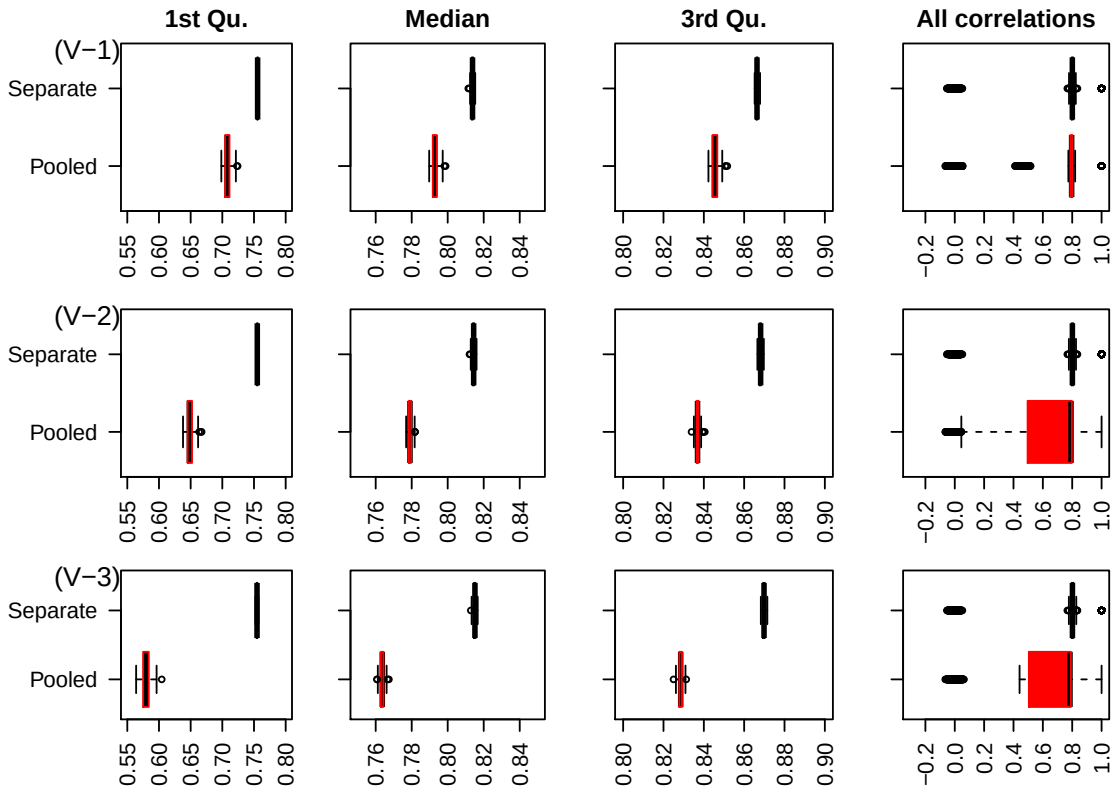


Figure A.5: Simulation study V: the first three columns show box-plots for the 1st to 3rd quartiles of 18,458 gene-wise correlations between normalized and true expression based on 100 replicates for three settings V1 – V3; the last column shows the box-plot of the $18,458$ gene-wise Pearson correlations calculated from the combined data, which contains 100×32 samples in each setting. Two strategies are compared when DE genes exist across two experimental conditions: separate normalization for each condition vs. pooled normalization for both conditions. Each setting contains $d\%$ genes that are differentially expressed, where d was set to 5, 10 and 15, respectively.

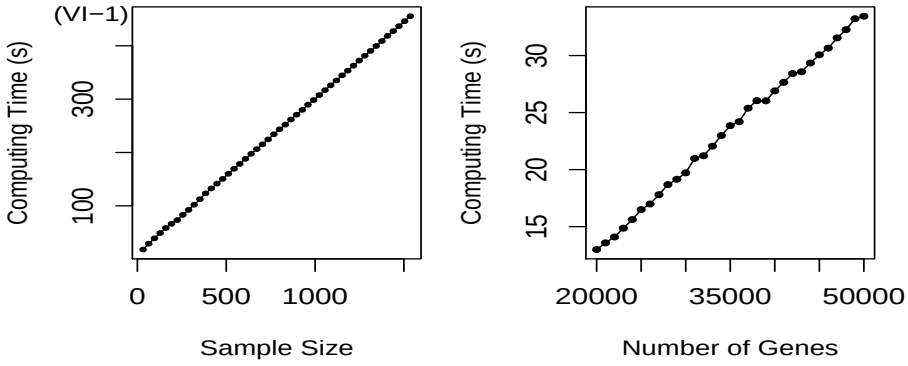


Figure A.6: Simulation study VI: average computing time of MIXnorm vs. sample size I (left panel) and number of genes J (right panel)

A.4. Additional Results for Data Applications

To check the suitability of MIXnorm for the ccRCC data, Figure A.7 shows empirical densities of log read counts for the 32 paired FFPE and RNAlater samples. As in Figure 1 for the soft tissue sarcomas data in the main paper, the density curves are all bimodal, with one spike at zero, which necessitates the use of a zero-inflated distribution for non-expressed genes. Also, the curves around the second mode are roughly bell-shaped, suggesting that (truncated) normality is plausible for expressed genes. It is worth noting that in this ccRCC example, the FFPE and FF density plots are quite similar, which implies that the quality of FFPE data is almost identical to that of RNAlater data. Also, the location of L_{ij} may vary from sample to sample but the spread of L_{ij} appears to be quite constant, meaning that we need the sample-specific mean μ_i but could reduce the variance σ_i^2 to a constant σ^2.

