Data-based RNA-seq Simulations by Binomial Thinning

David Gerard

Department of Mathematics and Statistics, American University, Washington, DC, 20016, USA

Abstract

With the explosion in the number of methods designed to analyze bulk and single-cell RNA-seq data, there is a growing need for approaches that assess and compare these methods. The usual technique is to compare methods on data simulated according to some theoretical model. However, as real data often exhibit violations from theoretical models, this can result in unsubstantiated claims of a method's performance. Rather than generate data from a theoretical model, in this paper we develop methods to add signal to real RNA-seq datasets. Since the resulting simulated data are not generated from an unrealistic theoretical model, they exhibit realistic (annoying) attributes of real data. This lets RNA-seq methods developers assess their procedures in non-ideal (model-violating) scenarios. Our procedures may be applied to both single-cell and bulk RNA-seq. We show that our simulation method results in more realistic datasets and can alter the conclusions of a differential expression analysis study. We also demonstrate our approach by comparing various factor analysis techniques on RNA-seq datasets. Our tools are available in the `seqgendiff` R package on the Comprehensive R Archive Network: `https://cran.r-project.org/package=seqgendiff`.

1 Introduction

Due to its higher signal-to-noise ratio, larger range of detection, and its ability to measure *a priori* unknown genes, RNA-seq has surpassed microarrays as the technology of choice to measure gene expression [Wang et al., 2009]. With the advent of single-cell RNA-seq technologies, researchers now even have the ability to explore expression variation at the individual cell level [Hwang et al., 2018]. This presents exciting opportunities for researchers to characterize the expression heterogeneity between and within organisms, and has brought about a plentiful flow of new datasets. In the wake of these new data, an explosion of methods has been developed to analyze them. In Sections 2.2, 2.3, 2.4, and 2.5 we provide a large (yet terribly incomplete) list of methods designed to analyze RNA-seq data.

The typical pipeline to evaluate a method is to first simulate data according to some theoretical model, then compare it to competing methods on these simulated data and show it to be superior in some fashion. This way of evaluation can be useful to see how a method works in ideal scenarios. However, real data rarely live in ideal scenarios. Real data often exhibit unwanted variation beyond that assumed by a model [Leek et al., 2010]. Theoretical distributional assumptions are also difficult to verify, and are sometimes mired in controversy [Svensson, 2019].

Keywords and phrases: RNA-seq, simulation, differential expression, factor analysis, confounders, scaling factors

In this paper, we propose an alternative approach. Rather than generate data with a prespecified signal according to some modeling assumptions, we take a real RNA-seq dataset and add a prespecified signal to it. The main advantage of our approach is that any unwanted variation in the real data is maintained in the simulated data, and this unwanted variation need not be prespecified by the researcher. The way we add signal does carry assumptions, but they are flexible (Supplementary Section S1.2). And we believe that this way of simulation, compared to simulating under a theoretical model, allows researchers to more realistically evaluate their methods.

This manuscript essentially generalizes the simulation techniques proposed in Gerard and Stephens [2017], Gerard and Stephens [2018], and Lu [2018]. These previous papers use binomial thinning (the approach used in this paper) in the case where there are just two groups that are differentially expressed (hereafter, the "two-group model"). These papers did not develop methods for more complicated design scenarios, they did not present user-friendly software implementations for their simulation techniques, and they did not justify their simulation techniques as broadly. Here, we allow for arbitrary experimental designs, we release software for users to implement their own simulations, and we justify our techniques using very flexible assumptions.

There has been some other previous work on "data-based" simulations in expression analyses. Datasets resulting from data-based simulations (sometimes called "plasmodes" [Mehta et al., 2004]) have been used in microarray studies before the development of RNA-seq [Nettleton et al., 2007, Gadbury et al., 2008]. All RNA-seq data-based simulation methods have so far operated in the two-group (or finite-group) model, without any ability to simulate data from arbitrary experimental designs. Rocke et al. [2015] and Sun and Stephens [2018] randomly shuffled group indicators in the two-group model, resulting in completely null data, and methods can be evaluated on their ability to control for type I error when the data are all null. Rigaill et al. [2016], in addition to generating null data by randomly shuffling group labels, incorporate multiple datasets to create some non-null genes within their simulated datasets. Benidt and Nettleton [2015] use a count-swapping algorithm in the two-group model to create differentially expressed genes when one already has two treatment groups. Kvam et al. [2012], Reeb and Steibel [2013], and van de Wiel et al. [2014] create non-null genes by multiplying counts for all individuals in a group by the fold-change in mean expression. Robinson and Storey [2014] uses a binomial distribution approach to uniformly decrease the sequencing depth of an entire dataset (but not to add differentially expressed genes). Concerning non-data-based methods, Vieth et al. [2017] and Zappia et al. [2017] use real RNA-seq data to estimate the parameters in a data-generating model before simulating data from the theoretical model using these estimated parameter values. Our work is the first to extend data-based RNA-seq simulation beyond the finite-group model.

Our paper is organized as follows. We first list the goals and assumptions of our simulation scheme (Section 2.1) before motivating it with four applications (Sections 2.2, 2.3, 2.4, and 2.5) and describing our process of simulating RNA-seq in detail (Section 2.6). We then demonstrate how our approach can more accurately preserve structure in a real dataset compared to simulating a dataset from a theoretical model (Section 3.1). We show that this can alter the conclusions of a differential expression analysis simulation study (Section 3.2). We then apply our simulation approach by comparing five factor analysis methods using the GTEx data [GTEx Consortium, 2017] (Section 3.3). We finish with a discussion and conclusions (Sections 4 and 5).

We adopt the following notation. We denote matrices by bold uppercase letters ($\boldsymbol{A}$), vectors by bold lowercase letters ($\boldsymbol{a}$), and scalars by non-bold letters (a or A). Indices typically run from 1 to their uppercase version, e.g. $a = 1, 2, \ldots, A$. Where there is no chance for confusion, we let

non-bold versions of letters represent the scalar elements of matrices and vectors. So a_{ij} is the (i,j)th element of $\boldsymbol{A}$, while a_i is the ith element of $\boldsymbol{a}$. We let $\mathbf{1}_A$ denote the A-vector of 1's and $\mathbf{1}_{A\times B}$ the $A \times B$ matrix of 1's. The matrix transpose is denoted by $\boldsymbol{A}^\intercal$.

2 Methods

2.1 Goals and Assumptions

We will now describe the goals and assumptions of our simulation method, which relies on a researcher having access to a real RNA-seq dataset. Suppose a researcher has a matrix $\boldsymbol{Y} \in \mathbb{R}^{G\times N}$ of RNA-seq read-counts for G genes and N individuals. Also suppose a researcher has access to a design matrix $\boldsymbol{X}_1 \in \mathbb{R}^{N\times P_1}$ with P_1 variables. We assume the RNA-seq counts, $\boldsymbol{Y}$, are generated according to the following model:

$$\begin{aligned} y_{gn} &\sim \text{Poisson}(2^{\theta_{gn}}), \text{ and} \\ \boldsymbol{\Theta} &= \boldsymbol{\mu}\mathbf{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}, \end{aligned} \tag{1}$$

where

- $\boldsymbol{\mu} \in \mathbb{R}^G$ is a vector of intercept terms for the genes,
- $\boldsymbol{B}_1 \in \mathbb{R}^{G\times P_1}$ is the corresponding coefficient matrix of $\boldsymbol{X}_1$,
- $\boldsymbol{Z} \in \mathbb{R}^{N\times K}$ is a matrix of unobserved surrogate variables,
- $\boldsymbol{A} \in \mathbb{R}^{G\times K}$ is the corresponding coefficient matrix of $\boldsymbol{Z}$, and
- $\boldsymbol{\Omega} \in \mathbb{R}^{G\times N}$ represents all other unwanted variation not accommodated by the other terms in the model,

where $\boldsymbol{\mu}$, $\boldsymbol{B}_1$, $\boldsymbol{Z}$, $\boldsymbol{A}$, and $\boldsymbol{\Omega}$ are all unknown. Given the above data-generating process, suppose a user provides the following (known) elements:

- $\boldsymbol{X}_2 \in \mathbb{R}^{N\times P_2}$, a design matrix with fixed-rows (see note 3 below),
- $\boldsymbol{B}_2 \in \mathbb{R}^{G\times P_2}$, the coefficient matrix corresponding to $\boldsymbol{X}_2$,
- $\boldsymbol{X}_3 \in \mathbb{R}^{N\times P_3}$, a design matrix with rows that can be permuted (see note 3 below), and
- $\boldsymbol{B}_3 \in \mathbb{R}^{G\times P_3}$, the coefficient matrix corresponding to $\boldsymbol{X}_3$.

Our goal is to generate a matrix $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{G\times N}$ from $\boldsymbol{Y}$ such that

$$\begin{aligned} \tilde{y}_{gn} &\sim \text{Poisson}(2^{\tilde{\theta}_{gn}}), \text{ and} \\ \tilde{\boldsymbol{\Theta}} &= \tilde{\boldsymbol{\mu}}\mathbf{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{B}_2\boldsymbol{X}_2^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}, \end{aligned} \tag{2}$$

where

- $\boldsymbol{\Pi} \in \mathbb{R}^{N\times N}$ is a random permutation matrix, whose distribution controls the level of association between the columns of $\boldsymbol{\Pi}\boldsymbol{X}_3$ and the columns of $\boldsymbol{Z}$, and
- $\tilde{\boldsymbol{\mu}}$ is a new vector of intercept terms for the genes.

We will provide the details on how to generate $\tilde{\boldsymbol{Y}}$ from $\boldsymbol{Y}$ in Section 2.6. But we would like to first provide some notes below, and then discuss the applications of being able to generate (2) from (1).

Note 1: For simplicity we use the Poisson distribution in the main text (equations (1) and (2)). However, our approach is valid under much more general assumptions. In particular, we note that if the counts were generated according to a negative binomial distribution, a zero-inflated negative binomial distribution, or even a mixture of binomials and negative binomials, then our simulation scheme still preserves the structure of the data (Supplementary Section S1.2). However, even when our general modeling assumptions are violated, one can show (via the law of total expectation) that if $\log_2(E[\boldsymbol{Y}]) = \boldsymbol{\Theta}$, then $\log_2(E[\tilde{\boldsymbol{Y}}]) = \tilde{\boldsymbol{\Theta}}$, where we are taking element-wise logarithms of $\mathrm{E}[\boldsymbol{Y}]$ and $\mathrm{E}[\tilde{\boldsymbol{Y}}]$. Thus, our procedure will produce the correct mean $\log_2$-fold change in the new dataset, but the resulting mean/variance relationship might not be as assumed.

Note 2: The $\boldsymbol{\Omega}$ term in (1) and (2) represents the realistic and annoying features of the data. In ideal situations, $\boldsymbol{\Omega} = \boldsymbol{0}_{G\times N}$. However, most datasets likely include non-zero $\boldsymbol{\Omega}$, and so assessing a method's ability to be robust in the presence of $\boldsymbol{\Omega}$, without the researcher having to prespecify $\boldsymbol{\Omega}$, is the key strength of our simulation approach.

Note 3: As described below, we include both $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ in (2) to control different aspects of a simulation study. One may control the level of association between the columns of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ as these are both observed and fixed by the user. The inclusion of $\boldsymbol{X}_3$ and $\boldsymbol{\Pi}$ allows us to try to control the level of association between $\boldsymbol{\Pi}\boldsymbol{X}_3$ and $\boldsymbol{Z}$.

Before we discuss obtaining (2) from (1), we point out four potential applications of this simulation approach: (i) evaluating differential expression analyses (Section 2.2), (ii) evaluating confounder adjustment approaches (Section 2.3), (iii) evaluating the effects of library size heterogeneity on differential expression analyses (Section 2.4), and (iv) evaluating factor analysis methods (Section 2.5).

2.2 Application: Evaluating Differential Expression Analysis

One of the more common applications of RNA-seq data is estimating and testing for differences in gene expression between two groups. Many packages and techniques exist to perform this task [Robinson and Smyth, 2007b, Hardcastle and Kelly, 2010, Van De Wiel et al., 2012, Kharchenko et al., 2014, Law et al., 2014, Love et al., 2014, Finak et al., 2015, Guo et al., 2015, Nabavi et al., 2015, Delmans and Hemberg, 2016, Korthauer et al., 2016, Costa-Silva et al., 2017, Qiu et al., 2017, Miao et al., 2018, Risso et al., 2018, Van den Berge et al., 2018, Wang and Nabavi, 2018, Wang et al., 2019, among others], and so developing approaches and software to compare these different software packages would be of great utility to the scientific community. Generating data from the two-group model is a special case of (1) and (2), where

$$\boldsymbol{\Theta} = \boldsymbol{\mu}\boldsymbol{1}_N^\intercal + \boldsymbol{\Omega}, \tag{3}$$

$$\tilde{\boldsymbol{\Theta}} = \tilde{\boldsymbol{\mu}}\boldsymbol{1}_N^\intercal + \boldsymbol{b}\boldsymbol{x}^\intercal\boldsymbol{\Pi}^\intercal + \boldsymbol{\Omega}, \tag{4}$$

and $\boldsymbol{\Pi}\boldsymbol{x} \in \mathbb{R}^N$ contains a single indicator variable, indicating membership to one of two groups. Researchers may specify $\boldsymbol{b}$ and $\boldsymbol{x}$ and evaluate a method's ability to (i) estimate $\boldsymbol{b}$ and (ii) detect which genes have non-zero b_g.

In many settings, a researcher may want to specify the distribution of the b_g's. Our software implementation allows for this. In addition, following Stephens [2016], we allow researchers to specify the distribution of b_g/s_g^α, where s_g is the sample standard deviation of the gth row of $\log_2(\boldsymbol{Y} + 0.5)$, and α is a user-specified constant. Allowing for $\alpha = 0$ corresponds to the scenario

of specifying the distribution of the effects, while allowing for $\alpha = 1$ corresponds to specifying the p-value prior of Wakefield [2009].

Though the two-group model is perhaps the most common scenario in differential expression analysis, our method also allows for arbitrary design matrices. Such design matrices have applications in many types of expression experiments [Smyth, 2004, McCarthy et al., 2012, Van De Wiel et al., 2012, Tang et al., 2015], and so the ability to simulate arbitrary designs gives researchers another tool to evaluate their methods in more complicated scenarios.

2.3 Application: Evaluating Confounder Adjustment

Unobserved confounding / batch effects / surrogate variables / unwanted variation has been recognized as a serious impediment to scientific studies in the modern "omics" era [Leek et al., 2010]. As such, there is a large literature on accounting for unwanted variation, particularly in RNA-seq studies [Leek and Storey, 2007, Carvalho et al., 2008, Kang et al., 2008a,b, Leek and Storey, 2008, Stegle et al., 2008, Friguet et al., 2009, Kang et al., 2010, Listgarten et al., 2010, Stegle et al., 2010, Wu and Aryee, 2010, Fusi et al., 2012, Gagnon-Bartsch and Speed, 2012, Stegle et al., 2012, Sun et al., 2012, Gagnon-Bartsch et al., 2013, Mostafavi et al., 2013, Yang et al., 2013, Leek, 2014, Risso et al., 2014, Perry and Pillai, 2015, Chen and Zhou, 2017, Gerard and Stephens, 2017, Lee et al., 2017, Wang et al., 2017, Caye et al., 2018, Gerard and Stephens, 2018, Hung, 2018, McKennan and Nicolae, 2018a,b, among others]. The glut of available methods indicates a need to realistically compare these methods.

Typically, the form and strength of any unobserved confounding is not known. So one way to assess different confounder adjustment methods would be to assume model (1) and add signal to the data resulting in the following submodel of (2):

$$\tilde{\boldsymbol{\Theta}} = \tilde{\boldsymbol{\mu}}\mathbf{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}. \tag{5}$$

A researcher would then explore how close a method's estimate of $\boldsymbol{B}_3$ is to the truth (assuming the researcher may use both $\boldsymbol{X}_1$ and $\boldsymbol{\Pi}\boldsymbol{X}_3$ to obtain this estimate). The researcher can control the correlation between the columns of $\boldsymbol{\Pi}\boldsymbol{X}_3$ and the columns of $\boldsymbol{Z}$ by specifying the distribution of $\boldsymbol{\Pi}$ (as described in Section 2.6). Intuitively, the stronger the correlation between the columns of $\boldsymbol{X}_3$ and the columns of $\boldsymbol{Z}$, the more difficult the confounder adjustment problem. This approach was used in the two-group model in Gerard and Stephens [2017] and [2018], but not for general design matrices.

