A Unified Model for Joint Normalization and Differential Gene Expression Detection in RNA-Seq data

Kefei Liu, Jieping Ye, *Senior Member, IEEE,* Yang Yang, Li Shen, *Senior Member, IEEE,* and Hui Jiang

Abstract—The RNA-sequencing (RNA-seq) is becoming increasingly popular for quantifying gene expression levels. Since the RNA-seq measurements are relative in nature, between-sample normalization of counts is an essential step in differential expression (DE) analysis. The normalization of existing DE detection algorithms is ad hoc and performed once for all prior to DE detection, which may be suboptimal since ideally normalization should be based on non-DE genes only and thus coupled with DE detection. We propose a unified statistical model for joint normalization and DE detection of log-transformed RNA-seq data. Sample-specific normalization factors are modeled as unknown parameters in the gene-wise linear models and jointly estimated with the regression coefficients. By imposing sparsity-inducing L1 penalty (or mixed L1/L2 penalty for multiple treatment conditions) on the regression coefficients, we formulate the problem as a penalized least-squares regression problem and apply the augmented lagrangian method to solve it. Simulation studies show that the proposed model and algorithms perform better than or comparably to existing methods in terms of detection power and false-positive rate. The performance gain increases with increasingly larger sample size or higher signal to noise ratio, and is more significant when a large proportion of genes are differentially expressed in an asymmetric manner.

Index Terms—RNA-Seq, differential expression analysis, normalization, linear regression, L1-Norm regularization, augmented Lagrangian method

1 Introduction

Ultra high-throughput sequencing of transcriptomes (RNA-seq) is a widely used method for quantifying gene expression levels due to its low cost, high accuracy and wide dynamic range for detection [1]. As of today, modern ultra high-throughput sequencing platforms can generate hundreds of millions of sequencing reads from each biological sample in a single day. RNA-seq also facilitates the detection of novel transcripts [2] and the quantification of transcripts on isoform level [3], [4]. For these reasons, RNA-seq has become the method of choice for assaying transcriptomes [5].

One major limitation of RNA-seq is that it only provides relative measurements of transcript abundances due to difference in library size (i.e., sequencing depth) between samples [6]. Normalization of RNA-seq read counts is required in gene differential expression analysis to correct for such variation between samples. A popular form of between-sample normalization is achieved by scaling raw read counts in each sample by a sample-specific factor related to library size [6], [7]. This include CPM/RPM (counts/reads per million) [8], quantile normalization [9], [10], upper-quartile normalization [11], trimmed mean of M values [8] and DESeq normalization [12]. Also, commonly-used gene expression measures, e.g., TPM (transcript per million) [13], and RPKM/FPKM (reads/fragments per kilobase of exon per million mapped reads) [1], [2], also correct for difference in gene length within a sample [14] (the so-called within-sample normalization). In particular, the CPM/RPM (counts/reads per million) [8], TPM (transcript per million) [13], and RPKM/FPKM (reads/fragments per kilobase of exon per million mapped reads) [1], [2] for the i-th gene from the j-th sample are respectively defined as

$$\begin{aligned} \mathrm{cpm}_{ij} &= 10^6 \frac{c_{ij}}{N_j} \\ \mathrm{fpkm}_{ij} &= 10^9 \frac{c_{ij}}{\ell_i \cdot N_j} \\ \mathrm{tpm}_{ij} &= 10^6 \frac{c_{ij}/\ell_i}{\sum_i c_{ij}/\ell_i} \end{aligned} \tag{1}$$

where c_{ij} is the observed read count for gene i from the j-th sample, $N_j = \sum_i c_{ij}$ is the sequencing depth in the j-th sample, and ℓ_i be the length of gene i. In this work we focus on between-sample normalization.

In traditional count-based RNA-seq analysis methods, the read counts for each gene are assumed to follow a Poisson [15] or negative binomial (NB) distribution. One issue with the count-based RNA-seq analysis methods is that their procedures are complicated and contain many ad hoc heuristics. Moreover, the Poisson or NB distributions of counts are mathematically less tractable than the normal distribution [16], [17]. This makes count-based methods difficult to generalize to new data. Moreover, commonly-used statistical methods for microarray data analysis, e.g., quality weighting of RNA samples, addition of random noise to generate technical replicates, and gene set test [16]

- *Kefei Liu and Li Shen are with the Department of Radiology and Imaging Sciences, Indiana University School of Medicine, Indianapolis, IN 46202. E-mail: {kefliu,shenli}@iu.edu*
- *Jieping Ye is with the Department of Computational Medicine and Bioinformatics, University of Michigan, MI 48109. E-mail: jpye@umich.edu*
- *Yang Yang is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China. E-mail: yangyangfuture@buaa.edu.cn*
- *Hui Jiang is with the Department of Biostatistics, University of Michigan, MI 48109. E-mail: jianghui@umich.edu*

This is the author's manuscript of the article published in final edited form as:
Liu, K., Ye, J., Yang, Y., Shen, L., & Jiang, H. (2018). A Unified Model for Joint Normalization and Differential Gene Expression Detection in RNA-Seq data. IEEE/ACM Transactions on Computational Biology and Bioinformatics. http://dx.doi.org/10.1109/TCBB.2018.2790918

have been designed for normally distributed data and it is unclear whether we can adapt them to count data. Also the presence of outliers is an issue that is not well addressed (addressed in a very ad hoc manner) by existing methods. To handle that, the authors of [16] take the logarithm of the raw count of reads and apply normal distribution-based statistical methods to analyze them. Note that by logarithmic transformation, the dynamic range of the RNA-seq counts is compressed such that the outlier counts are largely transformed into "normal" data. As a result, sophisticated way to detect and discard outliers [18], [19], [20] is not required.

In this paper, like in [16], [17] we work with log-transformed gene expression values and propose a unified statistical model for differential gene expression. Different from [16], [17], we model sample-specific scaling factors for between-sample normalization as unknown parameters and incorporate them into the gene-wise linear models. By imposing the sparsity-inducing penalty (L1 penalty for single treatment factor and mixed L1/L2 penalty for multiple treatment factors) on the regression coefficients and carefully choosing the tuning parameter, the model is able to achieve joint accurate detection of DE genes and between-sample normalization. To fit the model, we first eliminate sample-specific parameters using optimization argumentation to formulate the problem as a penalized linear regression problem, and then solve it with the alternating direction method of multipliers algorithm (ADMM), which is known for its fast convergence to modest accuracy [21]. Regarding the choice of tuning parameter, we theoretically derive the smallest tuning parameter $\alpha_{\max}$ that leads to all-zero solution, and thereby find a proper tuning parameter within $[0, \alpha_{\max}]$.