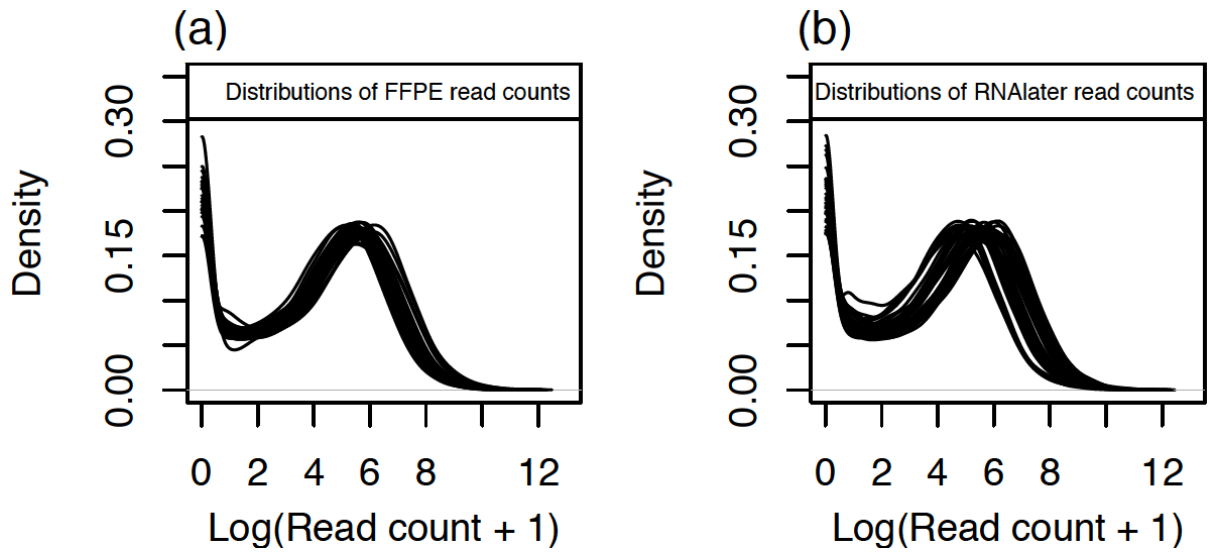


Figure A.7: An exploratory analysis of ccRCC RNA-seq data in Eikrem et al. (2016). Panel (a) plots the empirical densities of log read counts for the 32 FFPE samples. Panel (b) plots the empirical densities of log read counts for the paired RNAlater samples. Each curve represents the density for one sample across all the $18,458$ protein coding genes.

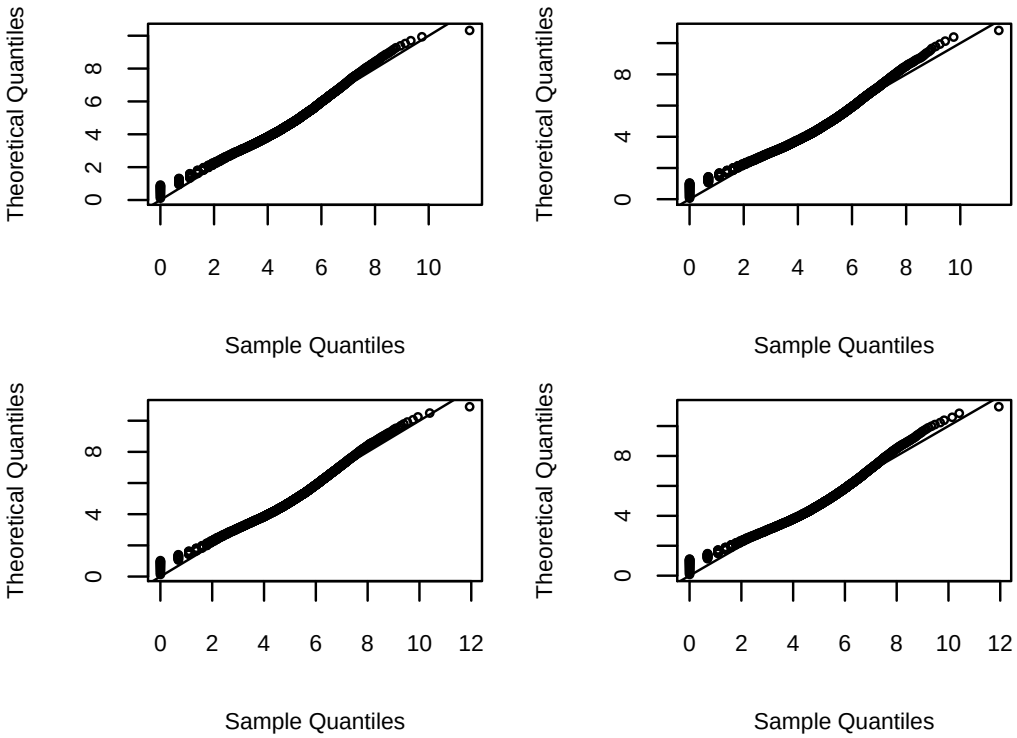


Figure A.8: Q-Q plots using the expressed genes identified by MIXnorm for four randomly selected FFPE samples (NF10, NF31, TF32 and TF33) from ccRCC RNA-seq data. The raw reads were transformed into the log scale to calculate sample quantiles. Theoretical quantiles were calculated from the TN distributions with sample-specific location and scale parameters estimated by MIXnorm.