2.4 Application: Evaluating Effects of Library Size Heterogeneity

"Library size" corresponds to the number of reads an individual sample contains. Adjusting for library size is surprisingly subtle and difficult, and thus many techniques have been proposed to perform this adjustment [Anders and Huber, 2010, Bullard et al., 2010, Robinson and Oshlack, 2010, Langmead et al., 2010, Dillies et al., 2012]. The most commonly-used techniques can be viewed as a form of confounder adjustment [Gerard and Stephens, 2017]. For most methods, this form of confounder adjustment corresponds to setting one column of $\boldsymbol{A}$ in (1) to be $\mathbf{1}_G$ and estimating the corresponding column in $\boldsymbol{Z}$ using some robust method that assumes that the majority of genes are non-differentially expressed.

One way to evaluate the performance of a library size adjustment procedure is to see how effect

size estimates change when the samples are thinned, changing the library size. First, assume we are operating in the following submodel of (1):

$$\boldsymbol{\Theta} = \boldsymbol{\mu}\mathbf{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \mathbf{1}_G\boldsymbol{z}^\intercal + \boldsymbol{\Omega}. \tag{6}$$

A researcher may specify (i) additional signal and (ii) a further amount of thinning on each sample by generating the following submodel of (2):

$$\tilde{\boldsymbol{\Theta}} = \tilde{\boldsymbol{\mu}}\mathbf{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal + \mathbf{1}_G\boldsymbol{x}_2^\intercal + \mathbf{1}_G\boldsymbol{x}_3^\intercal\boldsymbol{\Pi}^\intercal + \mathbf{1}_G\boldsymbol{z}^\intercal + \boldsymbol{\Omega} \tag{7}$$

$$= \tilde{\boldsymbol{\mu}}\mathbf{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal + \mathbf{1}_G(\boldsymbol{z} + \boldsymbol{x}_2 + \boldsymbol{\Pi}\boldsymbol{x}_3)^\intercal + \boldsymbol{\Omega}. \tag{8}$$

To evaluate the effectiveness of a library size adjustment procedure, researchers may observe the effects on the estimates of $\boldsymbol{B}_3$ under various amounts of library thinning (controlled by altering $\boldsymbol{x}_2$ and $\boldsymbol{x_3}$).

2.5 Application: Evaluating Factor Analysis

Factor analysis is a fundamental technique in every statistician's arsenal. Since its creation by Spearman [Spearman, 1904], literally hundreds of factor analysis / matrix decomposition / matrix factorization approaches have been developed, and new approaches are created each year to account for new features of new data [Hotelling, 1933, Eckart and Young, 1936, Comon, 1994, Tipping and Bishop, 1999, Lee and Seung, 1999, Hyvärinen and Oja, 2000, West, 2003, Zou et al., 2006, Hoff, 2007, Salakhutdinov and Mnih, 2008, Ghosh and Dunson, 2009, Witten et al., 2009, Engelhardt and Stephens, 2010, Stegle et al., 2010, Mayrink and Lucas, 2013, Yang et al., 2014, Josse and Wager, 2016, Leung and Drton, 2016, Wang and Stephens, 2018, to name a very few]. For RNA-seq, factor analysis methods have found applications in accounting for unwanted variation [Leek, 2014, Risso et al., 2014], estimating cell-cycle state [Buettner et al., 2015, Scialdone et al., 2015], and general quality assessments [Love et al., 2014]. Thus, techniques to realistically compare various factor analysis methods would be of great use to the scientific community. We demonstrate in this section how our simulation approaches can be used to evaluate factor analysis methods applied to RNA-seq.

We suppose that the RNA-seq read-counts follow the following submodel of (1):

$$\boldsymbol{\Theta} = \boldsymbol{\mu}\mathbf{1}_N^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}. \tag{9}$$

We then suppose that the researcher generates a modified dataset that follows the following submodel of (2):

$$\tilde{\boldsymbol{\Theta}} = \tilde{\boldsymbol{\mu}}\mathbf{1}_N^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}. \tag{10}$$

We assume that a researcher applies a factor analysis to (10) to estimate a low-rank matrix with $K + P_3$ factors. That is, the researcher fits the following model,

$$\log_2(\mathrm{E}[\tilde{Y}]) = \boldsymbol{\mu}\mathbf{1}_N^\intercal + \boldsymbol{L}\boldsymbol{F}^\intercal, \tag{11}$$

with factor matrix $\boldsymbol{F} \in \mathbb{R}^{N\times(K+P_3)}$ and loading matrix $\boldsymbol{L} \in \mathbb{R}^{G\times(K+P_3)}$, obtaining estimates $\hat{\boldsymbol{L}}$ and $\hat{\boldsymbol{F}}$. These estimates are obtained without using $\boldsymbol{\Pi}\boldsymbol{X}_3$. A researcher may evaluate their factor

analysis by

1. Assessing if any of the columns of $\hat{\boldsymbol{F}}$ are close to the columns of $\boldsymbol{\Pi X}_3$,
2. Assessing if any of the columns of $\hat{\boldsymbol{L}}$ are close to the columns of $\boldsymbol{B}_3$, and
3. Assessing if the column-space of $\boldsymbol{\Pi X_3}$ is close to the column-space of $\hat{\boldsymbol{F}}$, which would be an important consideration in downstream regression analyses [Leek and Storey, 2007, e.g.].

In a factor analysis, the factors and loadings are only identifiable after imposing assumptions on their structure (such as sparsity or orthogonality). Thus, researchers may vary the structure of $\boldsymbol{B}_3$ and $\boldsymbol{\Pi X}_3$ and observe the robustness of their factor analysis methods to violations of their structural assumptions.

2.6 Generating Modified RNA-seq Data

We will now discuss the approach of obtaining (2) from (1). We will use the following well-known fact of the Poisson distribution, which may be found in many elementary probability texts:

Lemma 1. *If* $y \sim \text{Poisson}(a)$ *and* $\tilde{y}|y \sim \text{Bin}(y, b)$, *then* $\tilde{y} \sim \text{Poisson}(ab)$.

In the case when $\boldsymbol{\Pi}$ is drawn uniformly from the space of permutation matrices, we have the simplified procedure described in Procedure 1. The validity of Procedure 1 follows directly from the modeling assumptions in (1) and Lemma 1. Since $y_{gn} \sim \text{Poisson}(2^{\theta_{gn}})$ and $\tilde{y}_{gn}|y_{gn} \sim \text{Bin}(y_{gn}, 2^{q_{gn}})$, we have that $\tilde{y}_{gn} \sim \text{Poisson}(2^{\theta_{gn}+q_{gn}})$. If we set $\tilde{\theta}_{gn} = \theta_{gn} + q_{gn}$, then we have

$$\begin{aligned}
\tilde{\boldsymbol{\Theta}} &= \boldsymbol{\Theta} + \boldsymbol{Q} && (12)\\
&= (\boldsymbol{\mu}\boldsymbol{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}) + (\boldsymbol{B}_2\boldsymbol{X}_2^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal - \boldsymbol{e}\boldsymbol{1}_N^\intercal) && (13)\\
&= (\boldsymbol{\mu} - \boldsymbol{e})\boldsymbol{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{B}_2\boldsymbol{X}_2^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega} && (14)\\
&= \tilde{\boldsymbol{\mu}}\boldsymbol{1}_N^\intercal + \boldsymbol{B}_1\boldsymbol{X}_1^\intercal + \boldsymbol{B}_2\boldsymbol{X}_2^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}. && (15)
\end{aligned}$$

Equation (13) follows from the definition of $\boldsymbol{\Theta}$ from (1) and the definition of $\boldsymbol{Q}$ from Step 4 of Procedure 1. Equation (15) follows by setting $\tilde{\boldsymbol{\mu}}$ to be $\boldsymbol{\mu} - \boldsymbol{e}$.

Procedure 1 Basic procedure to generate (2) from (1) when the permuted design matrix ($\boldsymbol{\Pi X}_3$) is independent of the surrogate variables.

Input: $\boldsymbol{Y}$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$, $\boldsymbol{B}_2$, $\boldsymbol{B}_3$.

1: Draw $\boldsymbol{\Pi}$ uniformly from the space of $N \times N$ permutation matrices.
2: Let $\boldsymbol{\Lambda} = \boldsymbol{B}_2\boldsymbol{X}_2^\intercal + \boldsymbol{B}_3\boldsymbol{X}_3^\intercal\boldsymbol{\Pi}^\intercal$.
3: Let $\boldsymbol{e} \in \mathbb{R}^G$ contain the row-wise maximums of $\boldsymbol{\Lambda}$. Thus, $e_g = \max(\lambda_{g1}, \ldots, \lambda_{gN})$.
4: Let $\boldsymbol{Q} = \boldsymbol{\Lambda} - \boldsymbol{e}\boldsymbol{1}_N^\intercal$.
5: Draw $\tilde{y}_{gn}|y_{gn} \sim \text{Bin}(y_{gn}, 2^{q_{gn}})$.

Output: $\tilde{\boldsymbol{Y}}$, $\boldsymbol{\Pi}$.

There are two main reasons to subtract the row-wise maximum from each row in Step 4 of Procedure 1: (i) this ensures that the binomial probabilities ($2^{q_{gn}}$) are always between 0 and 1, and (ii) this allows for minimal count-thinning while still obtaining our goal of (2). That is, the binomial probabilities will all be between 0 and 1, but they will be as close to 1 as possible while still yielding (2), thereby reducing the amount of discarded counts.

The main disadvantage to Procedure 1 is that the surrogate variables ($\boldsymbol{Z}$) will be independent of the user-specified covariates ($\boldsymbol{\Pi X}_3$). To allow the user to control the level of association between the surrogate variables and the user-provided variables, we propose using Procedure 2 to choose $\boldsymbol{\Pi}$, rather than drawing $\boldsymbol{\Pi}$ uniformly from the space of permutation matrices. In brief, the user specifies a "target correlation" matrix, $\boldsymbol{R} \in \mathbb{R}^{P_3 \times K}$, where r_{ik} is what the user desires to be the correlation between the ith column of $\boldsymbol{\Pi X}_3$ and the kth column of $\boldsymbol{Z}$. We then estimate the surrogate variables either using a factor analysis (such as the truncated singular value decomposition) or surrogate variable analysis [Leek and Storey, 2007, 2008]. We then draw a new random matrix $\boldsymbol{U} \in \mathbb{R}^{N \times P_3}$ from a conditional normal distribution assuming that each row of $\boldsymbol{U}$ and $\boldsymbol{Z}$ is jointly normal with covariance matrix (16), thus the correlation between the columns of $\boldsymbol{U}$ and $\boldsymbol{Z}$ will be approximately $\boldsymbol{R}$. We then match the rows of $\boldsymbol{X}_3$ with the rows of $\boldsymbol{U}$ using the pair-wise matching algorithm of Hansen and Klopfer [2006], though our software provides other options to match pairs via either the Gale-Shapley algorithm [Gale and Shapley, 1962] or the Hungarian algorithm [Kuhn, 1955]. This ensures that $\boldsymbol{\Pi X}_3$ is as close to $\boldsymbol{U}$ as possible. We denote the permutation matrix that matches the rows of $\boldsymbol{X}_3$ with the rows of $\boldsymbol{U}$ by $\boldsymbol{\Pi}$.

Procedure 2 Procedure to draw a permutation matrix such that the surrogate variables are correlated with the permuted design matrix.

Input: $\boldsymbol{Y}$, $\boldsymbol{X}_1$, $\boldsymbol{X}_3$, $\boldsymbol{R}$, and K.

1: Let $\boldsymbol{A} \in \mathbb{R}^{P_3 \times P_3}$ be the empirical correlation matrix between the columns of $\boldsymbol{X}_3$.

2: Adjust $\boldsymbol{R}$ by Procedure 3.

3: Estimate $\boldsymbol{Z} \in \mathbb{R}^{G \times K}$ in one of two ways:

   i. By surrogate variable analysis [Leek and Storey, 2007, 2008, Leek, 2014], using $(\boldsymbol{1}_N, \boldsymbol{X}_1)$ as the design matrix and $\boldsymbol{1}_N$ as the null design matrix.

   ii. By a factor analysis on the residuals of a regression of $\log_2(\boldsymbol{Y} + 0.5)$ on $(\boldsymbol{1}_N, \boldsymbol{X}_1)$.

   Call the centered and scaled estimates of the surrogate variables (so that the columns each have mean 0 and variance 1) $\hat{\boldsymbol{Z}}$.

4: Draw the rows of $\boldsymbol{U} \in \mathbb{R}^{N \times P_3}$ from a conditional normal distribution, assuming the nth rows of $\boldsymbol{U}$ and $\hat{\boldsymbol{Z}}$ are jointly $\mathrm{N}(\boldsymbol{0}_{P_3+K}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{A} & \boldsymbol{R} \\ \boldsymbol{R}^\intercal & \boldsymbol{I}_K \end{pmatrix} \tag{16}$$

5: Match the rows of the centered and scaled matrix $\boldsymbol{X}_3$ with the rows of the centered and scaled matrix $\boldsymbol{U}$ by pair-matching [Hansen and Klopfer, 2006] using Euclidean distance. Call the resulting permutation matrix $\boldsymbol{\Pi}$, such that row i of $\boldsymbol{\Pi X}_3$ matches with row i of $\boldsymbol{U}$.

Output: $\boldsymbol{\Pi}$.

The resulting covariance matrix (16) used in Procedure 2 is not guaranteed to be positive semi-definite. Rather than demand the user specify an appropriate target correlation matrix (which might be in general difficult for the typical user), we modify the target correlation matrix using Procedure 3 to iteratively shrink $\boldsymbol{R}$ until the Schur complement condition for positive semi-definiteness [Zhang, 2006] is satisfied.

Procedure 2 is a compromise between letting the user specify the full design matrix $\boldsymbol{X}_3$ and letting the user specify the correlation between the columns of $\boldsymbol{\Pi X}_3$ and $\boldsymbol{Z}$. A user might want to

Procedure 3 Procedure to scale the target correlation matrix so that the overall correlation matrix is positive semi-definite.

Input: $\boldsymbol{A}$, $\boldsymbol{R}$, and $\epsilon \in (0, 1]$.

1: Let ℓ be the smallest eigenvalue of $\boldsymbol{A} - \boldsymbol{R}\boldsymbol{R}^\intercal$.
2: Set $\tau = 1$.
3: **while** $\ell < 0$ **do**
4: $\quad \tau \leftarrow \max(\tau - \epsilon, 0)$.
5: $\quad$ Let ℓ be the smallest eigenvalue of $\boldsymbol{A} - \tau\boldsymbol{R}\boldsymbol{R}^\intercal$.
6: **end while**

Output: $\sqrt{\tau}\boldsymbol{R}$.

specify the correlation between $\boldsymbol{\Pi}\boldsymbol{X}_3$ and $\boldsymbol{Z}$ to evaluate factor analyses in the presence of correlated factors (Section 2.5), or to evaluate how well confounder adjustment approaches cope in the presence of correlated confounders (Section 2.3). In the simple case when $\boldsymbol{X}_3$ and $\hat{\boldsymbol{Z}}$ are drawn from a normal distribution, Procedure 2 will permute the rows of $\boldsymbol{X}_3$ so that $\boldsymbol{\Pi}\boldsymbol{X}_3$ and $\hat{\boldsymbol{Z}}$ consistently has the correct correlation structure (Theorem 1). However, for general design matrices this will not be the case. Procedure 4 (implemented in our software) provides a Monte Carlo algorithm to estimate the true correlation given the target correlation. Basically, the estimator approximates the expected value (conditional on $\hat{\boldsymbol{Z}}$) of the Pearson correlations between the columns of $\boldsymbol{\Pi}\boldsymbol{X}_3$ and the columns of $\hat{\boldsymbol{Z}}$. We justify this in an intuitive way by the law of total expectation. Consider $\boldsymbol{x}$ a single column of $\boldsymbol{\Pi}\boldsymbol{X}_3$ with empirical mean and standard deviation of $\bar{x}$ and s_x. Similarly consider $\boldsymbol{z}$ a single column of $\hat{\boldsymbol{Z}}$ with empirical mean and standard deviation of $\bar{z}$ and s_z. Then

$$\text{cor}(x_n, z_n) \approx \text{E}\left[\sum_{n=1}^{N} \frac{(x_n - \bar{x})(z_n - \bar{z})}{s_x s_z}\right] = \text{E}\left[\text{E}\left[\sum_{n=1}^{N} \frac{(x_n - \bar{x})(z_n - \bar{z})}{s_x s_z} | \boldsymbol{z}\right]\right]. \tag{17}$$

The estimator in Procedure 4 is a Monte Carlo approximation to the internal expectation in (17). We explore this correlation estimator through simulation in Supplementary Section S2.1.