Note that our work is preceded by [22] which address the differential expression problem in a similar way. The difference is that the model of [22] considers only categorical or qualitative predictor/explanatory variables (treatment conditions). For example, label "0" is assigned to samples from the control group and label "1" to samples from the treatment group. While in our model, the predictor/explanatory variables can take arbitrary numeric values, and is thus a generalization of [22] from discrete to continuous predictor-variable model case. Note that the algorithm in [22] does not apply to the current numeric variable model at hand, because (i) applicability: it requires that multiple samples are present in each group but in the continuous-predictor model the concept of "group" no longer exists, or more precisely, each group is formed by only one sample; (ii) algorithmic complexity: it requires an p-dimensional exhaustive search, where p is the number of treatment conditions. When $p > 1$ (see Section 4), the algorithm is computationally very expensive.

The remainder of the paper is organized as follows. In Section 2, we formulate the problem in the context of a single treatment factor. In Section 3, we formulate the problem as a penalized simple regression problem and derive efficient ADMM algorithm to solve it, together with the estimation of noise variance and tuning parameter. In Section 4, we extend the simple regression model to multiple linear regression model. Comparison with existing methods is presented in Section 5, followed by discussions in Section 6.

2 Data model and Problem Formulation

Throughout the paper, the subscript and superscript are used to index the vectors for rows and columns of a matrix, respectively. For example, the i-th row and j-th column vector of a matrix $\boldsymbol{A}$ is denoted as $\boldsymbol{a}_i$ and $\boldsymbol{a}^j$, respectively. Note that this does not conform to conventional notations where the subscript is used to index the columns of a matrix and the superscript is to index the rows.

2.1 Data model

Suppose there are a total of m genes measured in n samples. Let y_{ij}, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, be the log-transformed gene expression measurements (a small positive number is usually added before taking logarithm) for the i-th gene from the j-th sample. The following statistical model is assumed

$$y_{ij} = \beta_{i0} + \beta_i x_j + d_j + \varepsilon_{ij}, \tag{2}$$

where β_{i0} is the y-intercept for gene i, x_j, $j = 1, 2, \ldots, n$, is the predictor variable that represents the treatment condition (e.g., drug dosage) for sample j, β_i is the slope or regression coefficient representing log-fold-change of expression levels of gene i with unit change of x_j, d_j is the scaling factor (e.g., $\log$(sequencing depth) or $\log$(library size)) for sample j for between-sample normalization [6], and $\varepsilon_{ij} \sim \mathcal{N}\left(0, \sigma_i^2\right)$ models the measurement noise. We assume that the error terms ε_{ij} are uncorrelated with the predictor variable and uncorrelated with each other (across both gene i and sample j).

In (2), we consider a single treatment condition. Extension to models with multiple treatment conditions will be discussed in Section 4.

Our main interest is to detect differentially expressed (DE) genes, i.e., whether β_i is equal to zero. If $\beta_i \neq 0$ gene i is differentially expressed across the n samples; otherwise it is not.

Remark 2.1. Since β_{i0} and d_j in (2) respectively model gene-specific factor (e.g., gene length) and sample-specific factor, model (2) is able to work with any log-transformed gene expression measures in the form of

$$y_{ij} = \log \frac{c_{ij}}{\ell_i \cdot q_j}, \tag{3}$$

where c_{ij} is the raw counts, ℓ_i is the length of gene i and q_j is the normalization factor of the j-th sample, since ℓ_i and q_j can be absorbed into β_{i0} and d_j, respectively. Note that gene expression measures of form $c_{ij}/(\ell_i \cdot q_j)$ include the raw counts (with $\ell_i = q_j = 1$), measures based on between-sample normalization only ($\ell_i = 1$) [6], and FPKM and TPM which are shown in (1) and involve both between- and within-sample normalization.

2.2 Penalized likelihood

The likelihood function based on the measured data is given by

$$L\left(\boldsymbol{\beta}_0, \boldsymbol{\beta}, \boldsymbol{d}; \boldsymbol{y}\right) = \prod_{i=1}^{m}\prod_{j=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{-\frac{\left(y_{ij} - \beta_{i0} - \beta_i x_j - d_j\right)^2}{2\sigma_i^2}\right\}, \tag{4}$$

where

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix}.$$

Assume that $\{\sigma_i^2\}_{i=1}^m$ are known, maximization of (4) is equivalent to minimizing the negative log-likelihood:

$$\ell(\boldsymbol{\beta}_0, \boldsymbol{\beta}, \boldsymbol{d}) = \sum_{i=1}^m \sum_{j=1}^n \frac{1}{2\sigma_i^2} (y_{ij} - \beta_{i0} - \beta_i x_j - d_j)^2, \quad (5)$$

where we have ignored the irrelevant constant.

In practice, we solve for $\{\sigma_i^2\}_{i=1}^m$ using an ad hoc approach, which will be described in Section 3.4.

We introduce a L1 penalty on the β_i's:

$$p(\boldsymbol{\beta}) = \alpha \|\boldsymbol{\beta}\|_1 := \alpha \sum_{i=1}^m |\beta_i|. \quad (6)$$

It is well known that the L1 penalty favors sparse solutions (forces some coefficients to be exactly zero) [23]. This is reasonable since in practice many genes are not differentially expressed.

The objective function to be minimized is

$$f(\boldsymbol{\beta}_0, \boldsymbol{\beta}, \boldsymbol{d}) = \sum_{i=1}^m \sum_{j=1}^n \frac{1}{2\sigma_i^2} (y_{ij} - \beta_{i0} - x_j \beta_i - d_j)^2 + \alpha \sum_{i=1}^m |\beta_i|. \quad (7)$$

3 Algorithm Development

3.1 Formulation of (7) as Penalized Simple Linear Regression Model

It can be proved that the optimization problem in (7) is jointly convex in $(\boldsymbol{\beta}_0, \boldsymbol{\beta}, \boldsymbol{d})$. Therefore, the minimizer of (7) is the stationary point.

The derivative of $f(\boldsymbol{\beta}_0, \boldsymbol{\beta}, \boldsymbol{d})$ with respect to d_j, $j = 1, 2, \ldots, n$, is

$$\frac{\partial f}{\partial d_j} = \sum_{i=1}^m -\frac{1}{\sigma_i^2} (y_{ij} - \beta_{i0} - x_j \beta_i - d_j). \quad (8)$$

Setting (8) to zero gives

$$d_j = \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} (y_{ij} - \beta_{i0} - x_j \beta_i). \quad (9)$$

Model (2) is non-identifiable because we can simply add any constant to all the d_j's and subtract the same constant from all the β_{i0}'s, while having the same fit. To resolve this issue, we fix $d_1 = 0$. Therefore

$$d_j = d_j - d_1 = \left(\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)}\right) - (x_j - x_1)\, \bar{\beta}^{(w)}, \quad (10)$$

where

$$\bar{y}_{\cdot j}^{(w)} := \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} y_{ij}, \text{ for } j = 1, 2, \ldots, n, \quad (11)$$

$$\bar{\beta}^{(w)} := \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \frac{1}{\sigma_i^2} \beta_i. \quad (12)$$

Here, the superscript $^{(w)}$ indicates that the mean is a weighted mean instead of an unweighted one.