Besides the above explanatory analysis, we suggest conducting a confirmatory analysis after applying MIXnorm, by visually examining Q-Q plots or conducting distributional tests, to check whether the assumption of (truncated) normality is adequate for expressed genes in most of the samples. We first identified expressed genes by applying MIXnorm. Among the 32 samples in the ccRCC data, we randomly selected 4 samples to show the Q-Q plots in Figure A.8. The theoretical quantiles for the expressed genes were calculated using $(\hat{\mu}_i, \hat{\sigma}_i^2)$, the sample-specific TN parameters estimated from the nested EM algorithm. We further conducted a distributional test for each sample, where the difference between the distribution of log transformed reads for expressed genes and the theoretical probability distribution $TN(\hat{\mu}_i, \hat{\sigma}_i^2, 0, +\infty)$ was measured by the Kullback–Leibler (KL) divergence. We employed bootstrap resampling to control the type I error rate at the sig-

nificance level 0.05. With 500 bootstrap samples, the p-value was empirically calculated as the proportion of bootstrap KL divergence greater than the observed KL divergence. We summarized the p-values from the bootstrap KL tests in Figure A.9. Both Q-Q plots and p-values suggest that there was no gross departure from the assumed TN distributions.

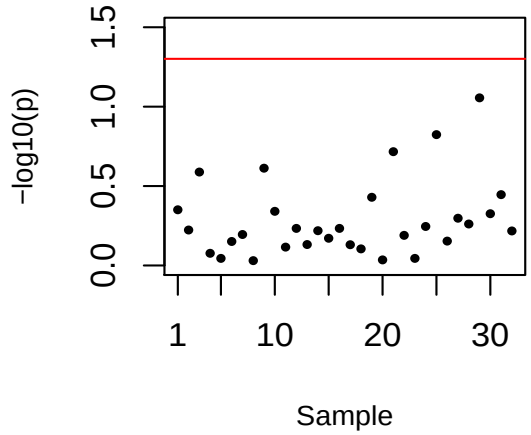


Figure A.9: Results from bootstrap KL distributional tests for the 32 FFPE samples from ccRCC RNA-seq data. Note that the y-axis represents $-\log_{10}$(p-value). The horizontal line is located at $y = -\log_{10}(0.05)$. All p-values are greater than 0.05.

To investigate the impact of removing genes with low expression on normalization performance, we adopted the quality control criterion from PoissonSeq [31]. Genes with mean reads across all samples less than or equal to 0.5 were excluded for normalization and correlation calculation. The preprocessed soft tissue sarcomas data contain $19,408$ genes and the ccRCC data contain $16,044$ genes. All the 67 genes in the CINSARC gene signature passed the quality control. Table A.2 summarizes results based on gene-wise correlations, where the left panel is for the CINSARC gene signature and the middle panel is for the 19,408 genes in the soft tissue sarcomas data, and the right panel is for the $16,044$ genes in the ccRCC data. MIXnorm gives similar results regardless of conducting quality control or not. DEseq and RPM normalization shows significant improvement on the CINSARC gene signature, especially in terms of the median and 3rd quartile. Among

the six methods, MIXnorm is still the best, though it is overtaken by DEseq and RPM for the 3rd Quartile in the CINSARC gene signature. Note that the selection of quality control methods and thresholds can be subjective. MIXnorm achieves strong performance without such an arbitrary preprocessing step.

Method	Soft tissue sarcomas						ccRCC		
	CINSARC gene signature			$19,408$ protein coding genes			$16,044$ protein coding genes		
	1st. Qu.	Median	3rd. Qu.	1st. Qu.	Median	3rd. Qu.	1st. Qu.	Median	3rd. Qu.
MIXnorm	**0.333**	**0.456**	0.517	**0.094**	**0.226**	**0.367**	**0.284**	**0.483**	**0.680**
DEseq	0.174	0.370	0.663	0.022	0.158	0.288	0.203	0.418	0.609
RPM	0.142	0.386	**0.676**	0.013	0.154	0.289	0.204	0.422	0.612
TMM	0.082	0.219	0.533	0.025	0.158	0.285	0.110	0.267	0.463
PS	-0.126	0.002	0.154	-0.381	-0.161	0.022	0.107	0.299	0.484
UQ	-	-	-	-	-	-	0.236	0.425	0.606
Original	0.020	0.107	0.181	0.013	0.144	0.269	0.170	0.312	0.477

Table A.2: Data applications: the left and middle panels show gene-wise correlations between normalized FFPE and FF expression for soft tissue sarcomas data; the right panel shows gene-wise correlations between normalized FFPE and RNAlater expression for ccRCC data. Genes with mean reads across all samples less than or equal to 0.5 were excluded for normalization and the correlation calculation. The filtered soft tissue sarcomas data contain $19,408$ genes and ccRCC data contain $16,044$ genes.

APPENDIX B

APPENDIX of CHAPTER 2

B.1. Nested EM algorithm for SMIXnorm

Consider the raw RNA-seq count data summarized in a matrix $\boldsymbol{C}$, where the (i, j) element C_{ij} denotes the number of reads measured for sample i gene j and $L_{ij} = \log(C_{ij}+1)$ for $i = 1, ..., I$ and $j = 1, .., J$. Suppose we knew, through a binary latent variable D_j, if gene j is truly expressed or not. Then we model the count data as a mixture of zero-inflated Poisson (ZIP) and normal distribution,

$$C_{ij} \sim \text{ZIP}(\pi_j, \delta), \text{if } D_j = 0, \tag{B.1}$$

$$L_{ij} \sim \text{N } (\mu_i, \sigma_i^2), \text{if } D_j = 1, \tag{B.2}$$

$$D_j \sim \text{Ber}(\phi),$$

To obtain the maximum likelihood estimates (MLE), direct implementation of the EM algorithm requires Newton-Raphson type optimization for parameters from the ZIP component, which may suffer from both numerical stability and computational efficiency issues [61, 67]. Note that ZIP distribution can be thought of as a mixture of the perfect zero state