Procedure 4 Monte Carlo procedure to estimate the true correlation matrix given the target correlation matrix.

Input: $\boldsymbol{Z}$, $\boldsymbol{X}_3$, $\boldsymbol{\Sigma}$, and $B \in \mathbb{N}$.

1: **for** b in $1, 2, \ldots, B$ **do**
2: $\quad$ Draw $\boldsymbol{U}$ as in Step 4 of Procedure 2.
3: $\quad$ Derive $\boldsymbol{\Pi}$ as in Step 5 of Procedure 2.
4: $\quad$ Set $\boldsymbol{R}_b \in \mathbb{R}^{P_3 \times K}$ to be the Pearson correlation matrix between the columns of $\boldsymbol{\Pi}\boldsymbol{X}_3$ and $\boldsymbol{Z}$.
5: **end for**
6: Set $\hat{\boldsymbol{R}} = (\boldsymbol{R}_1 + \cdots + \boldsymbol{R}_B)/B$.

Output: $\hat{\boldsymbol{R}}$.

3 Results

3.1 Features of Real Data

Real data exhibit characteristics that are difficult to capture by simulations. In this section, we demonstrate how our binomial thinning approach maintains these features, while simulating from a theoretical model results in unrealistic simulated RNA-seq data.

We took the GTEx muscle data [GTEx Consortium, 2017], and filtered out all genes with a mean read-depth of less than 10 reads. This resulted in a dataset containing 18,204 genes and 564 individuals. We then randomly assigned half of the individuals to one group and half to the other group, and used our `seqgendiff` software to add a $N(0, 0.8^2)$ $\log_2$-fold-change between groups to 25% of the genes. We similarly used the `powsimR` software [Vieth et al., 2017] to generate data according to a theoretical negative binomial model (with parameters estimated from the GTEx muscle data), again by adding a $N(0, 0.8^2)$ $\log_2$-fold-change between the two groups in 25% of the genes. The results below are from one simulation, but the results are robust and consistent across many datasets. The reader is encouraged to change the random seed in our code to explore the robustness of our conclusions.

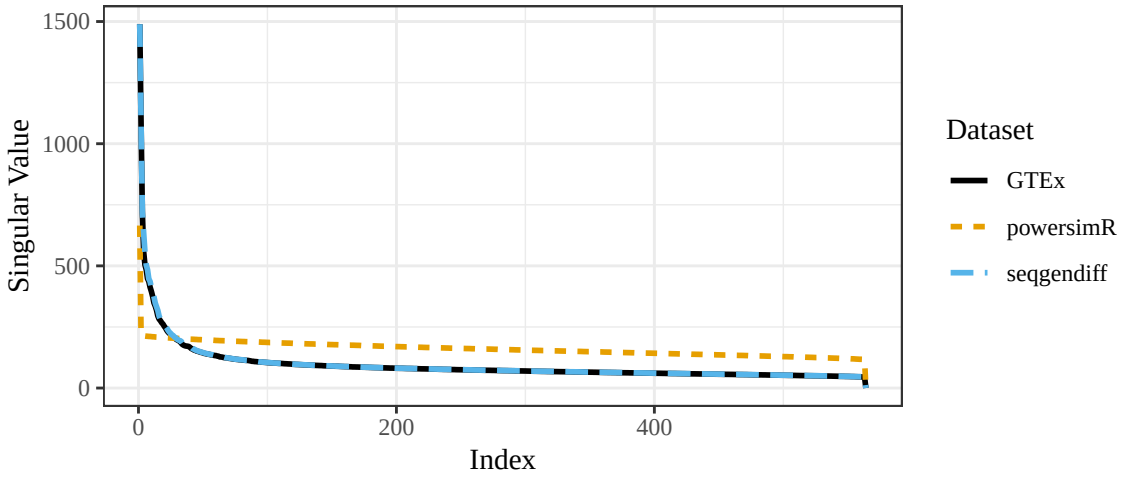

Figure 1: Scree plots for the GTEx dataset (black), `powsimR` dataset (orange), and the `seqgendiff` dataset (blue). The singular values for the GTEx and `seqgendiff` datasets are almost identical.

The structure of the `powsimR` dataset is very different from that observed in the `seqgendiff` and GTEx datasets. There seems to be more zeros in the `powsimR` dataset than in the `seqgendiff` and GTEx datasets (Supplementary Figure S2), even though we simulated the `powsimR` dataset under the negative binomial setting and not the zero-inflated negative binomial setting. Scree plots of the three datasets show that there are a lot more small factors influencing variation in the `seqgendiff` and GTEx datasets than in the `powsimR` dataset (Figure 1). The main source of variation in the `powsimR` dataset comes from the group membership, while other (unwanted) effects dominate the variation in the `seqgendiff` dataset (Figure 2). It is only the fourth principle component in the `seqgendiff` dataset that seems to capture the group membership (Supplementary Figure S3). Though this unwanted variation exists, with such a large sample size voom-limma can accurately estimate the effects (Supplementary Figure S4). The voom plots (visualizing the mean-variance

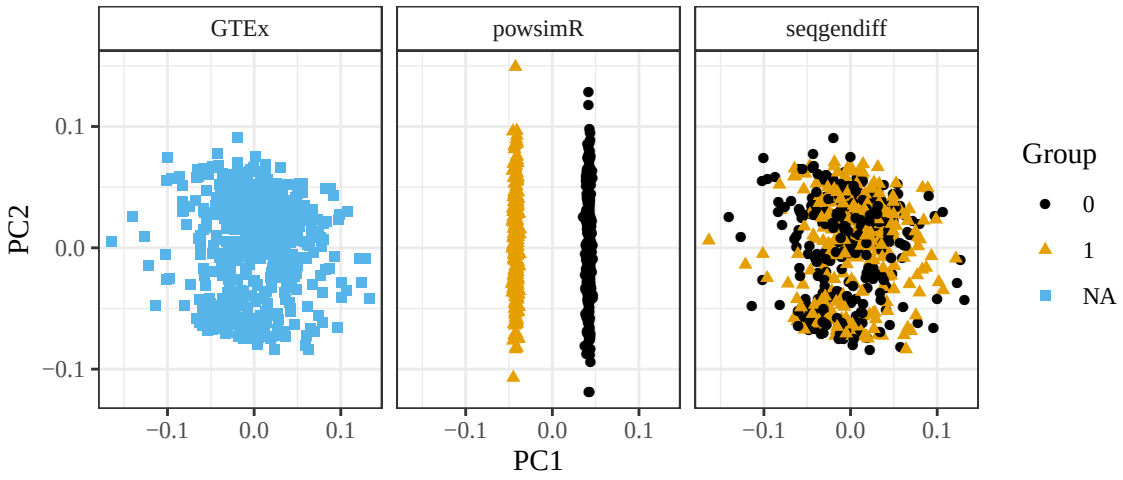

Figure 2: First and second principle components for the GTEx dataset (left), the `powsimR` dataset (center), and the `seqgendiff` dataset (right).

trend [Law et al., 2014]) are about the same in the GTEx and `seqgendiff` data, but the distribution of the square-root standard deviations appears more symmetric in the `powsimR` dataset (Figure 3). There is also an uncharacteristic hook in the mean-variance trend in the `powsimR` dataset for low-counts. These visualizations indicate that `seqgendiff` can generate more realistic datasets for RNA-seq simulation.

3.2 Effects on Differential Expression Analysis Simulations

The differences in real versus simulated data (as discussed in Section 3.1) have real implications when evaluating methods in simulation studies. To demonstrate this, we used the GTEx muscle data to simulate RNA-seq data from the two-group model as in Section 3.1. We did this for $N = 10$ individuals, $G = 10{,}000$ genes, setting 90% of the genes to be null, and generating the $\log_2$-fold change from a $N(0, 0.8^2)$ distribution for the non-null genes. We simulated 500 datasets this way using both `seqgendiff` and `powsimR`. Each replication, we applied DESeq2 [Love et al., 2014], edgeR [Robinson et al., 2009], and voom-limma [Law et al., 2014] to the simulated datasets. We evaluated the methods based on (i) false discovery proportion when using Benjamini-Hochberg [Benjamini and Hochberg, 1995] to control false discovery rate at the 0.05 level, (ii) power to detect non-null effects based on a 0.05 false discovery rate control threshold, and (iii) mean squared error of the estimates.

We wanted to make sure that the datasets generated from `powsimR` and `seqgendiff` were comparable, so we measured the proportion of variance explained (PVE) by the group membership for each gene, which we define as

$$V(\mathbf{\Pi} \boldsymbol{x}_3 b_{3g}) / V(\log_2(\tilde{\boldsymbol{y}}_g + 0.5)), \tag{18}$$

where b_{3g} is $\log_2$-fold change for gene g, $\tilde{\boldsymbol{y}}_g \in \mathbb{R}^N$ is the gth row of $\tilde{\boldsymbol{Y}}$, and $V(\cdot)$ returns the empirical variance of a vector. When we looked at the median (over the non-null genes) PVE across the datasets, the `seqgendiff` datasets and `powsimR` datasets had the same median PVE on

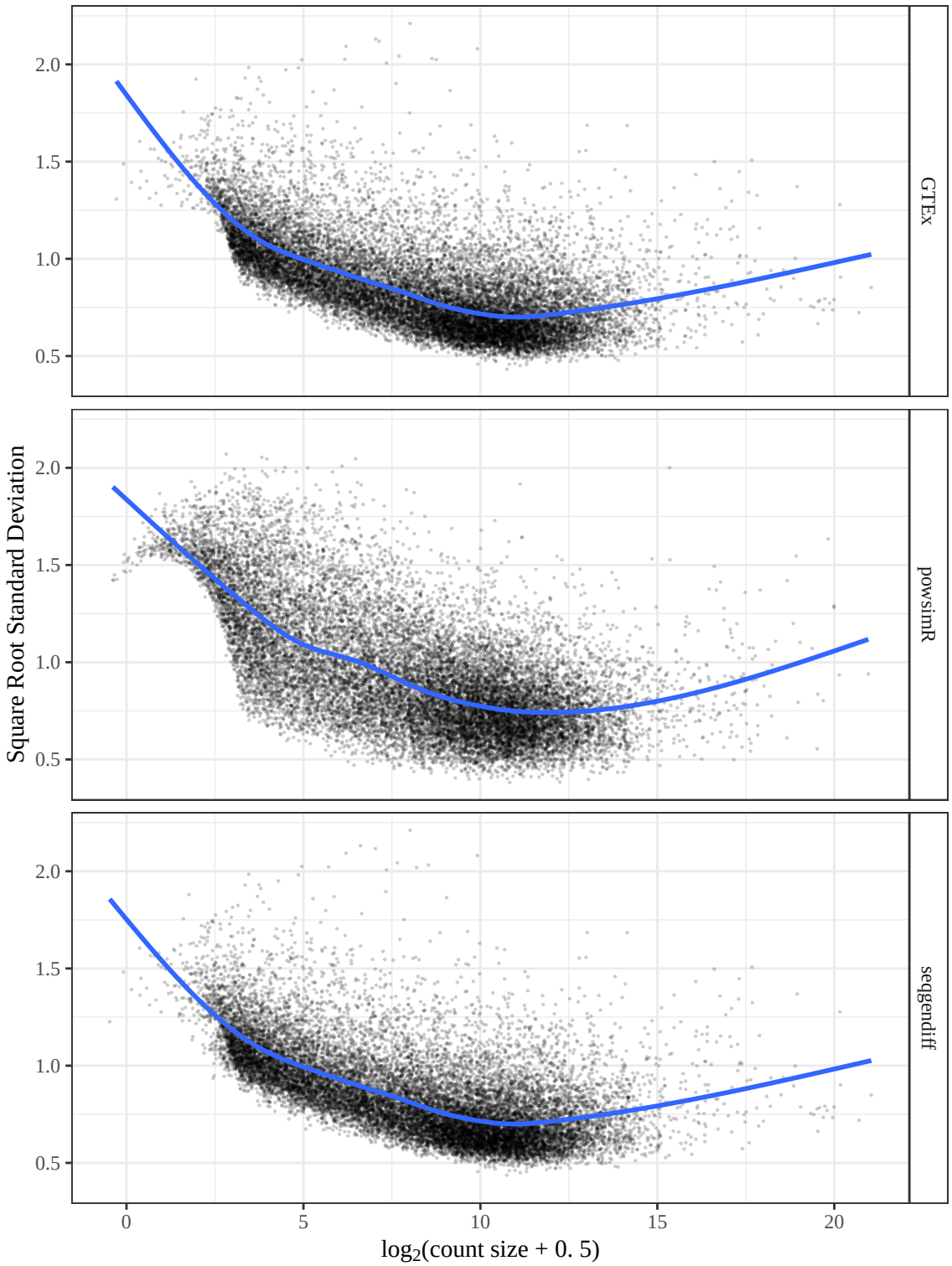

Figure 3: Voom plots [Law et al., 2014] visualizing the mean-variance trend in RNA-seq datasets. The voom plots are visually similar for the GTEx and `seqgendiff` datasets. The `powsimR` dataset has an uncharacteristic hook near the low counts in its voom plot.

average, though there was higher variability in the median PVE among the `seqgendiff` datasets (Supplementary Figure S5).

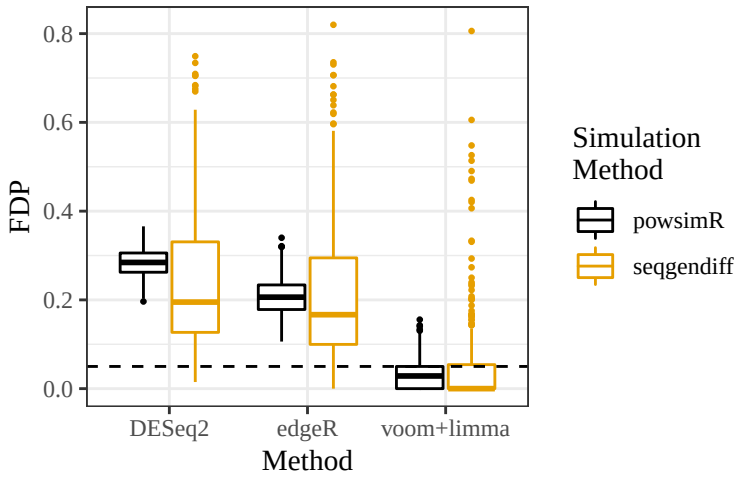

Figure 4: Boxplots of false discovery proportion (FDP) (y-axis) for various differential expression analysis methods (x-axis) when applied on different simulated datasets (color). Benjamini-Hochberg was used to control for false discovery rate at the 0.05 level (horizontal dashed line).

Boxplots of the false discovery proportion for each method in each dataset can be found in Figure 4. Both the `powsimR` and `seqgendiff` datasets indicate that only voom-limma can control false discovery rate adequately at the nominal level. However, the results based on the `seqgendiff` datasets indicate that there is a lot more variability in false discovery proportion than indicated by the `powsimR` datasets. In particular, it does not seem uncommon for `seqgendiff` to generate datasets with false discovery proportions well above the nominal rate. If a researcher were using only the theoretical datasets generated by `powsimR`, they would be overly confident in the methods' abilities to control false discovery proportion. Supplementary Figure S6 also indicates that methods generally have much more variable power between the `seqgendiff` datasets than between the `powsimR` datasets. Interestingly, the `seqgendiff` datasets indicate that methods tend to have smaller mean squared error than indicated by the `powsimR` datasets (Supplementary Figure S7).

3.3 Evaluating Factor Analyses

As we hope we have made clear, there are many approaches to differential expression analysis (Section 2.2), confounder adjustment (Section 2.3), library size adjustment (Section 2.4), and factor analysis (Section 2.5). We believe it to be beyond the scope of this work to exhaustively evaluate all of these methods — especially since new methods are being developed each year. Rather, we hope our simulation procedures will be used by the research community to more realistically evaluate and benchmark their approaches to RNA-seq data analysis.