On the other hand, from

$$\frac{\partial f}{\partial \beta_{i0}} = -\frac{1}{\sigma_i^2} \sum_{j=1}^n (y_{ij} - \beta_{i0} - x_j \beta_i - d_j) = 0, \quad (13)$$

we have

$$\beta_{i0} = \frac{1}{n} \sum_{j=1}^n (y_{ij} - x_j \beta_i - d_j) = \bar{y}_{i\cdot} - \bar{x}\beta_i - \frac{1}{n} \sum_{j=1}^n d_j, \quad (14)$$

where

$$\bar{y}_{i\cdot} := \frac{1}{n} \sum_{j=1}^n y_{ij}, \text{ for } i = 1, 2, \ldots, m, \quad (15)$$

$$\bar{x} := \frac{1}{n} \sum_{j=1}^n x_j. \quad (16)$$

From (10) we have

$$\frac{1}{n} \sum_{j=1}^n d_j = \left(\bar{y}^{(w)} - \bar{y}_{\cdot 1}^{(w)}\right) - (\bar{x} - x_1)\, \bar{\beta}^{(w)}, \quad (17)$$

where

$$\bar{y}^{(w)} := \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}} \sum_{i=1}^m \left(\frac{1}{\sigma_i^2} \cdot \frac{1}{n} \sum_{j=1}^n y_{ij} \right). \quad (18)$$

Substituting (17) into (14) yields

$$\beta_{i0} = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} + (\bar{x} - x_1)\, \bar{\beta}^{(w)} - \bar{x}\beta_i. \quad (19)$$

Without loss of generality, we make the following two assumptions:

Assumption 3.1.

$$\sum_{j=1}^n x_j = n\bar{x} = 0, \quad \sum_{j=1}^n x_j^2 = 1. \quad (20)$$

These assumptions are reasonable since in the model (2) the center and scaling factor of x_j's can be absorbed into β_{i0} and β_i, respectively.

Then (19) simplifies to

$$\beta_{i0} = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - x_1 \bar{\beta}^{(w)}. \quad (21)$$

The sum of (10) and (21) yields

$$\beta_{i0} + d_j = \bar{y}_{i\cdot} + \bar{y}_{\cdot j}^{(w)} - \bar{y}^{(w)} - x_j \bar{\beta}^{(w)}. \quad (22)$$

Substituting (22) into (7), the latter simplifies to

$$f(\boldsymbol{\beta}) = \sum_{i=1}^m \frac{1}{2\sigma_i^2} \sum_{j=1}^n \left(\tilde{y}_{ij} - x_j \beta_i + x_j \bar{\beta}^{(w)}\right)^2 + \alpha \sum_{i=1}^m |\beta_i|, \quad (23)$$

where

$$\tilde{y}_{ij} := y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)}. \quad (24)$$

It can be shown by straightforward calculation that $\{\tilde{y}_{ij}\}$ satisfies

$$\sum_{i=1}^m \frac{1}{\sigma_i^2} \tilde{y}_{ij} = 0, \quad (25)$$

$$\sum_{j=1}^n \tilde{y}_{ij} = 0. \quad (26)$$

3.2 Model Fitting by ADMM

We propose to use the alternating direction method of multipliers (ADMM) [21] to solve (23). Although ADMM can be very slow to converge to high accuracy, it is often the case that ADMM converges to modest accuracy very fast (within a few tens of iterations) [21].

To apply the ADMM, the problem (23) is reformulated as

$$f(\boldsymbol{\beta}) = \sum_{i=1}^{m}\frac{1}{2\sigma_i^2}\sum_{j=1}^{n}\left(\tilde{y}_{ij} - x_j\beta_i + x_j\delta_0\right)^2 + \alpha\sum_{i=1}^{m}|\beta_i|, \quad (27a)$$

subject to

$$\frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\beta_i = \delta_0. \quad (27b)$$

The augmented Lagrangian of (27) is (28) at the bottom of the page.

Step 1: Update β_i, $i = 1, 2, \ldots, m$:

The derivative of (28) with respect to β_i is

$$\begin{aligned}\frac{\partial L_\rho}{\partial \beta_i} =& \frac{1}{\sigma_i^2}\sum_{j=1}^{n} - x_j\left(\tilde{y}_{ij} - x_j\beta_i + x_j\delta_0\right) + \alpha\partial|\beta_i| \\ &+ \frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\frac{1}{\sigma_i^2}\lambda + \rho\frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\frac{1}{\sigma_i^2}\left(\frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}\beta_\ell - \delta_0\right),\end{aligned} \quad (29)$$

where $\partial|\beta_i|$ is the subgradient of $|\beta_i|$ with respect to β_i and is defined as

$$\partial|\beta_i| = \begin{cases} 1, & \beta_i > 0 \\ -1, & \beta_i < 0 \\ [-1, 1], & \beta_i = 0 \end{cases}$$

Setting (29) equal to zero gives (30) at the bottom of the page, where T is the soft-thresholding operator:

$$T_{\sigma_i^2\alpha}[x] \coloneqq \operatorname{sign}(x)\left(|x| - \sigma_i^2\alpha\right)_+ = \begin{cases} x - \sigma_i^2\alpha, & x > \sigma_i^2\alpha \\ x + \sigma_i^2\alpha, & x < -\sigma_i^2\alpha \\ 0, & -\sigma_i^2\alpha \le x \le \sigma_i^2\alpha \end{cases}$$

Step 2: Update δ_0:

The derivative of (28) with respect to δ_0 is

$$\begin{aligned}\frac{\partial L_\rho}{\partial \delta_0} =& \sum_{i=1}^{m}\frac{1}{\sigma_i^2}\sum_{j=1}^{n}x_j\left(\tilde{y}_{ij} - x_j\beta_i + x_j\delta_0\right) \\ &- \lambda + \rho\left(\delta_0 - \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\beta_i\right).\end{aligned} \quad (31)$$

Setting (31) equal to zero gives

$$\begin{aligned}\delta_0 &= \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2} + \rho}\left(\lambda - \sum_{i=1}^{m}\frac{1}{\sigma_i^2}\sum_{j=1}^{n}x_j\tilde{y}_{ij}\right) + \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\beta_i \\ &= \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2} + \rho}\lambda + \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\beta_i,\end{aligned} \quad (32)$$

where the second equality is due to (25).

Step 3: Update λ:

$$\lambda^{\text{new}} = \lambda^{\text{old}} + \rho\left(\frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\beta_i - \delta_0\right) \quad (33)$$

The model fitting algorithm is described in Algorithm 1.