and the Poisson state. For each non-expressed gene j, we introduce another binary latent variable Z_{ij}. Assume C_{ij} is from the perfect zero state if $Z_{ij} = 1$ and C_{ij} is from the Poisson state if $Z_{ij} = 0$. Obviously, $Z_{ij}|D_j = 0 \sim \text{Ber}(\pi_j)$. The complete data log-likelihood with latent variables $\boldsymbol{D}$ and $\mathbf{Z}$ is given by

$$\begin{aligned}\ell(\Theta|\mathbf{C}, \boldsymbol{D}, \mathbf{Z}) = \sum_{j=1}^{J}\sum_{i=1}^{I}\Big\{ & D_j \left[\log\phi + \log \mathrm{N}(L_{ij}|\mu_i, \sigma_i) - \log(C_{ij}+1)\right] \\ & + (1-D_j)\left[\log(1-\phi) + Z_{ij}\log\pi_j + (1-Z_{ij})\log(1-\pi_j)\right] \\ & + (1-D_j)(1-Z_{ij})\left[C_{ij}\log\delta - \delta - \log C_{ij}!\right]\Big\},\end{aligned} \tag{B.3}$$

where Θ denote the set of all parameters. Let t be the current iteration of the nested EM algorithm. Following van Dyk [61] and Yin et al. [67], the nested EM algorithm first treats $\boldsymbol{C}$ as observed data and $\boldsymbol{D}$ as missing data in the outer EM and calculates the conditional expectation $w_j^{(t+1)} = \mathbf{E}\left(D_j|\boldsymbol{C}_j, \Theta^{(t)}\right)$ by

$$w_j^{(t+1)} = \frac{\phi^{(t)} p(\boldsymbol{C}_j|D_j = 1, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)})}{\phi^{(t)} p(\boldsymbol{C}_j|D_j = 1, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)}) + (1-\phi^{(t)}) p(\boldsymbol{C}_j|D_j = 0, \pi_j^{(t)}, \delta^{(t)})}, \tag{B.4}$$

where $p(\boldsymbol{C}_j|D_j = 1, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{(t)})$ is the normal probability density function and $p(\boldsymbol{C}_j|D_j = 0, \pi_j^{(t)}, \delta^{(t)})$ is the ZIP probability mass function as described in (B.1) and (B.2). The inner

EM treats $(\boldsymbol{C}, \boldsymbol{w}^{(t+1)})$ as observed data and $\boldsymbol{Z}$ as missing data and calculates $Z_{ij}^{\left(t+\frac{k-1}{K}\right)}$ by

$$
\begin{aligned}
Z_{ij}^{\left(t+\frac{k-1}{K}\right)} &= \mathbf{E}\left(Z_{ij}|C_{ij}, w_j^{(t+1)}, \Theta^{\left(t+\frac{k-1}{K}\right)}\right) = \Pr\left(Z_{ij}=1|C_{ij}, w_j^{(t+1)}, \Theta^{\left(t+\frac{k-1}{K}\right)}\right) \\
&= \frac{p\left(Z_{ij}=1, C_{ij}, w_j^{(t+1)}|\Theta^{\left(t+\frac{k-1}{K}\right)}\right)}{p\left(Z_{ij}=1, C_{ij}, w_j^{(t+1)}|\Theta^{\left(t+\frac{k-1}{K}\right)}\right) + p\left(Z_{ij}=0, C_{ij}, w_j^{(t+1)}|\Theta^{\left(t+\frac{k-1}{K}\right)}\right)} \\
&= \begin{cases} \dfrac{\left(\pi_j^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t+1)}}}{\left(\pi_j^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t+1)}} + \left(1-\pi_j^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t+1)}} \exp\left(-\delta_i^{\left(t+\frac{k-1}{K}\right)}\right)^{1-w_j^{(t+1)}}} & \text{if } C_{ij}=0; \\ 0 & \text{if } C_{ij}\neq 0, \end{cases}
\end{aligned}
\tag{B.5}
$$

where $k = 1, ..., K$ is the current cycle of the inner EM. The maximization step maximizes the conditional expected complete data log-likelihood, which can be obtained by replacing $\boldsymbol{D}$ and $\boldsymbol{Z}$ by $\boldsymbol{w}^{(t+1)}$ and $\boldsymbol{Z}^{\left(t+\frac{k-1}{K}\right)}$ in (B.3), with respect to Θ. The proposed nested EM algorithm is summarized as follow:

Require: $t = 0$, convergence criteria ϵ, tolerance $= \epsilon + 1$, initialize $(\mu_i^{(0)}, \sigma_i^{(0)}, \pi_j^{(0)}, \delta^{(0)}, \phi^{(0)})$

while tolerance $> \epsilon$ **do**

Calculate

$w_j^{(t+1)} = \mathbf{E}\left(D_j|\boldsymbol{C}_j, \Theta^{(t)}\right)$ from (B.4)

for k in $1, 2, ..., K$ **do**

Calculate

$Z_{ij}^{\left(t+\frac{k-1}{K}\right)} = \mathbf{E}\left(Z_{ij}|C_{ij}, w_j^{(t+1)}, \Theta^{\left(t+\frac{k-1}{K}\right)}\right)$ from (B.5),