However, as a final highlight to the utility of our simulation approaches, we demonstrate these simulation techniques in one application: evaluating factor analysis methods in RNA-seq (Section 2.5). We have chosen to highlight this particular application because it uses the more general simulation techniques beyond the two-group model, which were first demonstrated in [Gerard and

Stephens, 2017].

We chose to focus on the following methods based on (i) previous use in expression studies, (ii) software availability, (iii) popularity, and (iv) ease of use.

1. Principle component analysis (PCA) [Hotelling, 1933],
2. Sparse singular value decomposition (SSVD) [Yang et al., 2014],
3. Independent component analysis (ICA) [Hyvärinen and Oja, 2000],
4. Factors and loadings by adaptive shrinkage (*flash*), an empirical Bayes matrix factorization approach proposed in [Wang and Stephens, 2018], and
5. Probabilistic estimation of expression residuals (PEER) [Stegle et al., 2010], a Bayesian factor analysis used in the popular PEER software to adjust for hidden confounders in gene expression studies.

All factor analysis methods were applied to the $\log_2$-counts after adding half a pseudo-count. To simulate RNA-seq data, we took the muscle GTEx data [GTEx Consortium, 2017] and removed all genes with less than an average of 10 reads per sample. Each replicate, we added a rank-1 term. That is we assumed model (9) for the muscle GTEx data, then generated RNA-seq data such that

$$\tilde{\boldsymbol{\Theta}} = \boldsymbol{\mu}\mathbf{1}_N^\intercal + \boldsymbol{b}_3\boldsymbol{x}_3^\intercal\boldsymbol{\Pi}^\intercal + \boldsymbol{A}\boldsymbol{Z}^\intercal + \boldsymbol{\Omega}, \tag{19}$$

where we simulated the components of $\boldsymbol{x}_3$ and the non-zero components of $\boldsymbol{b}_3$ from independent normal distributions. We varied the following parameters of the simulation study:

1. The sample size: $N \in \{10, 20, 40\}$
2. The signal strength: the standard deviation of the loadings (the b_{3g}'s) was set to one of $\{0.4, 0.8\}$, with higher standard deviations corresponding to higher signal. These values were chosen to have the median PVE vary greatly between the two settings (Supplementary Figure S8),
3. The sparsity: the proportion of loadings (the b_{3g}'s) that are 0 was set to one of $\{0, 0.9\}$, and
4. The target correlations of the added factor with the first unobserved factor: $r \in \{0, 0.5\}$.

This resulted in 24 unique simulation parameter settings. We also used 1000 genes each replication. For each setting, we ran 100 replications of generating data from model (19), and fitting the factors with the five methods under study assuming model (11) after we estimated the number of hidden factors using parallel analysis [Buja and Eyuboglu, 1992].

We chose three metrics to evaluate the performance of the different factor analysis methods:

1. The minimum mean squared error between $\boldsymbol{\Pi}\boldsymbol{x}_3$ and the columns of $\hat{\boldsymbol{F}}$. To account for scale and sign unidentifiability, the estimated factors and the added factor were all scaled to have an ℓ^2-norm of 1 prior to calculating the mean squared error. This measure is meant to evaluate if any of the estimated factors corresponds to the added factor.
2. The minimum mean squared error between $\boldsymbol{b}_3$ and the columns of $\hat{\boldsymbol{L}}$. We again accounted for scale and sign unidentifiability by calculating the mean squared error after scaling the estimated and true loadings to have an ℓ^2-norm of 1.
3. The angle between $\boldsymbol{\Pi}\boldsymbol{x}_3$ and its projection onto the column space of $\hat{\boldsymbol{F}}$. This measure is meant to evaluate if the estimated factor matrix includes $\boldsymbol{\Pi}\boldsymbol{x}_3$ among its unidentified factors.

The results are presented in Supplementary Figures S9-S14. Based on these figures, we have the following conclusions:

1. PEER performs very poorly when either the sparsity is high or when there are few samples. It also performs less well when the factors are correlated. A possible explanation is that PEER assumes a normal distribution on the factors and loadings, which is violated in the high-sparsity regime and is observed in the low-sparsity regime. Though, this does not explain its poor performance in small sample size settings.
2. SSVD estimates the loadings very poorly in low-sparsity regimes. This is to be as expected as SSVD assumes sparsity on the loadings. Surprisingly, though, it outperforms PCA in high sparsity regimes only when both the sample size and signal are also large.
3. ICA performs very poorly in low sparsity regimes. This is to be as expected as the normal distributions placed on the factors and loadings are a worst-case scenario for ICA. However, there is no scenario where ICA performs significantly better than PCA.
4. *flash* performs adequately in all scenarios and performs best in high-sparsity and high-signal regimes.
5. PCA performs adequately in most scenarios, and is only truly outperformed in high sparsity high signal regimes.

Based on these initial explorations, we would recommend users not use PEER, SSVD, or ICA and instead try either PCA or *flash*.

4 Discussion

We have focused on a log-linear model because of the large number of applications this generates (Sections 2.2, 2.3, 2.4, and 2.5). This linearity (on the log-link scale) is represented by the structure of the $\boldsymbol{Q}$ matrix in Procedure 1. However, it is possible to replace $\boldsymbol{Q}$ by any arbitrary $G \times N$ matrix that has non-positive entries. This might be useful for simulations that study adjusting for non-linear effects, such as bias due to GC content [Risso et al., 2011], as it allows you to introduce non-linear effects into an RNA-seq dataset. However, these non-linear effects would still be present only on the log-scale.

Our simulation procedures may be applicable beyond evaluating competing methods. Vieth et al. [2017] used their simulation software to estimate power given the sample size in a differential expression analysis, and thus to develop sample size suggestions. Our simulation methods may be used similarly. Given a large RNA-seq dataset (such as the GTEx data used in this paper), one can repeatedly down-sample the number of individuals in the dataset and explore how sample size affects the power of a differential expression analysis.

Similarly, Robinson and Storey [2014] already demonstrated that binomial thinning may be used for sequencing depth suggestions. That is, a researcher may repeatedly thin the libraries of the samples in an large RNA-seq dataset and explore the effects on power, thereby providing sequencing depth suggestions. Unlike Robinson and Storey [2014], which does this subsampling uniformly over all counts, we allow researchers to explore the effects of heterogeneous subsampling (as in Section 2.4). This might be useful if, say, researchers have more individuals in one group than in another and so wish to explore if they can sequence the larger group to a lower depth without affecting power.

In this manuscript, we have discussed our simulation techniques in the context of RNA-seq. However, our techniques would also be applicable to the comparative analysis of metagenomics methods [Jonsson et al., 2016]. Instead of quantifying gene expression, metagenomics quantifies

gene abundances within metagenomes. Our simulation techniques could be applied in this context by taking a real metagenomics dataset and adding signal to it by binomial thinning.

5 Conclusions

We developed a procedure to add a known amount of signal to any real RNA-seq dataset. We only assume that this signal comes in the form of a generalized linear model with a log-link function from a very flexible distribution. We demonstrated how real data contain features that are not captured by simulated data, and that this can cause important differences in the results of a simulation study. We highlighted our simulation approach by comparing a few popular factor analysis methods. We found that PCA and *flash* had the most robust performances across a wide range of simulation settings.

Availability of data and materials

The simulation methods discussed in this paper are implemented in the `seqgendiff` R package, available on the Comprehensive R Archive Network: https://cran.r-project.org/package=seqgendiff. All code to reproduce the simulation and analysis results is available on GitHub: https://github.com/dcgerard/reproduce_fasims.

The datasets analyzed during the current study are available in the GTEx portal: https://gtexportal.org.

Acknowledgments

We would like to thank Matthew Stephens for providing comments on a draft of this manuscript, and Joyce Hsiao for testing an early version of the `seqgendiff` software.

All graphics were made using ggplot2 [Wickham, 2016] in the R statistical language [R Core Team, 2019].

S1 Theoretical Considerations

S1.1 Target Correlation

Theorem 1. *Let (u_i, z_i) be iid jointly standard normal with correlation ρ for $i = 1, 2, \ldots, n$. Let w_j be iid standard normal for $j = 1, 2, \ldots, n$. Suppose we match the w_j's onto the u_i's by order statistics, resulting in (w_i, u_i) pairs such that the rank of w_i is the same as the rank of u_i. Then* $\text{cor}(w_i, z_i) \underset{n\to\infty}{\to} \rho$.

Proof. For a fixed proportion p, we note that $u_{(\lceil np \rceil)}$ and $w_{(\lceil np \rceil)}$ converge in probability to the theoretical p-quantile of the standard normal distribution [Arnold et al., 1992, e.g.]. Since the order statistics converge to the same values, and we match by order statistics, this implies that $u_i - w_i \xrightarrow{P} 0$. Thus, by Slutsky's theorem, we have that $u_i z_i - w_i z_i \xrightarrow{P} 0$.

We note that $|u_i z_i - w_i z_i| \leq |u_i z_i| + |w_i z_i|$, by the triangle inequality. The term on the right has finite expectation as (using Cauchy-Schwarz)

$$\text{E}[|u_i z_i| + |w_i z_i|] \leq \text{E}[u_i^2]^{1/2}\,\text{E}[z_i^2]^{1/2} + \text{E}[w_i^2]^{1/2}\,\text{E}[z_i^2]^{1/2} = 2. \tag{20}$$

Thus, by the Lebesgue dominated convergence theorem, we have $\text{E}[|u_i z_i - w_i z_i|] \to 0$. Since $-|u_i z_i - w_i z_i| \leq u_i z_i - w_i z_i \leq |u_i z_i - w_i z_i|$, this implies that $\text{E}[u_i z_i] - E[w_i z_i] \to 0$, and the theorem is proved.

□

To place the results of Theorem 1 in context of the matching in Procedure 2, note that the u_i's are the elements of $\boldsymbol{U}$, the z_i's are the elements of $\hat{\boldsymbol{Z}}$, the w_i's are the elements of $\boldsymbol{X}_3$, ρ is the target correlation between the one column in $\boldsymbol{X}_3$ and the one column in $\hat{\boldsymbol{Z}}$, and $\boldsymbol{\Pi}$ is the permutation matrix that results in the matching of the w_i's and the u_i's.

The results of Theorem 1 can be generalized to non-standard normal distributions by appealing to the weak law of large numbers and Slutsky's theorem.

S1.2 Generalizing the Poisson Assumption

For simplicity, we stated a Poisson distribution as the modeling assumption in (1). However, our methods are equally valid under more general conditions. We begin by showing how our methods are valid when using the negative binomial distribution, which is perhaps the most common distribution used to analyze RNA-seq counts [Robinson and Smyth, 2007a,b, Love et al., 2014]. To see this, we prove the following simple lemma which, though less well-known than Lemma 1, can still be found in some elementary texts (or at least a version of the following lemma) [exercise 4.32 of Casella and Berger, 2002, e.g.].

Lemma 2. *Suppose $y \sim \text{NB}(\mu, \phi)$, where we are using the parameterization such that $\text{E}[y] = \mu$ and $\text{var}(y) = \mu(1 + \mu\phi)$. Also suppose that $\tilde{y}|y \sim \text{Bin}(y, p)$. Then $\tilde{y} \sim \text{NB}(p\mu, \phi)$.*

Proof. Using the hierarchical characterization of the negative binomial distribution, we have that

$$\lambda \sim \text{Gamma}(1/\phi, \mu\phi) \tag{21}$$

$$y|\lambda \sim \text{Poisson}(\lambda), \tag{22}$$

where $1/\phi$ is the shape parameter and $\mu\phi$ is the scale parameter. This implies that $\tilde{y}|\lambda \sim \text{Poisson}(p\lambda)$. But $p\lambda \sim \text{Gamma}(1/\phi, p\mu\phi)$ by elementary properties of the gamma distribution. Hence, by the hierarchical characterization of the negative binomial distribution, we have that $\tilde{y} \sim \text{NB}(p\mu, \phi)$. □

The zero-inflated negative binomial distribution is sometimes used to model single-cell RNA-seq data as it can account for the abundance of zeros observed in such data [Miao et al., 2018, Risso et al., 2018, Eraslan et al., 2019]. A random variable y is distributed zero-inflated negative binomial, denoted $y \sim \text{ZINB}(\pi, \mu, \phi)$, if it is generated by the following hierarchical process:

$$z \sim \text{Bern}(1-\pi), \tag{23}$$
$$y|z=0 \sim \delta_0 \tag{24}$$
$$y|z=1 \sim \text{NB}(\mu, \phi), \tag{25}$$

where δ_0 is the degenerate distribution with a point-mass at 0. In words, the counts are either 0 with probability π or follow a negative binomial distribution with probability $1-\pi$. Our methods are equally valid in the zero-inflated negative binomial case.

Lemma 3. *Suppose* $y \sim \text{ZINB}(\pi, \mu, \phi)$ *and* $\tilde{y}|y \sim \text{Bin}(y, p)$. *Then* $\tilde{y} \sim \text{ZINB}(\pi, p\mu, \phi)$.

Proof. It's sufficient to note that

$$\tilde{y}|z=0 \sim \delta_0, \text{ and} \tag{26}$$
$$\tilde{y}|z=1 \sim \text{NB}(p\mu, \phi). \tag{27}$$

□

Finally, our simulation methods preserve the count distribution in the rich class of distributions which are mixtures of binomial and negative binomial distributions (some examples within this class of distributions are plotted in Supplementary Figure S1).

Lemma 4. *Let* $\pi_0, \pi_1, \ldots \pi_M$ *and* $\tau_1, \tau_2, \ldots, \tau_L$ *be non-negative mixing proportions such that*

$$\sum_{m=0}^{M} \pi_m + \sum_{\ell=1}^{L} \tau_\ell = 1. \tag{28}$$

Suppose that y *has a PMF which is a mixture of binomial and negative binomial PMF's*

$$f(y) = \pi_0 \delta_0(y) + \sum_{m=1}^{M} \pi_m \,\text{NB}(y|\mu_m, \phi_m) + \sum_{\ell=1}^{L} \tau_\ell \,\text{Bin}(y|\frac{\nu_\ell}{n_\ell}, n_\ell), \tag{29}$$

where $\text{NB}(y|\mu_m, \phi_m)$ *is the negative binomial PMF with mean* μ_m *and dispersion* ϕ_m, *and* $\text{Bin}(y|\frac{\nu_\ell}{n_\ell}, n_\ell)$ *is the binomial PMF with mean* ν_ℓ *and success probability* ν_ℓ/n_ℓ. *Suppose that* $\tilde{y}|y \sim \text{Bin}(y, p)$. *Then*

$$\text{E}[\tilde{y}] = p\,\text{E}[y], \text{ and} \tag{30}$$
$$f(\tilde{y}) = \pi_0 \delta_0(\tilde{y}) + \sum_{m=1}^{M} \pi_m \,\text{NB}(\tilde{y}|p\mu_m, \phi_m) + \sum_{\ell=1}^{L} \tau_\ell \,\text{Bin}(\tilde{y}|\frac{p\nu_\ell}{n_\ell}, n_\ell). \tag{31}$$

Proof. Equation (30) is just a consequence of the law of total expectation. To prove (31), note that if $y \sim \text{NB}(\mu_m, \phi_m)$ then $\tilde{y} \sim \text{NB}(p\mu_m, \phi_m)$ and if $y \sim \text{Bin}(\nu_\ell/n_\ell, n_\ell)$ then $\tilde{y} \sim \text{Bin}(p\nu_\ell/n_\ell, n_\ell)$. The proof follows by conditioning on the latent mixing group. □

S2 Additional Simulations

S2.1 Correlation Estimator

We explored the effects of changing the target correlation on the true correlation. We varied the sample size, $N \in \{6, 10, 20\}$, and the target correlations between $\boldsymbol{z}$ and the two columns in $\boldsymbol{\Pi}\boldsymbol{X}_3$, $\boldsymbol{r} \in \{(0,0), (0.5,0), (0.9,0), (0.5,0.5)\}$. Under each unique combination of simulation parameter settings, we iterative drew $\boldsymbol{z} \in \mathbb{R}^N$ from a standard normal. We also drew $\boldsymbol{X}_3 \in \mathbb{R}^{N\times 2}$ according to two schemes:

1. Normal: Each element of $\boldsymbol{X}_3$ is independently drawn from a standard normal distribution, and
2. Indicator: The first column of $\boldsymbol{X}_3$ consists of $(1, 0, 1, 0, \ldots, 1, 0)^\intercal$, and the second column of $\boldsymbol{X}_3$ consists of $(\mathbf{1}_{N/2}^\intercal, \mathbf{0}_{N/2}^\intercal)^\intercal$.