3.3 Estimation of Tuning Parameter α

Eq. (23) can be expressed in matrix form as

$$f(\boldsymbol{\beta}) = \frac{1}{2}\left\|\boldsymbol{\Sigma}^{1/2}\left(\widetilde{\boldsymbol{Y}} - \boldsymbol{M}\boldsymbol{\beta}\boldsymbol{x}^{\mathrm{T}}\right)\right\|_{\mathrm{F}}^2 + \alpha\|\boldsymbol{\beta}\|_1, \quad (34)$$

where

$$\boldsymbol{\Sigma} = \operatorname{diag}\{\boldsymbol{\sigma}\}, \quad (35)$$

with

$$\boldsymbol{\sigma} = \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \cdots & 1/\sigma_m^2 \end{pmatrix}^{\mathrm{T}}, \quad (36)$$

and

$$\begin{aligned}\boldsymbol{M} &= \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix} - \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \cdots & 1/\sigma_m^2 \\ 1/\sigma_1^2 & 1/\sigma_2^2 & \cdots & 1/\sigma_m^2 \\ \vdots & \vdots & \ddots & \vdots \\ 1/\sigma_1^2 & 1/\sigma_2^2 & \cdots & 1/\sigma_m^2 \end{pmatrix} \\ &= \boldsymbol{I}_m - \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\boldsymbol{1}_m\boldsymbol{\sigma}^{\mathrm{T}}.\end{aligned} \quad (37)$$

After expansion, (34) becomes

$$f(\boldsymbol{\beta}) = \frac{1}{2}\left\|\boldsymbol{\Sigma}^{1/2}\widetilde{\boldsymbol{Y}}\right\|_{\mathrm{F}}^2 - \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\boldsymbol{x} + \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\boldsymbol{\beta} + \alpha\|\boldsymbol{\beta}\|_1, \quad (38)$$

where we exploit the assumption $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$.

Since $\frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\boldsymbol{\beta} \ge 0$ with equality occurring at $\boldsymbol{\beta} = \boldsymbol{0}$, it is shown that $\hat{\boldsymbol{\beta}} = \boldsymbol{0}$ is the minimizer of $f(\boldsymbol{\beta})$ when

$$\alpha \ge \left\|\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\boldsymbol{x}\right\|_\infty \coloneqq \max_{1\le i\le m}\left|{\boldsymbol{m}^i}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\boldsymbol{x}\right|, \quad (39)$$

where $\boldsymbol{m}^i$ denotes the i-th column of $\boldsymbol{M}$ in (37).

$$L_\rho(\boldsymbol{\beta}, \delta_0, \lambda) = \sum_{i=1}^{m}\frac{1}{2\sigma_i^2}\sum_{j=1}^{n}\left(\tilde{y}_{ij} - x_j\beta_i + x_j\delta_0\right)^2 + \alpha\sum_{i=1}^{m}|\beta_i| + \lambda\left(\frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\beta_i - \delta_0\right) + \frac{\rho}{2}\left(\frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\beta_i - \delta_0\right)^2 \quad (28)$$

$$\beta_i = \frac{\sigma_i^2\left(\sum_{\ell=1}^{m}\sigma_\ell^{-2}\right)^2}{\sigma_i^2\left(\sum_{\ell=1}^{m}\sigma_\ell^{-2}\right)^2 + \rho}T_{\sigma_i^2\alpha}\left[\left(\sum_{j=1}^{n}x_j\tilde{y}_{ij} + \delta_0\right) - \frac{\rho}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\left(\frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\sum_{\ell\ne i}\frac{1}{\sigma_\ell^2}\beta_\ell - \delta_0 + \frac{\lambda}{\rho}\right)\right] \quad (30)$$

Algorithm 1 ADMM algorithm for fitting the simple linear regression model

Input: Log-transformed gene expression measurements: $\{\{y_{ij}\}_{i=1}^m\}_{j=1}^n$, predictor variables: $\{x_j\}_{j=1}^n$ and estimated noise variance: $\{\sigma_i^2\}_{i=1}^m$.

Transform data. Normalize $\{x_j\}_{j=1}^n$ to zero mean and unit norm:

$$\tilde{x}_j = \frac{x_j - \bar{x}}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \text{ with } \bar{x} \coloneqq \frac{1}{n}\sum_{j=1}^n x_j.$$

Center y_{ij} to zero mean over row index i and column index j:

$$\tilde{y}_{ij} = y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)},$$

where $\bar{y}_{i\cdot}$, $\bar{y}_{\cdot j}^{(w)}$ and $\bar{y}^{(w)}$ are defined in (15), (11) and (18), respectively.

Set the penalty parameter to $\rho = 1$ [21]; select the tuning parameter α according to Section 3.3.

Initialization: $k = 0$; randomly initialize $\beta_i = \beta_i^0$, $i = 1, 2, \ldots, m$, $\delta_0 = \delta_0^0$, and $\lambda = \lambda^0$.

repeat

Update β_i, $i = 1, 2, \ldots, m$:

$$\beta_i^{k+1} = \frac{\sigma_i^2\left(\sum_{\ell=1}^m \sigma_\ell^{-2}\right)^2}{\sigma_i^2\left(\sum_{\ell=1}^m \sigma_\ell^{-2}\right)^2 + \rho} T_{\sigma_i^2\alpha}\left[\left(\sum_{j=1}^n \tilde{x}_j\tilde{y}_{ij} + \delta_0^k\right) - \frac{\rho}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}}\left(\frac{1}{\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}}\sum_{\ell\neq i}\frac{1}{\sigma_\ell^2}\beta_\ell^k - \delta_0^k + \frac{\lambda^k}{\rho}\right)\right]$$

Update δ_0:

$$\delta_0^{k+1} = \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2} + \rho}\lambda^k + \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}}\sum_{i=1}^m \frac{1}{\sigma_i^2}\beta_i^{k+1}$$

Update λ:

$$\lambda^{k+1} = \lambda^k + \rho\left(\frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}}\sum_{i=1}^m \frac{1}{\sigma_i^2}\beta_i^{k+1} - \delta_0^{k+1}\right)$$

$k \leftarrow k + 1$;

until convergence or maximum number of iterations is reached.

Output: $\beta_i = \beta_i^k$, $i = 1, 2, \ldots, m$, $\bar{\beta}^{(w)} = \delta_0^k$, and

$$\beta_{i0} = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - \tilde{x}_1\bar{\beta}^{(w)},\ i = 1, 2, \ldots, m$$

$$d_1 = 0, \quad d_j = \left(\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)}\right) - (\tilde{x}_j - \tilde{x}_1)\,\bar{\beta}^{(w)},\ j = 2, \ldots, n$$

Recover the original parameter space:

$$\beta_{i0}' = \beta_{i0} - \frac{\beta_i\bar{x}}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \quad \beta_i' = \frac{\beta_i}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \quad i = 1, 2, \ldots, m.$$

Note that

$$\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}} = \left(\boldsymbol{I}_m - \frac{1}{\sum_{i=1}^m \frac{1}{\sigma_i^2}}\boldsymbol{\sigma}\boldsymbol{1}_m^{\mathrm{T}}\right)\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}} = \boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}, \tag{40}$$

where the last equality holds because $\boldsymbol{1}_m^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}} = \boldsymbol{0}$ due to (25).