Update

$$\pi_j^{\left(t+\frac{k}{K}\right)} = \frac{\sum_{i=1}^{I} Z_{ij}^{\left(t+\frac{k-1}{K}\right)}}{I}$$

$$\delta^{\left(t+\frac{k}{K}\right)} = \frac{\sum_{j=1}^{J}\left(1-w_j^{(t+1)}\right)\left(1-Z_{ij}^{\left(t+\frac{k-1}{K}\right)}\right)C_{ij}}{\sum_{j=1}^{J}\left(1-w_j^{(t+1)}\right)\left(1-Z_{ij}^{\left(t+\frac{k-1}{K}\right)}\right)}$$

end for

Update

$\phi^{(t+1)} = \frac{\sum_{j=1}^{J} w_j^{(t+1)}}{J}$

$\mu_i^{(t+1)} = \frac{\sum_{j=1}^{J} w_j^{(t+1)} L_{ij}}{\sum_{j=1}^{J} w_j^{(t+1)}}$

$\sigma_i^{(t+1)} = \sqrt{\frac{\sum_{j=1}^{J} w_j^{(t+1)} L_{ij}^2}{\sum_{j=1}^{J} w_j^{(t+1)}} - \left(\mu_i^{(t+1)}\right)^2}$

Set $\left(\boldsymbol{\pi}^{(t+1)}, \delta^{(t+1)}\right) = \left(\boldsymbol{\pi}^{(t+\frac{K}{K})}, \delta^{(t+\frac{K}{K})}\right)$

tolerance $= |[\ell(\Theta^{(t)}|\boldsymbol{C}) - \ell(\Theta^{(t+1)}|\boldsymbol{C})]/\ell(\Theta^{(t)}|\boldsymbol{C})|$

$t = t + 1$

end while

return MLE for all the parameters $\hat{\Theta}$

B.2. Web Appendix B: Additional simulation results

The following modified MIXnorm model [67] was used to generate synthetic data sets:

$$C_{ij} \sim \text{ZIP}(\pi_j, \delta_i), \text{for } j > \phi J,$$

$$L_{ij} \sim \text{TN } (\mu_i + g_j, \sigma_i^2, 0, \infty), \text{for } j \leq \phi J,$$

$$g_j \sim \text{N}(0, \sigma_g^2),$$

where g_j is extra gene-wise noise and $\sigma_g = 1.5$. Other model parameters were set to their MLEs estimated from a public RNA-seq dataset from FFPE soft tissue sarcomas samples [30], which contains expression levels for $20,242$ protein-coding genes from 41 patients.

The performance of seven normalization methods is evaluated by the gene-wise Pearson correlations for the $20,242$ genes between the normalized and true expression. Fig. B.1 summarizes the correlation quantiles for the 50 simulated data sets under the setting of 41

samples. SMIXnorm and MIXnorm show almost identical correlation quartiles and have significantly higher quartiles than others. TMM performs the best among the five existing FF RNA-seq normalization methods. The most straightforward method, RPM, gives even worse results than the use of original data without any normalization.

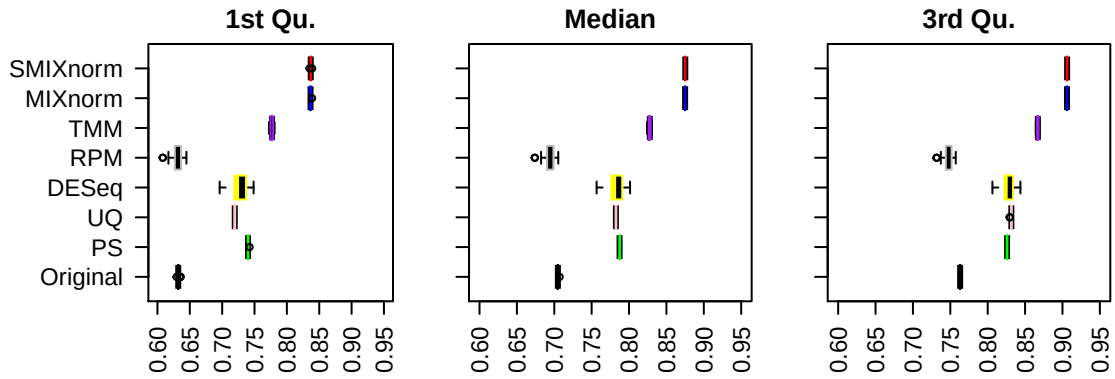


Figure B.1: Simulation study I. The 1st to 3rd quartiles of gene-wise Pearson correlations for $20,242$ genes between the normalized and true expression for 50 simulated data set under the setting of 41 samples.

APPENDIX C

APPENDIX of CHAPTER 3

C.1. MCMC algorithm

An efficient MCMC algorithm is designed to sample from the posterior distributions of the model parameters and latent variables. As discussed in Section 3.2.3, we have integrated out the Gaussian Process mean and variance scaling parameter to speed up the MCMC convergence. In each MCMC iteration, we can directly sample $\boldsymbol{G}$, and via a data-augmentation method based on Polya-Gamma distributions, we can directly sample $\boldsymbol{\beta}$ as well. Below we provide the M-H algorithms built in the Gibbs sampler to update $\boldsymbol{\lambda}$ and $\boldsymbol{Z}$.

Update of the underlying expression level Z_i. From (3.3), $Z_i \equiv Y_i$ with probability 1 given $G_i = 1$. We update Z_i that corresponds to $G_i = 0$ sequentially by a Metropolis-Hastings algorithm. We propose a new z_i^* from Poi(z_i) and then accept the proposed value with probability min$(1, m_z)$, where the Hastings ratio is

$$m_z = \frac{p(z_i^*|G_i = 0, \lambda_i, \beta_0, \beta_1)J(z_i|z_i^*)}{p(z_i|G_i = 0, \lambda_i, \beta_0, \beta_1)J(z_i^*|z_i)},$$

where $J(\cdot|\cdot)$ is the probability mass function of the proposal distribution. We set $J(\cdot|0)$ to a Poisson distribution with mean 1 to avoid the algorithm from sticking at 0.