Each replicate, we used Procedure 4 to estimate the correlation between $\boldsymbol{\Pi}\boldsymbol{X}_3$ and $\boldsymbol{z}$. We did this for a total of 100 replications for each combination of simulation parameters.

The results are presented in Supplementary Figure S15. Because we are approximating the expected (conditional on $\boldsymbol{z}$) Pearson correlation between the columns of $\boldsymbol{\Pi}\boldsymbol{X}_3$ and $\boldsymbol{z}$, the true correlations between $\boldsymbol{\Pi}\boldsymbol{X}_3$ and $\boldsymbol{z}$ are approximately the mean of the estimates over the 100 replications (see (17)). From Supplementary Figure S15, we note that the true correlation is generally closer to 0 than the target correlation. When the sample size is 20, there seems to be very little variability in the correlation estimates.

S3 Supplementary Figures

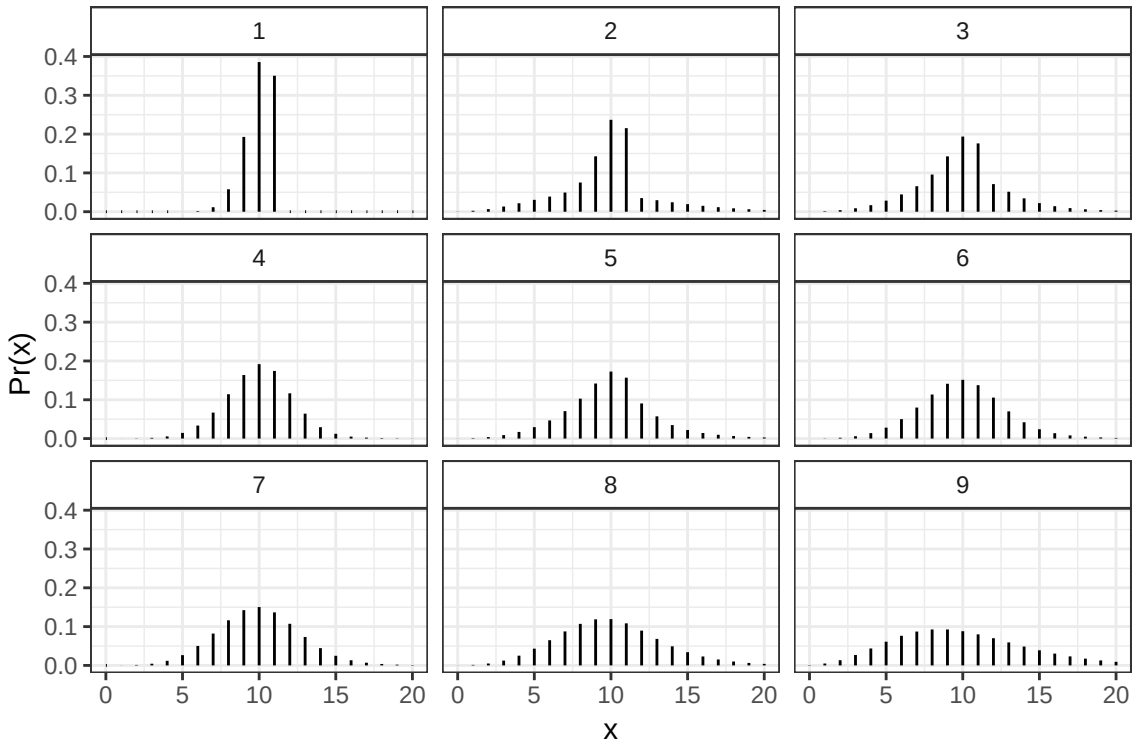

Figure S1: Some distributions in the class of mixtures of binomial and negative binomial distributions, demonstrating the flexibility of this class. The mean was set to 10 for all mixing components.

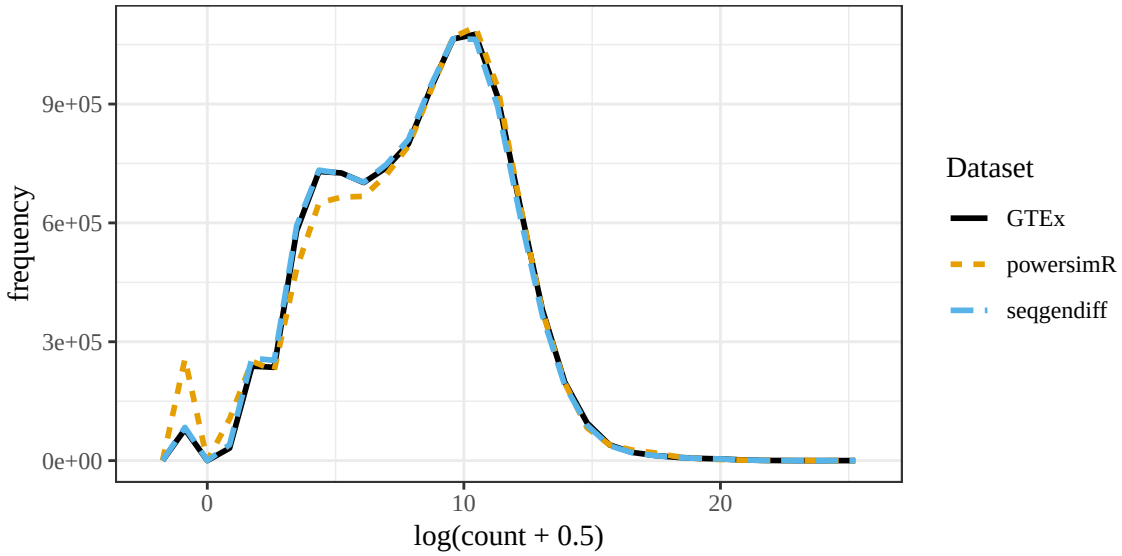

Figure S2: Frequency polygons for the $\log_2$ counts for the dataset simulated by `seqgendiff` (blue), the dataset simulated by `powsimR` (orange), and the GTEx muscle dataset (black). The `seqgendiff` and GTEx count distributions are almost exactly the same.

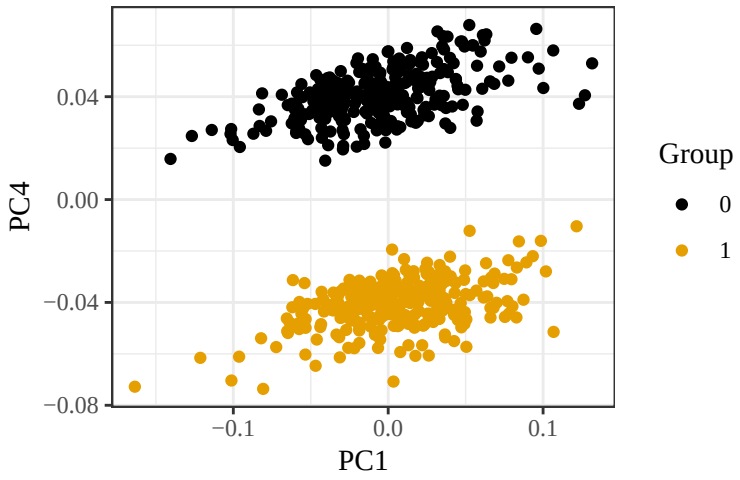

Figure S3: First principle component (x-axis) versus the fourth principle component (y-axis) in the `seqgendiff` dataset. The fourth principle component seems capture group membership.

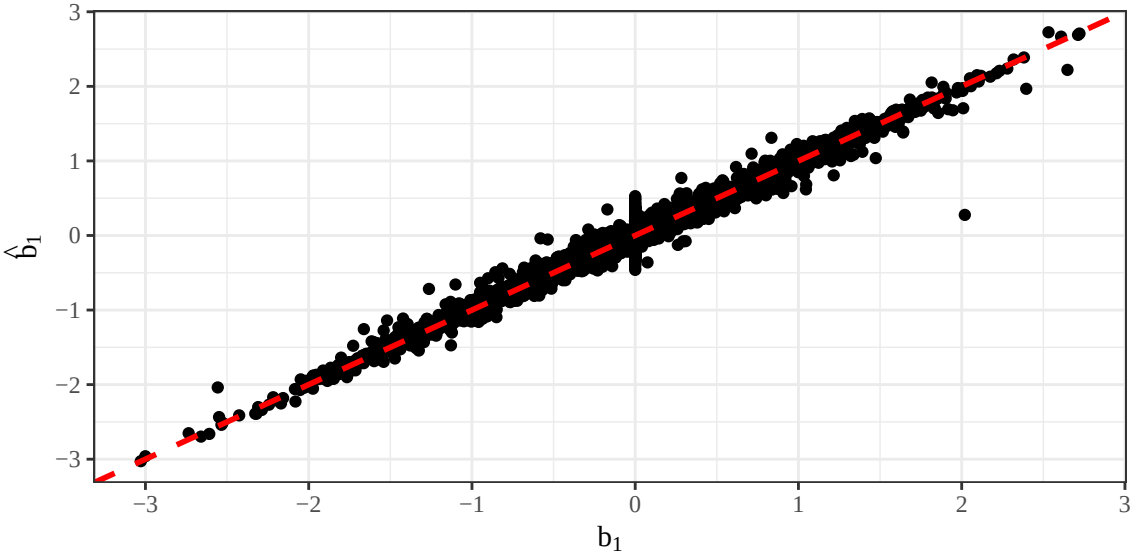

Figure S4: True coefficient values (x-axis) versus their corresponding estimates (y-axis) in the `seqgendiff` dataset. Estimates were obtained using the voom-limma pipeline [Smyth, 2004, Law et al., 2014].

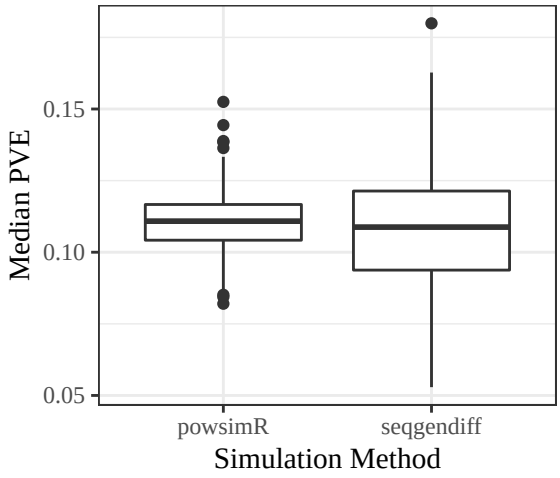

Figure S5: Median (across non-null genes) proportion of variance explained (18) (y-axis) for datasets generated by `powsimR` or `seqgendiff` (x-axis). The two sets of simulated datasets have the same expected median PVE, though the median PVE is more variable among the `seqgendiff` datasets.

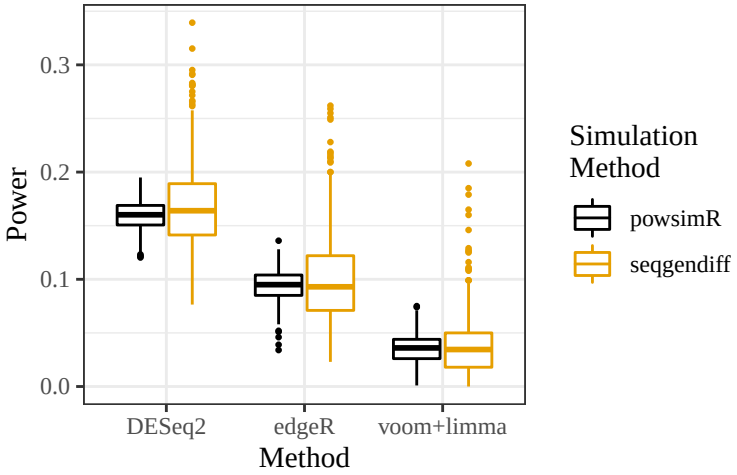

Figure S6: Boxplots of power (y-axis) for various differential expression analysis methods (x-axis) when applied on different simulated datasets (color).

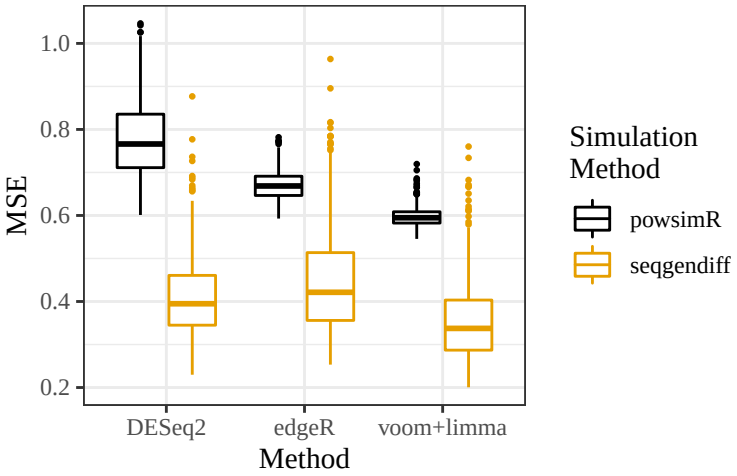

Figure S7: Boxplots of mean squared error (y-axis) for various differential expression analysis methods (x-axis) when applied on different simulated datasets (color).

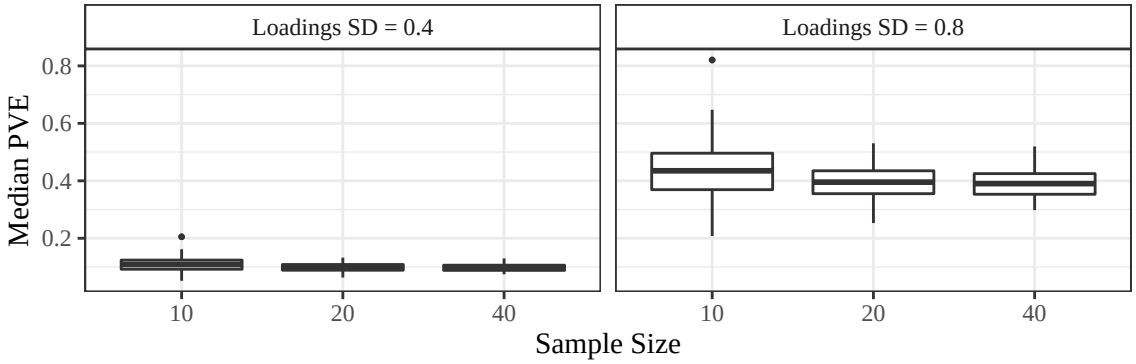

Figure S8: Median proportion of variance explained (18) for the genes with non-zero loadings (y-axis) stratified by the sample size (x-axis) from the simulation study in Section 3.3. The facets index the different standard deviations of the loadings.