Substituting (40) into (39) yields

$$\alpha_{\max} = \left\|\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\boldsymbol{x}\right\|_\infty = \max_i \left|\frac{1}{\sigma_i^2}\boldsymbol{x}^{\mathrm{T}}\tilde{\boldsymbol{y}}_i\right|. \tag{41}$$

Our strategy is to first sort $\left|\frac{1}{\sigma_1^2}\boldsymbol{x}^{\mathrm{T}}\tilde{\boldsymbol{y}}_1\right|, \left|\frac{1}{\sigma_2^2}\boldsymbol{x}^{\mathrm{T}}\tilde{\boldsymbol{y}}_2\right|, \ldots, \left|\frac{1}{\sigma_m^2}\boldsymbol{x}^{\mathrm{T}}\tilde{\boldsymbol{y}}_m\right|$ in ascending order, from least to greatest, and then set α as the P-th percentile ($0 < P < 100$) of the m ordered values. We set $P = 5$ in Section 5.

3.4 Maximum likelihood estimation of $\{\sigma_i^2\}_{i=1}^m$

To solve for $\{\sigma_i^2\}_{i=1}^m$, consider the negative log-likelihood function in (4) with $\{\sigma_i^2\}_{i=1}^m$ being unknown parameters as well:

$$\ell\left(\boldsymbol{\beta}_0, \boldsymbol{\beta}, \boldsymbol{d}, \{\sigma_i^2\}_{i=1}^m\right) = \sum_{i=1}^m\left[\frac{n}{2}\log(2\pi\sigma_i^2) + \frac{1}{2\sigma_i^2}\sum_{j=1}^n (y_{ij} - \beta_{i0} - x_j\beta_i - d_j)^2\right. \tag{42}$$

Taking the partial derivatives of $\ell(\cdot)$ with respect to d_j and β_{i0} and setting the results to zero, we arrive at (10) and (21) respectively. The sum of (10) and (21) gives (22).

Taking the partial derivative of $\ell(\cdot)$ with respect to β_i and setting the result to zero, we have

$$\beta_i = \sum_{j=1}^n x_j y_{ij} - \sum_{j=1}^n x_j\left(\beta_{i0} + d_j\right). \tag{43}$$

Substituting (22) into (43) yields

$$\beta_i = \sum_{j=1}^{n} x_j y_{ij} - \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}} \sum_{i=1}^{m}\frac{1}{\sigma_i^2}\sum_{j=1}^{n} x_j y_{ij} + \bar{\beta}^{(w)}, \tag{44}$$

where $\bar{\beta}^{(w)}$ is defined in (12).

Taking the partial derivative of σ_i^2 and setting the result to zero gives

$$\sigma_i^2 = \frac{1}{n}\sum_{j=1}^{n}\left(y_{ij} - \beta_{i0} - x_j\beta_i - d_j\right)^2. \tag{45}$$

Substituting (22) into (45) yields

$$\sigma_i^2 = \frac{1}{n}\sum_{j=1}^{n}\left(y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)} - x_j\beta_i + x_j\bar{\beta}^{(w)}\right)^2, \tag{46}$$

where $\bar{y}_{i\cdot}$, $\bar{y}_{\cdot j}^{(w)}$ and $\bar{y}^{(w)}$ are defined in (15), (11) and (18), respectively.

Given initial estimates for $\bar{\beta}^{(w)}$ and $\{\sigma_i^2\}_{i=1}^m$, we can alternate equations (44), (46) and (12) iteratively to gradually refine the estimates for β_i and σ_i^2, as shown in Algorithm 2.

Algorithm 2 Maximum likelihood estimation of $\{\sigma_i^2\}_{i=1}^m$ for simple linear regression model

Input: Log-transformed gene expression measurements: $\{\{y_{ij}\}_{i=1}^m\}_{j=1}^n$ and predictor variables: $\{x_j\}_{j=1}^n$.

1: Normalize $\{x_j\}_{j=1}^n$ to zero mean and unit norm:

$$x_j \leftarrow \frac{x_j - \bar{x}}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \text{ with } \bar{x} \coloneqq \frac{1}{n}\sum_{j=1}^n x_j.$$

2: *Initialization*: $\bar{\beta}^{(w)} = 0$, $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_m^2 = 1$.
3: **repeat**
4: Update β_i, $i = 1, 2, \ldots, m$, according to (44);
5: Update σ_i^2, $i = 1, 2, \ldots, m$, according to (46);
6: Update $\bar{\beta}^{(w)}$ according to (12);
7: **until** convergence or maximum number of iterations is reached.

Output: $\hat{\sigma}_i^2 = \sigma_i^2$, $i = 1, 2, \ldots, m$.

To obtain a robust estimate for σ_i^2, we further take the weighted average of $\hat{\sigma}_i^2$ and the estimated mean variance across all the genes. That is

$$\hat{\sigma}_i'^2 = (1-w)\hat{\sigma}_i^2 + w\overline{\hat{\sigma}^2} \tag{47}$$

where

$$\overline{\hat{\sigma}^2} = \frac{1}{m}\sum_{i=1}^m \hat{\sigma}_i^2, \tag{48}$$

and the weight w is calculated using the following formula which is derived based on an empirical Bayes approach [24]

$$w = \frac{2(m-1)}{n+1}\left(\frac{1}{m} + \frac{(\overline{\hat{\sigma}^2})^2}{\sum_{i=1}^m\left(\hat{\sigma}_i^2 - \overline{\hat{\sigma}^2}\right)^2}\right). \tag{49}$$

This kind of variance estimation approach is widely used in differential gene expression analysis with small sample sizes [25], [26]. The estimated variances $\hat{\sigma}_i'^2$, $i = 1, 2, \ldots, m$, can then be used in Algorithm 1 to solve for $\{\beta_i\}_{i=1}^m$.

Remark 3.1. In the special case of $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_m^2 = \sigma^2$, it no longer requires to estimate σ^2 since the unknown σ^2 in (7) can be absorbed into the tuning parameter α.

4 Extension to Multiple Linear Regression Model and Algorithm Development

In the multiple linear regression model, each response or outcome is modeled by $p > 1$ predictors:

$$y_{ij} = \beta_{i0} + \boldsymbol{\beta}_i^{\mathrm{T}}\boldsymbol{x}_j + d_j + \varepsilon_{ij}, \tag{50}$$

where

$$\boldsymbol{\beta}_i = \begin{pmatrix} \beta_{i1} \\ \beta_{i2} \\ \vdots \\ \beta_{ip} \end{pmatrix} \in \mathbb{R}^{p\times 1} \tag{51}$$

is a vector of regression coefficients representing log-fold-change of expression levels of gene i between treatment conditions, and

$$\boldsymbol{x}_j = \begin{pmatrix} x_{j1} \\ x_{j2} \\ \vdots \\ x_{jp} \end{pmatrix} \in \mathbb{R}^{p\times 1} \tag{52}$$

is a vector of predictors representing the treatment conditions (drug dosage, blood pressure, age, BMI, etc.) for sample j, and β_{i0}, d_j and $\varepsilon_{ij} \sim \mathcal{N}\left(0, \sigma_i^2\right)$ are the y-intercept, scaling factor for sample j and measurement noise, respectively.We assume that the error terms ε_{ij} are uncorrelated with all the predictor variables and uncorrelated with each other.