Update of the normalized expression level λ_i. We update each λ_i in natural logarithm scale sequentially for $i = 1, ..., I$ by a random walk Metropolis-Hastings algorithm. We first propose a new $\log \lambda_i^*$ from N$(\log \lambda_i, \sigma_\lambda^2)$. Then accept the proposed $\log \lambda_i^*$ with probability min$(1, m_z)$, where the Hastings ratio is

$$m_\lambda = \frac{p(\log \lambda_i^* | \boldsymbol{Z}, l, \log \boldsymbol{\lambda}_{-i}) J(\log \lambda_i | \log \lambda_i^*)}{p(\log \lambda_i | \boldsymbol{Z}, l, \log \boldsymbol{\lambda}_{-i}) J(\log \lambda_i^* | \log \lambda_i)},$$

where the proposal density ratio $\frac{J(\log \lambda_i | \log \lambda_i^*)}{J(\log \lambda_i^* | \log \lambda_i)} = 1$ in this random walk Metropolis-Hastings algorithm.

BIBLIOGRAPHY

[1] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, Oct 2010.

[2] Rhonda Bacher, Li-Fang Chu, Ning Leng, Audrey P Gasch, James A Thomson, Ron M Stewart, Michael Newton, and Christina Kendziorski. Scnorm: robust normalization of single-cell rna-seq data. *Nature methods*, 14(6):584, 2017.

[3] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[4] James Bullard, Elizabeth Purdom, Kasper Hansen, and Sandrine Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC Bioinformatics*, 11(1):94, 02 2010.

[5] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[6] Huiyi Chen, Katsuyuki Shiroguchi, Hao Ge, and Xiaoliang Sunney Xie. Genome-wide study of mrna degradation and transcript elongation in escherichia coli. *Molecular systems biology*, 11(1), 2015.

[7] Kok Hao Chen, Alistair N Boettiger, Jeffrey R Moffitt, Siyuan Wang, and Xiaowei Zhuang. Spatially resolved, highly multiplexed rna profiling in single cells. *Science*, 348(6233), 2015.

[8] Li Chen, James Reeve, Lujun Zhang, Shengbing Huang, Xuefeng Wang, and Jun Chen. Gmpr: A robust normalization method for zero-inflated count data with application to microbiome sequencing data. *PeerJ*, 6:e4600, 2018.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J Royal Stat Soc: Ser B (Stat Methodol)*, 39(1):1–38, 1977.

[10] Elena Denisenko, Belinda B Guo, Matthew Jones, Rui Hou, Leanne De Kock, Timo Lassmann, Daniel Poppe, Olivier Clément, Rebecca K Simmons, Ryan Lister, et al. Systematic assessment of tissue dissociation and storage biases in single-cell and single-nucleus rna-seq workflows. *Genome biology*, 21:1–25, 2020.

[11] Marie-Agnès Dillies, Andrea Rau, Julie Aubert, Christelle Hennequet-Antier, Marine Jeanmougin, Nicolas Servant, Céline Keime, Guillemette Marot, David Castel, Jordi Estelle, Gregory Guernec, Bernd Jagla, Luc Jouneau, Denis Laloë, Caroline Le Gall, Brigitte Schaëffer, Stéphane Le Crom, MickaÃ«l Guedj, and Florence Jaffrézic. A comprehensive evaluation of normalization methods for illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics*, 14(6):671–683, 2013.

[12] Daniel Edsgärd, Per Johnsson, and Rickard Sandberg. Identification of spatial expression trends in single-cell gene expression data. *Nature methods*, 15(5):339–342, 2018.

[13] Oystein Eikrem, Christian Beisland, Karin Hjelle, Arnar Flatberg, Andreas Scherer, Lea Landolt, Trude Skogstrand, Sabine Leh, Vidar Beisvag, and Hans-Peter Marti. Transcriptome Sequencing (RNAseq) Enables Utilization of Formalin-Fixed, Paraffin-Embedded Biopsies with Clear Cell Renal Cell Carcinoma for Exploration of Disease Biology and Biomarker Development. *PLoS ONE*, 11(2), 2016.

[14] Chee-Huat Linus Eng, Michael Lawson, Qian Zhu, Ruben Dries, Noushin Koulena, Yodai Takei, Jina Yun, Christopher Cronin, Christoph Karp, Guo-Cheng Yuan, et al. Transcriptome-scale super-resolved imaging in tissues by rna seqfish+. *Nature*, 568(7751):235–239, 2019.

[15] Ciaran Evans, Johanna Hardin, and Daniel M Stoebel. Selecting between-sample rna-seq normalization methods from the perspective of their assumptions. *Briefings in bioinformatics*, 19(5):776–792, 2018.

[16] Andrea M Femino, Fredric S Fay, Kevin Fogarty, and Robert H Singer. Visualization of single rna transcripts in situ. *Science*, 280(5363):585–590, 1998.

[17] Christine Gjerdrum, Crina Tiron, Torill Høiby, Ingunn Stefansson, Hallvard Haugen, Tone Sandal, Karin Collett, Shan Li, Emmet McCormack, Bjørn Tore Gjertsen, et al. Axl is an essential epithelial-to-mesenchymal transition-induced regulator of breast cancer metastasis and patient survival. *Proceedings of the National Academy of Sciences*, 107(3):1124–1129, 2010.

[18] Alberto Monterro Glueck and Muaiad Kittane. Molecular profiling for breast cancer: A comprehensive review. *Biomarkers in Cancer*, page 61, 2013.

[19] Stefan Graw, Richard Meier, Kay Minn, Clark Bloomer, Andrew K. Godwin, Brooke Fridley, Anda Vlad, Peter Beyerlein, and Jeremy Chien. Robust gene expression and mutation analyses of RNA-sequencing of formalin-fixed diagnostic tumor samples. *Scientific Reports*, 5, 2015.