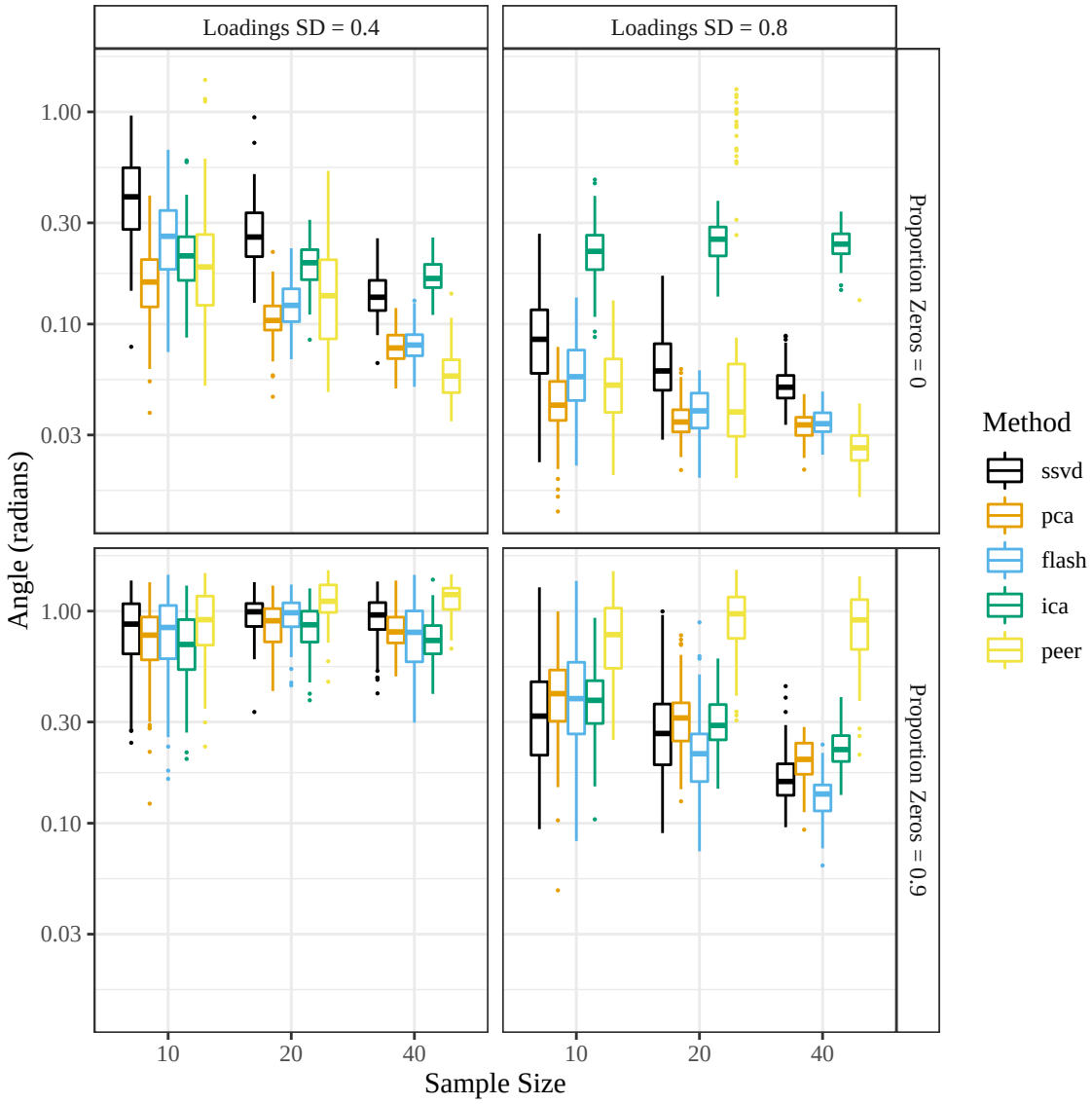

Figure S9: Angle ($\log_{10}$-scale) between the added factor and its projection onto the column space of the estimated factors (y-axis), stratified by sample size (x-axis), factor analysis method (color), signal strength (column facets), and sparsity of the loadings (row facets). The target correlation between the added factor and the first unobserved factor was set to 0. A smaller angle is better.

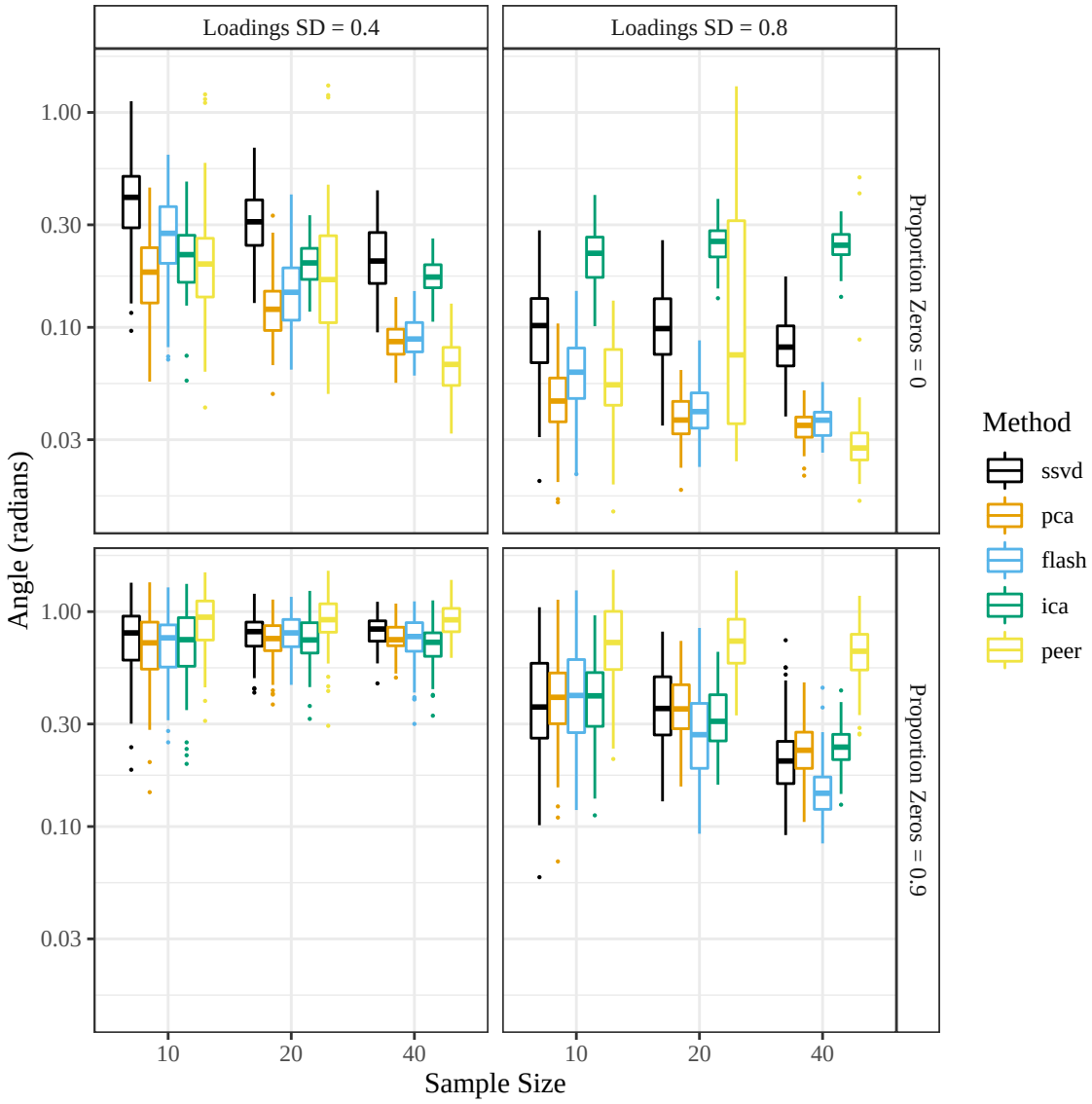

Figure S10: Angle ($\log_{10}$-scale) between the added factor and its projection onto the column space of the estimated factors (y-axis), stratified by sample size (x-axis), factor analysis method (color), signal strength (column facets), and sparsity of the loadings (row facets). The target correlation between the added factor and the first unobserved factor was set to 0.5. A smaller angle is better.

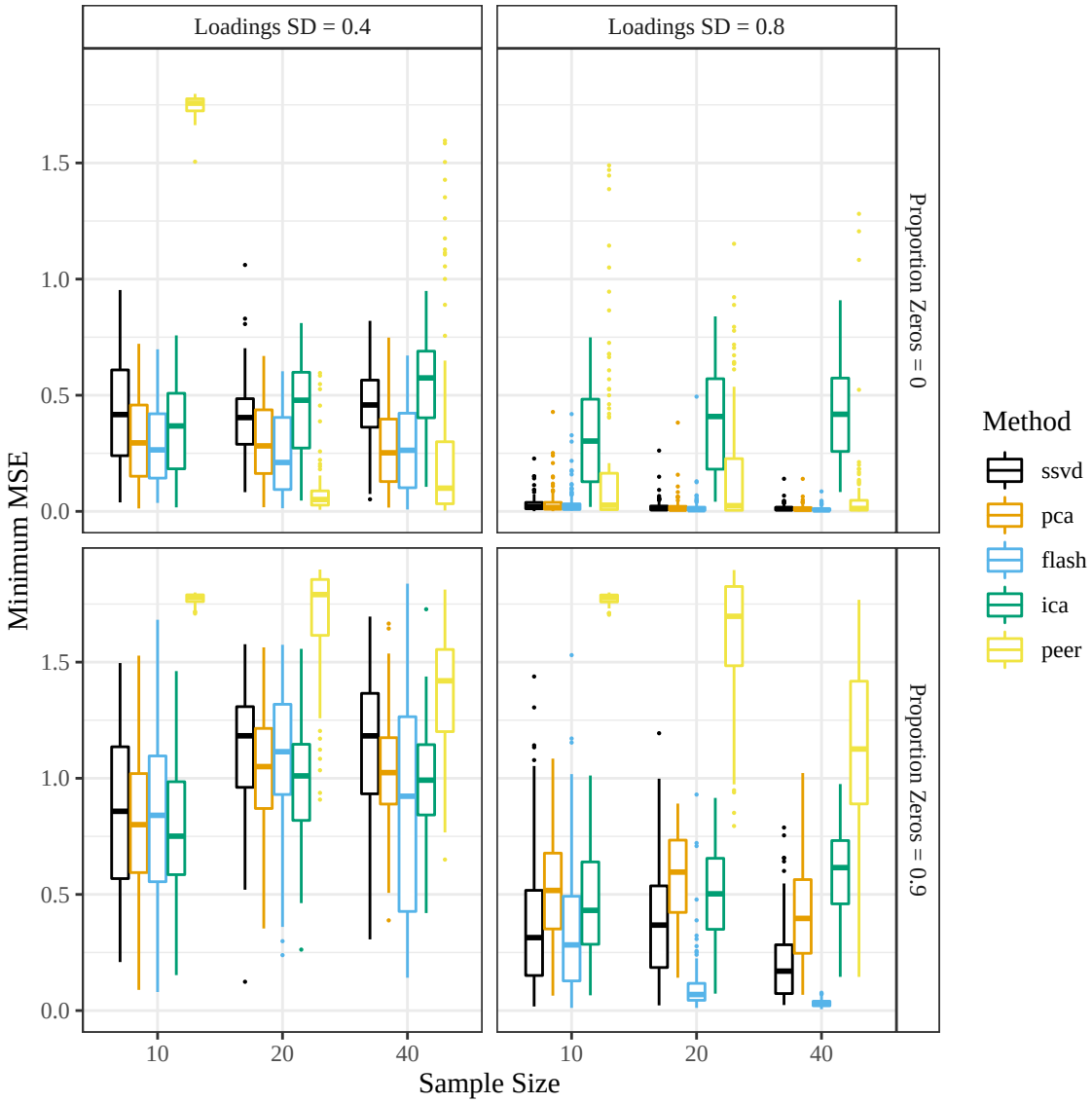

Figure S11: Minimum mean squared error between the added factor and the estimated factors (y-axis), stratified by sample size (x-axis), factor analysis method (color), signal strength (column facets), and of the loadings sparsity (row facets). Before calculating the MSE, all factors were scaled to have ℓ^2-norm of 1. The target correlation between the added factor and the first unobserved factor was set to 0. A smaller minimum MSE is better.

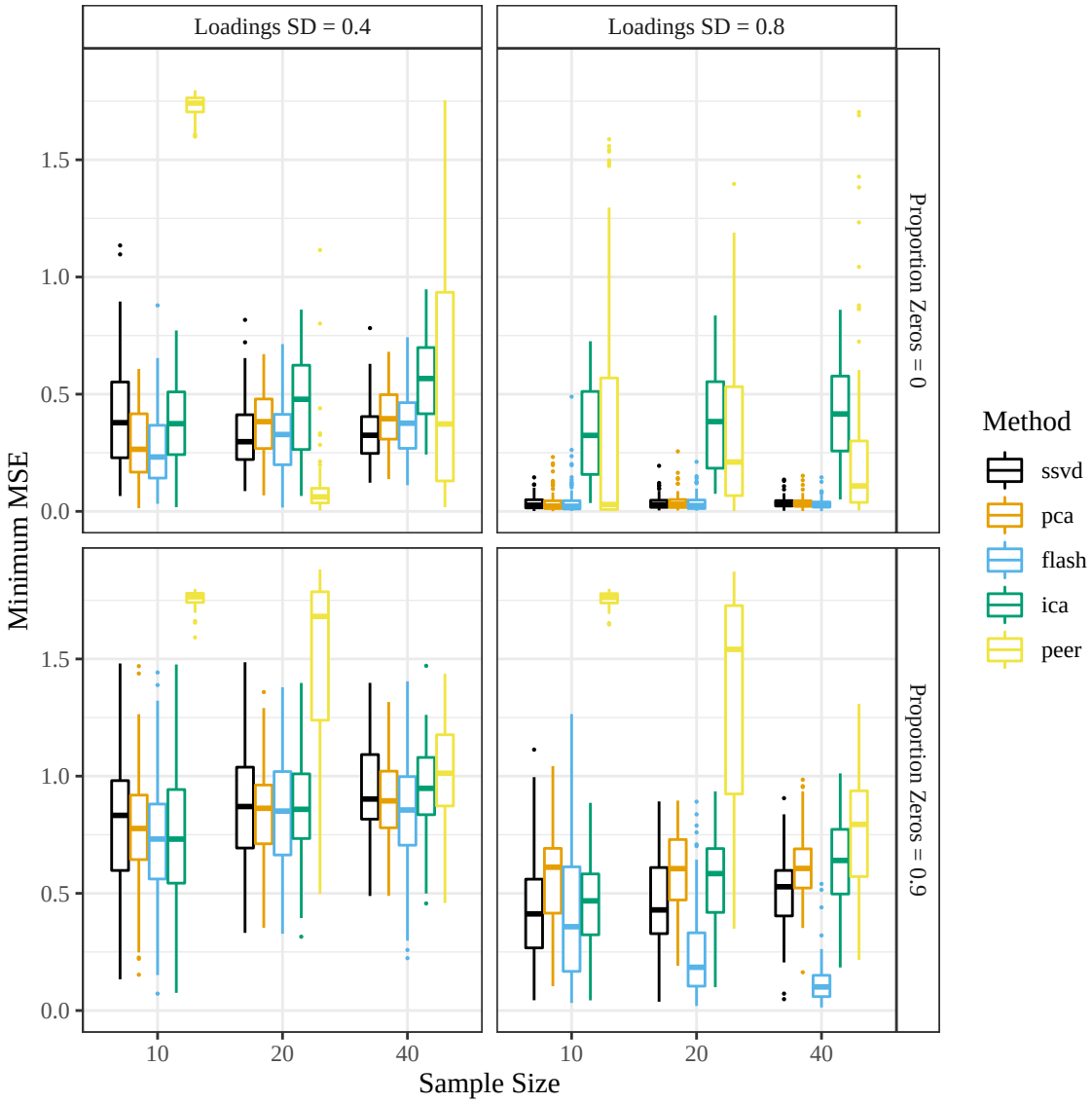

Figure S12: Minimum mean squared error between the added factor and the estimated factors (y-axis), stratified by sample size (x-axis), factor analysis method (color), signal strength (column facets), and sparsity of the loadings (row facets). Before calculating the MSE, all factors were scaled to have ℓ^2-norm of 1. The target correlation between the added factor and the first unobserved factor was set to 0.5. A smaller minimum MSE is better.