The likelihood function based on the observed data is given by

$$L\left(\boldsymbol{\beta}_0, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{d}; \boldsymbol{Y}\right) = \prod_{i=1}^m\prod_{j=1}^n \frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left\{-\frac{\left(y_{ij} - \beta_{i0} - \boldsymbol{\beta}_i^{\mathrm{T}}\boldsymbol{x}_j - d_j\right)^2}{2\sigma_i^2}\right\}. \tag{53}$$

Assume that $\{\sigma_i^2\}_{i=1}^m$ are known, maximization of (53) leads to minimizing the negative log-likelihood:

$$\ell\left(\boldsymbol{\beta}_0, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{d}\right) = \sum_{i=1}^m\sum_{j=1}^n \frac{1}{2\sigma_i^2}\left(y_{ij} - \beta_{i0} - \boldsymbol{\beta}_i^{\mathrm{T}}\boldsymbol{x}_j - d_j\right)^2 \tag{54}$$

The objective function to be minimized is

$$f\left(\boldsymbol{\beta}_0, \{\boldsymbol{\beta}_i\}, \boldsymbol{d}\right) = \sum_{i=1}^m\sum_{j=1}^n \frac{1}{2\sigma_i^2}\left(y_{ij} - \beta_{i0} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i - d_j\right)^2 + \sum_{i=1}^m p\left(\boldsymbol{\beta}_i\right). \tag{55}$$

Below we introduce two types of penalty function $p\left(\boldsymbol{\beta}_i\right)$.

1) Type I penalty:

$$p\left(\boldsymbol{\beta}_i\right) = \alpha\left|\beta_{ip}\right|. \tag{56}$$

Gene i is differentially expressed if $\beta_{ip} \neq 0$ and not otherwise. This penalty is for the applications where one covariate is of main interest (e.g., treatment) while we want to adjust for all possible effects of other confounding covariates (e.g., age, gender, etc).

2) Type II penalty:

$$p\left(\boldsymbol{\beta}_i\right) = \alpha\left\|\boldsymbol{\beta}_i\right\|. \tag{57}$$

Gene i is differentially expressed if $\boldsymbol{\beta}_i \neq 0$ and not otherwise. This penalty is for the applications where all covariates are of interest and we want to identify the genes for which at least one covariate has an effect.

It can be proved that the optimization problem (55) with penalty (56) or (57) is jointly convex in $(\boldsymbol{\beta}_0, \{\boldsymbol{\beta}_i\}, \boldsymbol{d})$.

Assume that

$$\sum_{j=1}^{n} \boldsymbol{x}_j = 0, \tag{58}$$

and set $d_1 = 0$. Using similar argumentation as in Section 3.1 to eliminate $\boldsymbol{\beta}_0$ and $\boldsymbol{d}$, we simplify (55) to

$$f(\{\boldsymbol{\beta}_i\}) = \sum_{i=1}^{m} \frac{1}{2\sigma_i^2} \sum_{j=1}^{n} \left(\tilde{y}_{ij} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i + \boldsymbol{x}_j^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}\right)^2 + \sum_{i=1}^{m} p(\boldsymbol{\beta}_i), \tag{59}$$

where $\tilde{y}_{ij}$ is the same as that in (24), and

$$\bar{\boldsymbol{\beta}}^{(w)} \coloneqq \frac{1}{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}} \sum_{i=1}^{m} \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i. \tag{60}$$

4.1 Regression with type I penalty: Model fitting by ADMM

To apply the ADMM, we reformulate the Type I penalized regression problem as

$$f(\{\boldsymbol{\beta}_i\}, \boldsymbol{\delta}_0) = \sum_{i=1}^{m} \frac{1}{2\sigma_i^2} \sum_{j=1}^{n} \left(\tilde{y}_{ij} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i + \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\delta}_0\right)^2 + \alpha \sum_{i=1}^{m} |\beta_{ip}|, \tag{61a}$$

subject to

$$\frac{1}{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}} \sum_{i=1}^{m} \frac{1}{\sigma_i^2} \boldsymbol{\beta}_i = \boldsymbol{\delta}_0. \tag{61b}$$

The augmented Lagrangian of (61) is (62) at the bottom of the page.

Step 1: Update $\boldsymbol{\beta}_i$, $i = 1, 2, \ldots, m$:

Taking the partial derivative of (62) with respect to $\boldsymbol{\beta}_i$ and setting the result to zero gives

$$\left[\frac{1}{\sigma_i^2}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \frac{\rho}{\sigma_i^4} \frac{1}{\left(\sum_{\ell=1}^{m} \frac{1}{\sigma_\ell^2}\right)^2} \boldsymbol{I}_p\right] \boldsymbol{\beta}_i + \alpha \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \partial|\beta_{ip}| \end{pmatrix} = \boldsymbol{v}_i, \tag{63}$$

where

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \in \mathbb{R}^{n \times p}, \tag{64}$$

$\partial|\beta_{ip}|$ is the subgradient of $|\beta_{ip}|$ with respect to β_{ip}, and

$$\begin{aligned} \boldsymbol{v}_i = & \frac{1}{\sigma_i^2}\left(\sum_{j=1}^{n} \boldsymbol{x}_j \tilde{y}_{ij} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\delta}_0\right) \\ & - \frac{\rho}{\sigma_i^2} \frac{1}{\sum_{\ell=1}^{m} \frac{1}{\sigma_\ell^2}} \left(\frac{1}{\sum_{\ell=1}^{m} \frac{1}{\sigma_\ell^2}} \sum_{\ell \neq i} \frac{1}{\sigma_\ell^2} \boldsymbol{\beta}_\ell - \boldsymbol{\delta}_0 + \frac{\boldsymbol{\lambda}}{\rho}\right). \end{aligned} \tag{65}$$

Given matrix partition in the following form:

$$\frac{1}{\sigma_i^2}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \frac{\rho}{\sigma_i^4} \frac{1}{\left(\sum_{\ell=1}^{m} \frac{1}{\sigma_\ell^2}\right)^2} \boldsymbol{I}_p = \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{q} \\ \boldsymbol{q}^{\mathrm{T}} & q_{pp} \end{pmatrix},$$

$$\boldsymbol{\beta}_i = \begin{pmatrix} \boldsymbol{\beta}_i^- \\ \beta_{ip} \end{pmatrix}, \quad \boldsymbol{v}_i = \begin{pmatrix} \boldsymbol{v}_i^- \\ v_{ip} \end{pmatrix},$$

where $\boldsymbol{Q}_{11}$ is the submatrix of $\boldsymbol{Q}$ with last row and last column deleted, from (63) we have

$$\boldsymbol{Q}_{11}\boldsymbol{\beta}_i^- + \boldsymbol{q}\beta_{ip} = \boldsymbol{v}_i^- \tag{66}$$

$$\boldsymbol{q}^{\mathrm{T}}\boldsymbol{\beta}_i^- + q_{pp}\beta_{ip} + \alpha\partial|\beta_{ip}| = v_{ip}. \tag{67}$$