[20] Jennifer K. Grenier, Polly A. Foureman, Erica A. Sloma, and Andrew D. Miller. RNA-seq transcriptome analysis of formalin fixed, paraffin-embedded canine meningioma. *PLoS ONE*, 12(10):1–17, 10 2017.

[21] Jennifer Hansson, David Lindgren, Helén Nilsson, Elinn Johansson, Martin Johansson, Lena Gustavsson, and Håkan Axelson. Overexpression of functional slc6a3 in clear cell renal cell carcinoma. *Clinical Cancer Research*, 23(8):2105–2115, 2017.

[22] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[23] Daniel F Heitjan and Donald B Rubin. Inference from coarse data via multiple imputation with application to age heaping. *Journal of the American Statistical Association*, 85(410):304–314, 1990.

[24] Daniel F Heitjan and Donald B Rubin. Ignorability and coarse data. *The annals of statistics*, pages 2244–2253, 1991.

[25] Rafael A Irizarry, Bridget Hobbs, Francois Collin, Yasmin D Beazer-Barclay, Kristen J Antonellis, Uwe Scherf, and Terence P Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[26] Shintaro Katayama, Virpi Töhönen, Sten Linnarsson, and Juha Kere. Samstrt: statistical test for differential expression in single-cell transcriptome with spike-in normalization. *Bioinformatics*, 29(22):2943–2945, 2013.

[27] Hea-Jung Kim. Moments of truncated student-t distribution. *Journal of the Korean Statistical Society*, 37(1):81 – 87, 2008.

[28] Diane Lambert. Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing. *Technometrics*, 34(1):1–14, 1992.

[29] Tom Lesluyes, Gaëlle Pérot, Marine Roxane Largeau, Céline Brulard, Pauline Lagarde, Valérie Dapremont, Carlo Lucchesi, Agnès Neuville, Philippe Terrier, Dominique Vince-Ranchère, et al. RNA sequencing validation of the Complexity INdex in SARComas prognostic signature. *European Journal of Cancer*, 57:104–111, 2016.

[30] Tom Lesluyes, Gaëlle Pérot, Marine Roxane Largeau, Céline Brulard, Pauline Lagarde, Valérie Dapremont, Carlo Lucchesi, Agnès Neuville, Philippe Terrier, Dominique Vince-Ranchère, et al. Rna sequencing validation of the complexity index in sarcomas prognostic signature. *European Journal of Cancer*, 57:104–111, 2016.

[31] Jun Li, Daniela M Witten, Iain M Johnstone, and Robert Tibshirani. Normalization, testing, and false discovery rate estimation for RNA-sequencing data. *Biostatistics (Oxford, England)*, 13(3):523–538, July 2012.

[32] Ping Li, Andrew Conley, Hao Zhang, and Hyung L Kim. Whole-transcriptome profiling of formalin-fixed, paraffin-embedded renal cell carcinoma by rna-seq. *BMC genomics*, 15(1):1087, 2014.

[33] Xuanhui Sharron Lin, Lan Hu, Kirley Sandy, Mick Correll, John Quackenbush, Chin-Lee Wu, and William Scott McDougal. Differentiating progressive from non-progressive t1 bladder cancer by gene expression profiling: applying RNA-sequencing analysis on archived specimens. In *Urologic Oncology: Seminars and Original Investigations*, volume 32, pages 327–336. Elsevier, 2014.

[34] Eric Lubeck, Ahmet F Coskun, Timur Zhiyentayev, Mubhij Ahmad, and Long Cai. Single-cell in situ rna profiling by sequential hybridization. *Nature methods*, 11(4):360, 2014.

[35] Aaron TL Lun, Karsten Bach, and John C Marioni. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. *Genome biology*, 17(1):75, 2016.

[36] Robert C McCarthy, Andrew G Breite, Michael L Green, and Francis E Dwulet. Tissue dissociation enzymes for isolating human islets for transplantation: factors to consider in setting enzyme acceptance criteria. *Transplantation*, 91(2):137, 2011.

[37] 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.

[38] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods*, 5:621 EP–, May 2008.

[39] Matthew L Morton, Xiaodong Bai, Callie R Merry, Philip A Linden, Ahmad M Khalil, Rom S Leidner, and Cheryl L Thompson. Identification of mrnas and lincrnas associated with lung cancer progression using next-generation rna sequencing from laser micro-dissected archival ffpe tissue specimens. *Lung cancer*, 85(1):31–39, 2014.

[40] John G Nutt, Julie H Carter, and Gary J Sexton. The dopamine transporter: importance in parkinson's disease. *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, 55(6):766–773, 2004.

[41] Bernard Omolo, Mingli Yang, Fang Yin Lo, Michael J Schell, Sharon Austin, Kellie Howard, Anup Madan, and Timothy J Yeatman. Adaptation of a ras pathway activation signature from ff to ffpe tissues in colorectal cancer. *BMC medical genomics*, 9(1):65, 2016.

[42] Mark A Perlmutter, Carolyn JM Best, John W Gillespie, Yvonne Gathright, Sergio GonzÃ¡lez, Alfredo Velasco, W Marston Linehan, Michael R Emmert-Buck, and Rodrigo F Chuaqui. Comparison of snap freezing versus ethanol fixation for gene expression profiling of tissue specimens. *The Journal of molecular diagnostics*, 6(4):371–377, November 2004.

[43] RM Pickering. Digit preference in estimated gestational age. *Statistics in medicine*, 11(9):1225–1238, 1992.