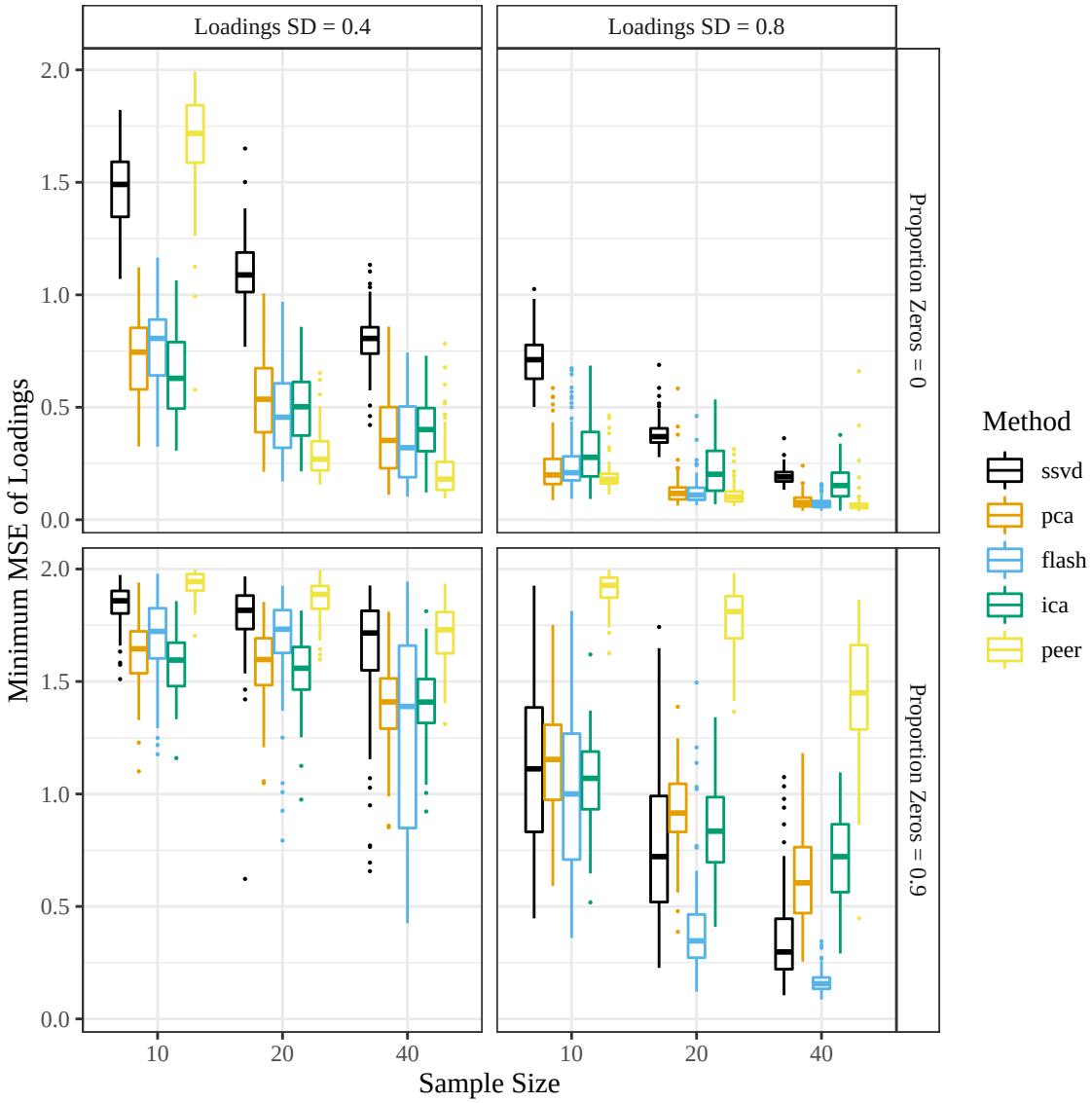

Figure S13: Minimum mean squared error between the added loading and the estimated loadings (y-axis), stratified by the sample size (x-axis), the factor analysis method (color), signal strength (column facets), and sparsity of the loadings (row facets). Before calculating the MSE, all loadings were scaled to have ℓ^2-norm of 1. The target correlation between the added factor and the first unobserved factor was set to 0. A smaller minimum MSE is better.

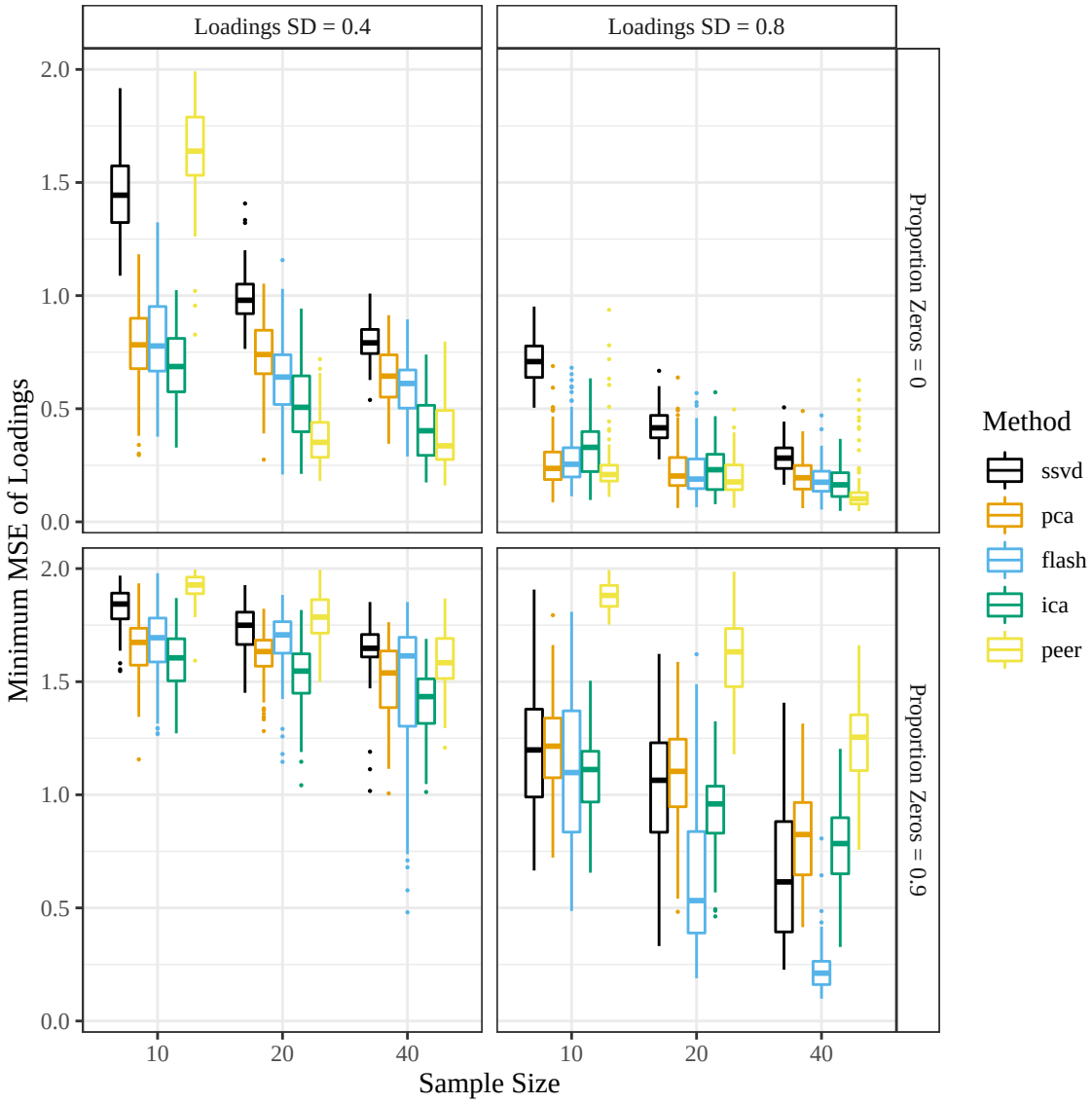

Figure S14: Minimum mean squared error between the added loading and the estimated loadings (y-axis), stratified by the sample size (x-axis), the factor analysis method (color), signal strength (column facets), and sparsity of the loadings (row facets). Before calculating the MSE, all loadings were scaled to have ℓ^2-norm of 1. The target correlation between the added factor and the first unobserved factor was set to 0.5. A smaller minimum MSE is better.

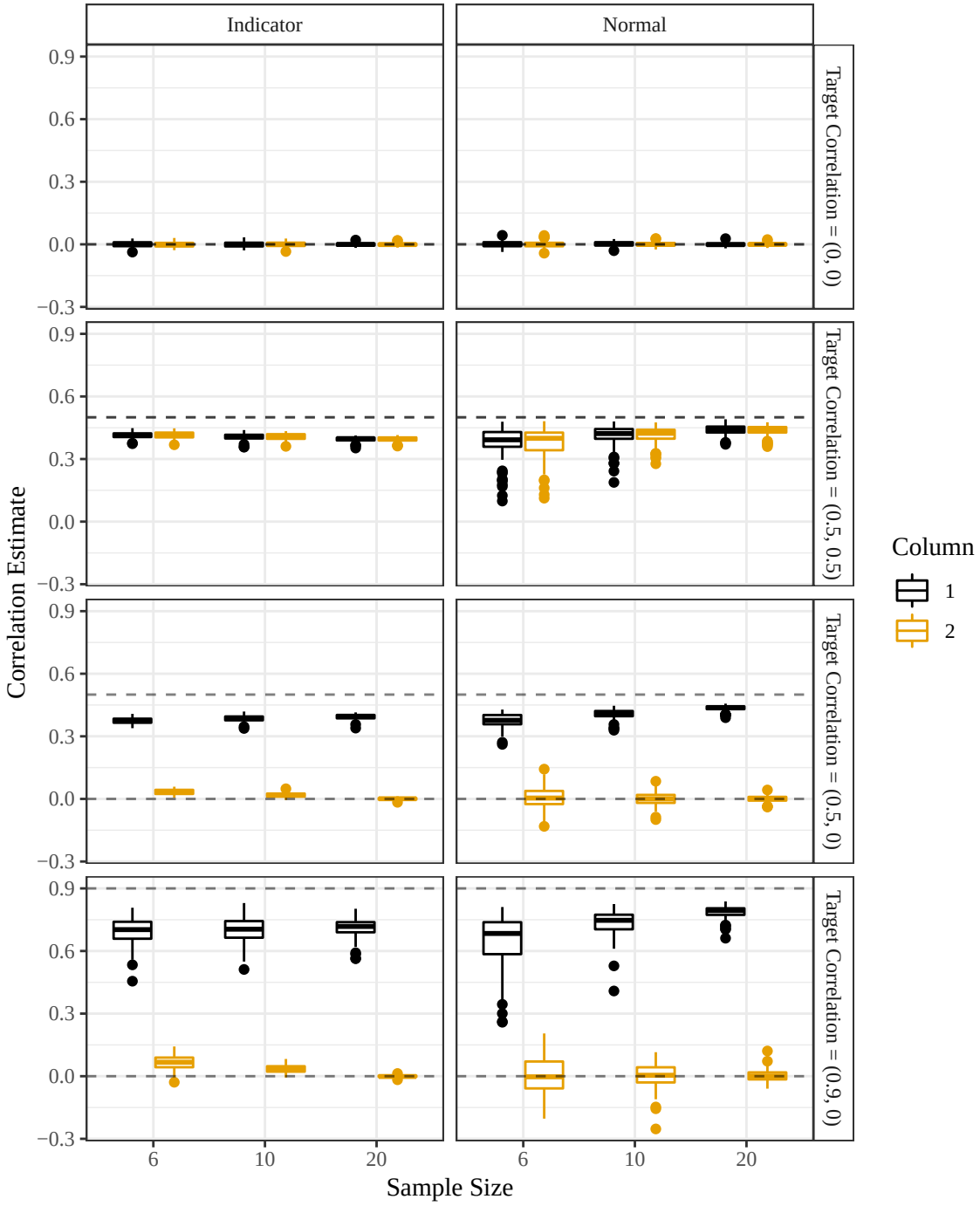

Figure S15: Boxplots of the Monte Carlo correlation estimator from Procedure 4 (y-axis), stratified by sample size (x-axis), the type of design matrix (column facets), the target correlations (row facets), and the column of the design matrix (color). Horizontal dashed lines are the two target correlations. The mean of the correlation estimates is approximately the true correlation.

References

Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, Oct 2010. ISSN 1474-760X. doi: 10.1186/gb-2010-11-10-r106.

Barry C Arnold, Narayanaswamy Balakrishnan, and Haikady Navada Nagaraja. *A first course in order statistics*, volume 54. Siam, 1992.

Sam Benidt and Dan Nettleton. SimSeq: a nonparametric approach to simulation of RNA-sequence datasets. *Bioinformatics*, 31(13):2131–2140, 02 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv124.

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)*, pages 289–300, 1995. URL `http://www.jstor.org/stable/2346101`.

Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155, 2015. doi: 10.1038/nbt.3102.

Andreas Buja and Nermin Eyuboglu. Remarks on parallel analysis. *Multivariate behavioral research*, 27(4): 509–540, 1992. doi: 10.1207/s15327906mbr2704_2.

James H. Bullard, Elizabeth Purdom, Kasper D. Hansen, and Sandrine Dudoit. Evaluation of statistical methods for normalization and differential expression in mRNA-seq experiments. *BMC Bioinformatics*, 11(1):94, Feb 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-94.

Carlos M. Carvalho, Jeffrey Chang, Joseph E. Lucas, Joseph R. Nevins, Quanli Wang, and Mike West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008. doi: 10.1198/016214508000000869. PMID: 21218139.

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

Kevin Caye, Basile Jumentier, and Olivier François. LFMM 2.0: Latent factor models for confounder adjustment in genome and epigenome-wide association studies. *bioRxiv*, 2018. doi: 10.1101/255893.

Mengjie Chen and Xiang Zhou. Controlling for confounding effects in single cell RNA sequencing studies using both control and target genes. *Scientific reports*, 7(1):13587, 2017. doi: 10.1038/s41598-017-13665-w.

Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, 1994. ISSN 0165-1684. doi: 10.1016/0165-1684(94)90029-9. Higher Order Statistics.

Juliana Costa-Silva, Douglas Domingues, and Fabricio Martins Lopes. RNA-Seq differential expression analysis: An extended review and a software tool. *PLOS ONE*, 12(12):1–18, 12 2017. doi: 10.1371/journal.pone.0190152.

Mihails Delmans and Martin Hemberg. Discrete distributional differential expression (D3E) - a tool for gene expression analysis of single-cell RNA-seq data. *BMC Bioinformatics*, 17(1):110, Feb 2016. ISSN 1471-2105. doi: 10.1186/s12859-016-0944-6.

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, Mickaël 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, 09 2012. ISSN 1477-4054. doi: 10.1093/bib/bbs046.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1 (3):211–218, 1936. ISSN 0033-3123. doi: 10.1007/BF02288367.

Barbara E. Engelhardt and Matthew Stephens. Analysis of population structure: A unifying framework and novel methods based on sparse factor analysis. *PLOS Genetics*, 6(9):1–12, 09 2010. doi: 10.1371/journal.pgen.1001117.

Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390, 2019. doi: 10.1038/s41467-018-07931-2.

Greg Finak, Andrew McDavid, Masanao Yajima, Jingyuan Deng, Vivian Gersuk, Alex K. Shalek, Chloe K. Slichter, Hannah W. Miller, M. Juliana McElrath, Martin Prlic, Peter S. Linsley, and Raphael Gottardo. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology*, 16(1):278, Dec 2015. ISSN 1474-760X. doi: 10.1186/s13059-015-0844-5.

Chloé Friguet, Maela Kloareg, and David Causeur. A factor model approach to multiple testing under dependence. *Journal of the American Statistical Association*, 104(488):1406–1415, 2009. doi: 10.1198/jasa.2009.tm08332.

Nicoló Fusi, Oliver Stegle, and Neil D. Lawrence. Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies. *PLoS Computational Biology*, 8(1):1–9, 01 2012. doi: 10.1371/journal.pcbi.1002330.

Gary L. Gadbury, Qinfang Xiang, Lin Yang, Stephen Barnes, Grier P. Page, and David B. Allison. Evaluating statistical methods using plasmode data sets in the age of massive public databases: An illustration using false discovery rates. *PLOS Genetics*, 4(6):1–8, 06 2008. doi: 10.1371/journal.pgen.1000098.

Johann Gagnon-Bartsch, Laurent Jacob, and Terence Speed. Removing unwanted variation from high dimensional data with negative controls. Technical report, Technical Report 820, Department of Statistics, University of California, Berkeley, 2013. URL http://statistics.berkeley.edu/tech-reports/820.

Johann A Gagnon-Bartsch and Terence P Speed. Using control genes to correct for unwanted variation in microarray data. *Biostatistics*, 13(3):539–552, 2012. doi: 10.1093/biostatistics/kxr034.

D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962. ISSN 00029890, 19300972. doi: 10.2307/2312726. URL http://www.jstor.org/stable/2312726.