From (66) it follows

$$\boldsymbol{\beta}_i^- = \boldsymbol{Q}_{11}^{-1}\left(\boldsymbol{v}_i^- - \boldsymbol{q}\beta_{ip}\right). \tag{68}$$

Substituting (68) into (67) yields

$$\beta_{ip} = \frac{1}{q_{pp} - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{Q}_{11}^{-1}\boldsymbol{q}} T_\alpha\left[v_{ip} - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{Q}_{11}^{-1}\boldsymbol{v}_i^-\right]. \tag{69}$$

Step 2: Update $\boldsymbol{\delta}_0$:

Taking the derivative of (62) with respect to $\boldsymbol{\delta}_0$ and setting the result to zero gives

$$\boldsymbol{\delta}_0 = \left(\sum_{i=1}^{m} \frac{1}{\sigma_i^2}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \rho\boldsymbol{I}_p\right)^{-1} \boldsymbol{\lambda} + \frac{1}{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}} \sum_{i=1}^{m} \frac{1}{\sigma_i^2}\boldsymbol{\beta}_i, \tag{70}$$

where we have exploited (25).

Step 3: Update $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}^{\text{new}} = \boldsymbol{\lambda}^{\text{old}} + \rho\left(\frac{1}{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}} \sum_{i=1}^{m} \frac{1}{\sigma_i^2}\boldsymbol{\beta}_i - \boldsymbol{\delta}_0\right). \tag{71}$$

The model fitting algorithm is described in Algorithm 3.

4.2 Regression with type II penalty: Model fitting by ADMM

The Type II penalized regression problem is reformulated as

$$f(\{\boldsymbol{\beta}_i\}, \boldsymbol{\delta}_0) = \sum_{i=1}^{m} \frac{1}{2\sigma_i^2} \sum_{j=1}^{n} \left(\tilde{y}_{ij} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i + \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\delta}_0\right)^2 + \alpha \sum_{i=1}^{m} \|\boldsymbol{\beta}_i\|, \tag{72a}$$

subject to

$$\frac{1}{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}} \sum_{i=1}^{m} \frac{1}{\sigma_i^2}\boldsymbol{\beta}_i = \boldsymbol{\delta}_0. \tag{72b}$$

The augmented Lagrangian of (72) is (73) at the bottom of the page.

Step 1: Update $\boldsymbol{\beta}_i$, $i = 1, 2, \ldots, m$:

The relevant terms to compute the derivatives of (73) with respect to $\boldsymbol{\beta}_i$ is (74) at the bottom of the page, where c is an irrelevant constant which does not depend on $\boldsymbol{\beta}_i$, and $\boldsymbol{v}_i$ is defined in (65).

$$L_\rho(\{\boldsymbol{\beta}_i\}, \boldsymbol{\delta}_0, \boldsymbol{\lambda}) = \sum_{i=1}^{m} \frac{1}{2\sigma_i^2} \sum_{j=1}^{n} \left(\tilde{y}_{ij} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i + \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\delta}_0\right)^2 + \alpha \sum_{i=1}^{m} |\beta_{ip}| + \boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{1}{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}} \sum_{i=1}^{m} \frac{1}{\sigma_i^2}\boldsymbol{\beta}_i - \boldsymbol{\delta}_0\right) + \frac{\rho}{2}\left\|\frac{1}{\sum_{i=1}^{m} \frac{1}{\sigma_i^2}} \sum_{i=1}^{m} \frac{1}{\sigma_i^2}\boldsymbol{\beta}_i - \boldsymbol{\delta}_0\right\|^2. \tag{62}$$

It can be shown that when $\|\boldsymbol{v}_i\| \leq \alpha$ then $\boldsymbol{\beta}_i = \mathbf{0}$; otherwise denote the eigendecomposition of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ as $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathrm{T}}$, we have that minimization of (74) is equivalent to

$$\min_{\boldsymbol{\beta}_i} \frac{1}{2}\left\|\boldsymbol{Z}_i\boldsymbol{\beta}_i - \boldsymbol{b}_i\right\|^2 + \alpha\left\|\boldsymbol{\beta}_i\right\|, \tag{75a}$$

where

$$\boldsymbol{Z}_i = \left[\frac{1}{\sigma_i^2}\boldsymbol{D} + \frac{\rho}{\sigma_i^4}\frac{1}{\left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}\right)^2}\boldsymbol{I}_p\right]^{1/2}\boldsymbol{U}^{\mathrm{T}}, \tag{75b}$$

$$\boldsymbol{b}_i = \left[\frac{1}{\sigma_i^2}\boldsymbol{D} + \frac{\rho}{\sigma_i^4}\frac{1}{\left(\sum_{\ell=1}^m \frac{1}{\sigma_\ell^2}\right)^2}\boldsymbol{I}_p\right]^{-1/2}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{v}_i. \tag{75c}$$

As in [27], we use a coordinate descent procedure to optimize (75). For each s, given the estimate of $\{\hat{\beta}_{i\ell}\}_{\ell\neq s}$, β_{is} can be estimated by solving

$$\min_{\beta_{is}} \frac{1}{2}\left\|\boldsymbol{z}^s\beta_{is} - \boldsymbol{r}_i^{(s)}\right\|^2 + \alpha\sqrt{\beta_{is}^2 + \sum_{\ell\neq s}\hat{\beta}_{i\ell}^2}, \tag{76}$$

where

$$\boldsymbol{r}_i^{(s)} = \boldsymbol{b}_i - \sum_{\ell\neq s}\boldsymbol{z}^\ell\hat{\beta}_{i\ell}. \tag{77}$$

We solve (76) via a one-dimensional search. Note that the solution to (76) falls between 0 and $\beta_{i\ell}^{\mathrm{o}} = \boldsymbol{z}^{s\mathrm{T}}\boldsymbol{r}_i^{(s)}/\left\|\boldsymbol{z}^s\right\|^2$, the ordinary least-squares estimate. We can use the `optimize` function in the R package, or `fminbnd` function in MATLAB, which performs one-dimensional search based on golden section search and successive parabolic interpolation.

After updating $\{\boldsymbol{\beta}_i\}_{i=1}^m$, the updates of $\boldsymbol{\delta}_0$ and $\boldsymbol{\lambda}$ turn out to be the same as that in Section 4.1. The model fitting algorithm is described in Algorithm 4.

4.3 Estimation of Tuning Parameter α

Eq. (59) can be expressed in matrix form as

$$f(\boldsymbol{B}) = \frac{1}{2}\left\|\boldsymbol{\Sigma}^{1/2}\left(\widetilde{\boldsymbol{Y}} - \boldsymbol{M}\boldsymbol{B}\boldsymbol{X}^{\mathrm{T}}\right)\right\|_{\mathrm{F}}^2 + p(\boldsymbol{B}), \tag{78}$$

where $\boldsymbol{M}$ and $\boldsymbol{X}$ are respectively defined in (37) and (64), and

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{\beta}_1^{\mathrm{T}} \\ \boldsymbol{\beta}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{\beta}_m^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1p} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{m1} & \beta_{m2} & \cdots & \beta_{mp} \end{pmatrix} \in \mathbb{R}^{m\times p}, \tag{79}$$

and $p(\boldsymbol{B})$ is the penalty function.