[44] Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

[45] Thomas P Quinn, Ionas Erb, Mark F Richardson, and Tamsyn M Crowley. Understanding sequencing data as compositions: an outlook and review. *Bioinformatics*, 34(16):2870–2878, 2018.

[46] Arjun Raj, Patrick Van Den Bogaard, Scott A Rifkin, Alexander Van Oudenaarden, and Sanjay Tyagi. Imaging individual mrna molecules using multiple singly labeled probes. *Nature methods*, 5(10):877–879, 2008.

[47] Patricia P. Reis, Levi Waldron, Rashmi S. Goswami, Wei Xu, Yali Xuan, Bayardo Perez-Ordonez, Patrick Gullane, Jonathan Irish, Igor Jurisica, and Suzanne Kamel-Reid. mRNA transcript quantification in archival samples using multiplexed, color-coded probes. *BMC Biotechnology*, 11(1):46, May 2011.

[48] Florenza Lüder Ripoli, Annika Mohr, Susanne Conradine Hammer, Saskia Willenbrock, Marion Hewicker-Trautwein, Silvia Hennecke, Hugo Murua Escobar, and Ingo Nolte. A comparison of fresh frozen vs. formalin-fixed, paraffin-embedded specimens of canine mammary tumors via branched-DNA assay. In *International journal of molecular sciences*, 2016.

[49] Mark D. Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25, Mar 2010.

[50] Samuel G Rodriques, Robert R Stickels, Aleksandrina Goeva, Carly A Martin, Evan Murray, Charles R Vanderburg, Joshua Welch, Linlin M Chen, Fei Chen, and Evan Z Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467, 2019.

[51] Sarah Schrödter, Martin Braun, Isabella Syring, Niklas Klümper, Mario Deng, Doris Schmidt, Sven Perner, Stefan C Müller, and Jörg Ellinger. Identification of the dopamine transporter slc6a3 as a biomarker for patients with renal cell carcinoma. *Molecular cancer*, 15(1):10, 2016.

[52] Jérome Solassol, Jeanne Ramos, Evelyne Crapez, Majda Saifi, Alain Mangé, Evelyne Vianès, Pierre-Jean Lamy, Valérie Costes, and Thierry Maudelonde. Kras mutation detection in paired frozen and formalin-fixed paraffin-embedded (FFPE) colorectal cancer tissues. *International Journal of Molecular Sciences*, 12(5):3191–3204, 2011.

[53] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016.

[54] Shiquan Sun, Jiaqiang Zhu, and Xiang Zhou. Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies. *Nature methods*, 17(2):193–200, 2020.

[55] Valentine Svensson, Sarah A Teichmann, and Oliver Stegle. Spatialde: identification of spatially variable genes. *Nature methods*, 15(5):343–346, 2018.

[56] Martin A. Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.

[57] Jacques Tostain, Guorong Li, Anne Gentil-Perret, and Marc Gigante. Carbonic anhydrase 9 in clear cell renal cell carcinoma: a marker for diagnosis, prognosis and treatment. *European journal of cancer*, 46(18):3141–3148, 2010.

[58] Catalina A Vallejos, John C Marioni, and Sylvia Richardson. Basics: Bayesian analysis of single-cell sequencing data. *PLoS computational biology*, 11(6):e1004333, 2015.

[59] Catalina A Vallejos, Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C Marioni. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature Methods*, 14, May 2017.

[60] Catalina A Vallejos, Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C Marioni. Normalizing single-cell rna sequencing data: challenges and opportunities. *Nature methods*, 14(6):565, 2017.

[61] David A. van Dyk. Nesting EM Algorithms for Computational Efficiency. *Statistica Sinica*, 10(1):203–225, 2000.

[62] Sanja Vickovic, Gökcen Eraslan, Fredrik Salmén, Johanna Klughammer, Linnea Stenbeck, Denis Schapiro, Tarmo Äijö, Richard Bonneau, Ludvig Bergenstråhle, José Fernandéz Navarro, et al. High-definition spatial transcriptomics for in situ tissue profiling. *Nature methods*, 16(10):987–990, 2019.

[63] Silke Von Ahlfen, Andreas Missel, Klaus Bendrat, and Martin Schlumpberger. Determinants of rna quality from ffpe samples. *PloS one*, 2(12):e1261, 2007.

[64] Hao Wang and Daniel F Heitjan. Modeling heaping in self-reported cigarette counts. *Statistics in medicine*, 27(19):3789–3804, 2008.

[65] Magdalena B Wozniak, Florence Le Calvez-Kelm, Behnoush Abedi-Ardekani, Graham Byrnes, Geoffroy Durand, Christine Carreira, Jocelyne Michelon, Vladimir Janout, Ivana Holcatova, Lenka Foretova, et al. Integrative genome-wide gene expression profiling of clear cell renal cell carcinoma in czech republic and in the united states. *PloS one*, 8(3), 2013.

[66] David E Wright and Isabelle Bray. A mixture model for rounded data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(1):3–13, 2003.

[67] Shen Yin, Xinlei Wang, Gaoxiang Jia, and Yang Xie. Mixnorm: normalizing rna-seq data from formalin-fixed paraffin-embedded samples. *Bioinformatics*, 36(11):3401–3408, 2020.

[68] Minzhe Zhang, Thomas Sheffield, Xiaowei Zhan, Qiwei Li, Donghan M Yang, Yunguan Wang, Shidan Wang, Yang Xie, Tao Wang, and Guanghua Xiao. Spatial molecular profiling: platforms, applications and analysis tools. *Briefings in Bioinformatics*, 2020.

[69] Yan Zhou, Xiang Wan, Baoxue Zhang, and Tiejun Tong. Classifying next-generation sequencing data using a zero-inflated poisson model. *Bioinformatics*, 34(8):1329–1335, 2018.