David Gerard and Matthew Stephens. Unifying and generalizing methods for removing unwanted variation based on negative controls. *arXiv preprint arXiv:1705.08393*, 2017. URL https://arxiv.org/abs/1705.08393.

David Gerard and Matthew Stephens. Empirical Bayes shrinkage and false discovery rate estimation, allowing for unwanted variation. *Biostatistics*, 07 2018. ISSN 1465-4644. doi: 10.1093/biostatistics/kxy029.

Joyee Ghosh and David B Dunson. Default prior distributions and efficient posterior computation in Bayesian factor analysis. *Journal of Computational and Graphical Statistics*, 18(2):306–320, 2009. doi: 10.1198/jcgs.2009.07145.

GTEx Consortium. Genetic effects on gene expression across human tissues. *Nature*, 550(7675):204, 2017. doi: 10.1038/nature24277.

Minzhe Guo, Hui Wang, S. Steven Potter, Jeffrey A. Whitsett, and Yan Xu. SINCERA: A pipeline for single-cell RNA-seq profiling analysis. *PLOS Computational Biology*, 11(11):1–28, 11 2015. doi: 10.1371/journal.pcbi.1004575.

Ben B Hansen and Stephanie Olsen Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627, 2006. doi: 10.1198/106186006X137047.

Thomas J. Hardcastle and Krystyna A. Kelly. baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, 11(1):422, Aug 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-422.

Peter D. Hoff. Model averaging and dimension selection for the singular value decomposition. *J. Amer. Statist. Assoc.*, 102(478):674–685, 2007. ISSN 0162-1459. doi: 10.1198/016214506000001310.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933. doi: 10.1037/h0071325.

Hung Hung. A robust RUV-testing procedure via γ-divergence. *Biometrics*, 2018. doi: 10.1111/biom.13002.

Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, 50(8):96, 2018. doi: 10.1038/s12276-018-0071-8.

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411 – 430, 2000. ISSN 0893-6080. doi: 10.1016/S0893-6080(00)00026-5.

Viktor Jonsson, Tobias Österlund, Olle Nerman, and Erik Kristiansson. Statistical evaluation of methods for identification of differentially abundant genes in comparative metagenomics. *BMC genomics*, 17(1): 78, 2016. doi: 10.1186/s12864-016-2386-y.

Julie Josse and Stefan Wager. Bootstrap-based regularization for low-rank matrix estimation. *Journal of Machine Learning Research*, 17(124):1–29, 2016. URL `http://jmlr.org/papers/v17/14-534.html`.

Hyun Min Kang, Chun Ye, and Eleazar Eskin. Accurate discovery of expression quantitative trait loci under confounding from spurious and genuine regulatory hotspots. *Genetics*, 180(4):1909–1925, 2008a. doi: 10.1534/genetics.108.094201.

Hyun Min Kang, Noah A Zaitlen, Claire M Wade, Andrew Kirby, David Heckerman, Mark J Daly, and Eleazar Eskin. Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709–1723, 2008b. doi: 10.1534/genetics.107.080101.

Hyun Min Kang, Jae Hoon Sul, Susan K Service, Noah A Zaitlen, Sit-yee Kong, Nelson B Freimer, Chiara Sabatti, and Eleazar Eskin. Variance component model to account for sample structure in genome-wide association studies. *Nature genetics*, 42(4):348–354, 2010. doi: 10.1038/ng.548.

Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740, 2014. doi: 10.1038/nmeth.2967.

Keegan D. Korthauer, Li-Fang Chu, Michael A. Newton, Yuan Li, James Thomson, Ron Stewart, and Christina Kendziorski. A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biology*, 17(1):222, Oct 2016. ISSN 1474-760X. doi: 10.1186/s13059-016-1077-y.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 (1-2):83–97, 1955. doi: 10.1002/nav.3800020109.

Vanessa M. Kvam, Peng Liu, and Yaqing Si. A comparison of statistical methods for detecting differentially expressed genes from RNA-seq data. *American Journal of Botany*, 99(2):248–256, 2012. doi: 10.3732/ajb.1100340.

Ben Langmead, Kasper D. Hansen, and Jeffrey T. Leek. Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome Biology*, 11(8):R83, Aug 2010. ISSN 1474-760X. doi: 10.1186/gb-2010-11-8-r83.

Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome biology*, 15(R29), 2014. doi: 10.1186/gb-2014-15-2-r29.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999. doi: 10.1038/44565.

Seunggeun Lee, Wei Sun, Fred A. Wright, and Fei Zou. An improved and explicit surrogate variable analysis procedure by coefficient adjustment. *Biometrika*, 104(2):303–316, 2017. doi: 10.1093/biomet/asx018.

Jeffrey T. Leek. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Research*, 42(21):e161–e161, 10 2014. ISSN 0305-1048. doi: 10.1093/nar/gku864.

Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):1724–1735, 2007. doi: 10.1371/journal.pgen.0030161.

Jeffrey T Leek and John D Storey. A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences*, 105(48):18718–18723, 2008. doi: 10.1073/pnas.0808709105.

Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010. doi: 10.1038/nrg2825.

Dennis Leung and Mathias Drton. Order-invariant prior specification in Bayesian factor analysis. *Statistics & Probability Letters*, 111:60–66, 2016. doi: 10.1016/j.spl.2016.01.006.

Jennifer Listgarten, Carl Kadie, Eric E Schadt, and David Heckerman. Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences*, 107(38): 16465–16470, 2010. doi: 10.1073/pnas.1002425107.

Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550, Dec 2014. ISSN 1474-760X. doi: 10.1186/s13059-014-0550-8.

Mengyin Lu. *Generalized Adaptive Shrinkage Methods and Applications in Genomics Studies*. PhD thesis, University of Chicago, 2018. URL http://proxyau.wrlc.org/login?url=https://search.proquest.com/docview/2161785175?accountid=8285.

Vinicius Diniz Mayrink and Joseph Edward Lucas. Sparse latent factor models with interactions: Analysis of gene expression data. *Ann. Appl. Stat.*, 7(2):799–822, 06 2013. doi: 10.1214/12-AOAS607.

Davis J. McCarthy, Yunshun Chen, and Gordon K. Smyth. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297, 01 2012. ISSN 0305-1048. doi: 10.1093/nar/gks042.

Chris McKennan and Dan Nicolae. Accounting for unobserved covariates with varying degrees of estimability in high dimensional biological data. *arXiv preprint arXiv:1801.00865*, 2018a. URL https://arxiv.org/abs/1801.00865.

Chris McKennan and Dan Nicolae. Estimating and accounting for unobserved covariates in high dimensional correlated data. *arXiv preprint arXiv:1808.05895*, 2018b. URL https://arxiv.org/abs/1808.05895.

Tapan Mehta, Murat Tanik, and David B Allison. Towards sound epistemological foundations of statistical methods for high-dimensional biology. *Nature genetics*, 36(9):943, 2004. doi: 10.1038/ng1422.

Zhun Miao, Ke Deng, Xiaowo Wang, and Xuegong Zhang. DEsingle for detecting three types of differential expression in single-cell RNA-seq data. *Bioinformatics*, 34(18):3223–3224, 04 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty332.

Sara Mostafavi, Alexis Battle, Xiaowei Zhu, Alexander E Urban, Douglas Levinson, Stephen B Montgomery, and Daphne Koller. Normalizing RNA-sequencing data by modeling hidden covariates with prior knowledge. *PLoS One*, 8(7), 2013. doi: 10.1371/journal.pone.0068141.

Sheida Nabavi, Daniel Schmolze, Mayinuer Maitituoheti, Sadhika Malladi, and Andrew H Beck. Emdomics: a robust and powerful method for the identification of genes differentially expressed between heterogeneous classes. *Bioinformatics*, 32(4):533–541, 2015. doi: 10.1093/bioinformatics/btv634.

Dan Nettleton, Justin Recknor, and James M. Reecy. Identification of differentially expressed gene categories in microarray studies using nonparametric multivariate analysis. *Bioinformatics*, 24(2):192–201, 11 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm583.

Patrick O Perry and Natesh S Pillai. Degrees of freedom for combining regression with factor analysis. *arXiv preprint arXiv:1310.7269*, 2015. URL https://arxiv.org/abs/1310.7269.

Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. Single-cell mRNA quantification and differential analysis with Census. *Nature methods*, 14(3):309, 2017. doi: 10.1038/nmeth.4150.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/.

Pablo Reeb and Juan Steibel. Evaluating statistical analysis models for RNA sequencing experiments. *Frontiers in Genetics*, 4:178, 2013. ISSN 1664-8021. doi: 10.3389/fgene.2013.00178.

Guillem Rigaill, Sandrine Balzergue, Véronique Brunaud, Eddy Blondet, Andrea Rau, Odile Rogier, José Caius, Cathy Maugis-Rabusseau, Ludivine Soubigou-Taconnat, Sébastien Aubourg, Claire Lurin, Marie-Laure Martin-Magniette, and Etienne Delannoy. Synthetic data sets for the identification of key ingredients for RNA-seq differential analysis. *Briefings in Bioinformatics*, 19(1):65–76, 10 2016. ISSN 1477-4054. doi: 10.1093/bib/bbw092.

Davide Risso, Katja Schwartz, Gavin Sherlock, and Sandrine Dudoit. GC-content normalization for RNA-seq data. *BMC Bioinformatics*, 12(1):480, Dec 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-480.

Davide Risso, John Ngai, Terence P Speed, and Sandrine Dudoit. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature biotechnology*, 32(9):896, 2014. doi: 10.1038/nbt.2931.

Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature communications*, 9(1): 284, 2018. doi: 10.1038/s41467-017-02554-5.

David G. Robinson and John D. Storey. subSeq: Determining Appropriate Sequencing Depth Through Efficient Read Subsampling. *Bioinformatics*, 30(23):3424–3426, 09 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu552.

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. ISSN 1474-760X. doi: 10.1186/gb-2010-11-3-r25.

Mark D. Robinson and Gordon K. Smyth. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, 23(21):2881–2887, 09 2007a. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm453.

Mark D. Robinson and Gordon K. Smyth. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, 9(2):321–332, 08 2007b. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm030.

Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 11 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp616.

David M Rocke, Luyao Ruan, Yilun Zhang, J. Jared Gossett, Blythe Durbin-Johnson, and Sharon Aviran. Excess false positive rates in methods for differential gene expression analysis using RNA-seq data. *bioRxiv*, 2015. doi: 10.1101/020784.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 880–887, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390267.

Antonio Scialdone, Kedar N Natarajan, Luis R Saraiva, Valentina Proserpio, Sarah A Teichmann, Oliver Stegle, John C Marioni, and Florian Buettner. Computational assignment of cell-cycle stage from single-cell transcriptome data. *Methods*, 85:54–61, 2015. doi: 10.1016/j.ymeth.2015.06.021.

G. K. Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004. doi: 10.2202/1544-6115.1027.

C. Spearman. "general intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904. ISSN 00029556. doi: 10.2307/1412107.

Oliver Stegle, Anitha Kannan, Richard Durbin, and John Winn. Accounting for non-genetic factors improves the power of eQTL studies. In Martin Vingron and Limsoon Wong, editors, *Research in Computational Molecular Biology: 12th Annual International Conference, RECOMB 2008, Singapore, March 30 - April 2, 2008. Proceedings*, pages 411–422, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-78839-3. doi: 10.1007/978-3-540-78839-3_35.

Oliver Stegle, Leopold Parts, Richard Durbin, and John Winn. A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies. *PLOS Computational Biology*, 6(5):1–11, 05 2010. doi: 10.1371/journal.pcbi.1000770.

Oliver Stegle, Leopold Parts, Matias Piipari, John Winn, and Richard Durbin. Using probabilistic estimation of expression residuals (PEER) to obtain increased power and interpretability of gene expression analyses. *Nature protocols*, 7(3):500–507, 2012. doi: 10.1038/nprot.2011.457.

Matthew Stephens. False discovery rates: a new deal. *Biostatistics*, 18(2):275–294, 10 2016. ISSN 1465-4644. doi: 10.1093/biostatistics/kxw041.

Lei Sun and Matthew Stephens. Solving the empirical Bayes normal means problem with correlated noise. *arXiv preprint arXiv:1812.07488*, 2018. URL `https://arxiv.org/abs/1812.07488`.

Yunting Sun, Nancy R. Zhang, and Art B. Owen. Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *Ann. Appl. Stat.*, 6(4):1664–1688, 12 2012. doi: 10.1214/12-AOAS561.

Valentine Svensson. Droplet scRNA-seq is not zero-inflated. *bioRxiv*, 2019. doi: 10.1101/582064.

Min Tang, Jianqiang Sun, Kentaro Shimizu, and Koji Kadota. Evaluation of methods for differential expression analysis on multi-group RNA-seq count data. *BMC bioinformatics*, 16(1):360, 2015. doi: 10.1186/s12859-015-0794-7.

Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. doi: 10.1111/1467-9868.00196.

Mark A. Van De Wiel, Gwenaël G.R. Leday, Luba Pardo, Håvard Rue, Aad W. Van Der Vaart, and Wessel N.

Van Wieringen. Bayesian analysis of RNA sequencing data by estimating multiple shrinkage priors. *Biostatistics*, 14(1):113–128, 09 2012. ISSN 1465-4644. doi: 10.1093/biostatistics/kxs031.

Mark A. van de Wiel, Maarten Neerincx, Tineke E. Buffart, Daoud Sie, and Henk MW Verheul. ShrinkBayes: a versatile R-package for analysis of count-based sequencing data in complex study designs. *BMC Bioinformatics*, 15(1):116, Apr 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-116.

Koen Van den Berge, Fanny Perraudeau, Charlotte Soneson, Michael I. Love, Davide Risso, Jean-Philippe Vert, Mark D. Robinson, Sandrine Dudoit, and Lieven Clement. Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biology*, 19(1):24, Feb 2018. ISSN 1474-760X. doi: 10.1186/s13059-018-1406-4.

Beate Vieth, Christoph Ziegenhain, Swati Parekh, Wolfgang Enard, and Ines Hellmann. powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics*, 33(21):3486–3488, 07 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx435.

Jon Wakefield. Bayes factors for genome-wide association studies: comparison with p-values. *Genetic Epidemiology*, 33(1):79–86, 2009. ISSN 1098-2272. doi: 10.1002/gepi.20359.

Jingshu Wang, Qingyuan Zhao, Trevor Hastie, and Art B. Owen. Confounder adjustment in multiple hypothesis testing. *Ann. Statist.*, 45(5):1863–1894, 10 2017. doi: 10.1214/16-AOS1511.

Tianyu Wang and Sheida Nabavi. SigEMD: A powerful method for differential gene expression analysis in single-cell RNA sequencing data. *Methods*, 145:25 – 32, 2018. ISSN 1046-2023. doi: 10.1016/j.ymeth.2018.04.017.

Tianyu Wang, Boyang Li, Craig E. Nelson, and Sheida Nabavi. Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. *BMC Bioinformatics*, 20(1):40, Jan 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-2599-6.

Wei Wang and Matthew Stephens. Empirical Bayes matrix factorization. *arXiv preprint arXiv:1802.06931*, 2018. URL `https://arxiv.org/abs/1802.06931`.

Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57, 2009. doi: 10.1038/nrg2484.

M West. Bayesian factor regression models in the "large p, small n" paradigm. In JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West, editors, *Bayesian Statistics 7*, Proceedings of the Seventh Valencia International Meeting, pages 733–742, Oxford, UK, 2003. Clarendon Press.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`.

Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 04 2009. ISSN 1465-4644. doi: 10.1093/biostatistics/kxp008.

Zhijin Wu and Martin J Aryee. Subset quantile normalization using negative control features. *Journal of Computational Biology*, 17(10):1385–1395, 2010. doi: 10.1089/cmb.2010.0049.

Can Yang, Lin Wang, Shuqin Zhang, and Hongyu Zhao. Accounting for non-genetic factors by low-rank representation and sparse regression for eQTL mapping. *Bioinformatics*, 29(8):1026–1034, 2013. doi: 10.1093/bioinformatics/btt075.

Dan Yang, Zongming Ma, and Andreas Buja. A sparse singular value decomposition method for high-dimensional data. *Journal of Computational and Graphical Statistics*, 23(4):923–942, 2014. doi: 10.1080/10618600.2013.858632.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, 18(1):174, Sep 2017. ISSN 1474-760X. doi: 10.1186/s13059-017-1305-0.

Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006. doi: 10.1198/106186006X113430.