The derivative of $f(\boldsymbol{B})$ with respect to $\boldsymbol{B}$ is

$$\frac{\partial f}{\partial \boldsymbol{B}} = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\boldsymbol{B}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\boldsymbol{X} + \frac{\partial p(\boldsymbol{B})}{\partial \boldsymbol{B}}. \tag{80}$$

4.3.1 Type I Penalty

When $p(\boldsymbol{B}) = \alpha\sum_{i=1}^m |\beta_{ip}|$, its derivative with respect to $\boldsymbol{B}$ is

$$\frac{\partial p(\boldsymbol{B})}{\partial \boldsymbol{B}} = \alpha\begin{pmatrix} 0 & \cdots & 0 & \partial|\beta_{1p}| \\ 0 & \cdots & 0 & \partial|\beta_{2p}| \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \partial|\beta_{mp}| \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{m\times(p-1)} & \alpha\frac{\partial\|\boldsymbol{\beta}^p\|_1}{\partial\boldsymbol{\beta}^p} \end{pmatrix}. \tag{81}$$

Denote

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^1 & \cdots & \boldsymbol{x}^{p-1} & \boldsymbol{x}^p \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{x}^p \end{bmatrix},$$
$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{\beta}^1 & \cdots & \boldsymbol{\beta}^{p-1} & \boldsymbol{\beta}^p \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{\beta}^p \end{bmatrix}.$$

Setting (80) equal to zero gives

$$\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\left(\boldsymbol{B}_1\boldsymbol{X}_1^{\mathrm{T}} + \boldsymbol{\beta}^p\boldsymbol{x}^{p\mathrm{T}}\right)\boldsymbol{X}_1 = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\boldsymbol{X}_1 \tag{82}$$

$$\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\left(\boldsymbol{B}_1\boldsymbol{X}_1^{\mathrm{T}} + \boldsymbol{\beta}^p\boldsymbol{x}^{p\mathrm{T}}\right)\boldsymbol{x}^p + \alpha\frac{\partial\|\boldsymbol{\beta}^p\|_1}{\partial\boldsymbol{\beta}^p} = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\boldsymbol{x}^p. \tag{83}$$

Since $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}$ is rank deficient[2], the solution to (82) is not unique. We apply the pseudoinverse of $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}$ to obtain the minimum-norm solution to (82):

$$\boldsymbol{B}_1 = \left(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\right)^{\dagger}\left(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}} - \boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\boldsymbol{\beta}^p\boldsymbol{x}^{p\mathrm{T}}\right)\boldsymbol{X}_1\left(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{X}_1\right)^{-1}. \tag{84}$$

Substituting (84) into (83) yields

$$\begin{aligned}&\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\boldsymbol{\beta}^p\boldsymbol{x}^{p\mathrm{T}}\left[\boldsymbol{I}_n - \boldsymbol{X}_1\left(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{X}_1\right)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\right]\boldsymbol{x}^p + \alpha\frac{\partial\|\boldsymbol{\beta}^p\|_1}{\partial\boldsymbol{\beta}^p}\\ =&\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\left[\boldsymbol{I}_n - \boldsymbol{X}_1\left(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{X}_1\right)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\right]\boldsymbol{x}^p.\end{aligned} \tag{85}$$

2. Simple analysis shows that the rank of $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}$ is $m-1$.

$$L_\rho(\{\boldsymbol{\beta}_i\},\boldsymbol{\delta}_0,\boldsymbol{\lambda}) = \sum_{i=1}^m\frac{1}{2\sigma_i^2}\sum_{j=1}^n\left(\tilde{y}_{ij} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i + \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\delta}_0\right)^2 + \alpha\sum_{i=1}^m\|\boldsymbol{\beta}_i\| + \boldsymbol{\lambda}^{\mathrm{T}}\left(\frac{1}{\sum_{i=1}^m\frac{1}{\sigma_i^2}}\sum_{i=1}^m\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i - \boldsymbol{\delta}_0\right) + \frac{\rho}{2}\left\|\frac{1}{\sum_{i=1}^m\frac{1}{\sigma_i^2}}\sum_{i=1}^m\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i - \boldsymbol{\delta}_0\right\|^2 \tag{73}$$

$$\begin{aligned}L_i(\{\boldsymbol{\beta}_i\},\boldsymbol{\delta}_0,\boldsymbol{\lambda}) =& \frac{1}{2\sigma_i^2}\sum_{j=1}^n\left(\tilde{y}_{ij} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i + \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\delta}_0\right)^2 + \alpha\|\boldsymbol{\beta}_i\| + \boldsymbol{\lambda}^{\mathrm{T}}\frac{1}{\sum_{\ell=1}^m\frac{1}{\sigma_\ell^2}}\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i + \frac{\rho}{2}\left(\frac{1}{\sum_{\ell=1}^m\frac{1}{\sigma_\ell^2}}\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i + \frac{1}{\sum_{\ell=1}^m\frac{1}{\sigma_\ell^2}}\sum_{\ell\neq i}\frac{1}{\sigma_\ell^2}\boldsymbol{\beta}_\ell - \boldsymbol{\delta}_0\right)^2\\ =&\frac{1}{2}\boldsymbol{\beta}_i^{\mathrm{T}}\left(\frac{1}{\sigma_i^2}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \frac{\rho}{\sigma_i^4}\frac{1}{\left(\sum_{\ell=1}^m\frac{1}{\sigma_\ell^2}\right)^2}\boldsymbol{I}_p\right)\boldsymbol{\beta}_i - \boldsymbol{\beta}_i^{\mathrm{T}}\boldsymbol{v}_i + \alpha\|\boldsymbol{\beta}_i\| + c\end{aligned} \tag{74}$$

Note that to arrive at (85), we have exploited the fact that $\left(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\right)\left(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\right)^{\dagger}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma} = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}$ which is due to that $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M} = \boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}$ according to the definition of $\boldsymbol{M}$ in (37) and the definition of the pseudoinverse of a matrix.

Since the coefficient matrix of $\boldsymbol{\beta}^p$, i.e., $\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{M}\cdot\left(\boldsymbol{x}^{p\mathrm{T}}\left[\boldsymbol{I}_n - \boldsymbol{X}_1\left(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{X}_1\right)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\right]\boldsymbol{x}^p\right)$ is positive semidefinite, (85) implies that when

$$\begin{aligned}\alpha &\geq \left\|\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\left[\boldsymbol{I}_n - \boldsymbol{X}_1\left(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{X}_1\right)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\right]\boldsymbol{x}^p\right\|_\infty \\ &= \left\|\boldsymbol{\Sigma}\widetilde{\boldsymbol{Y}}\left[\boldsymbol{I}_n - \boldsymbol{X}_1\left(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{X}_1\right)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\right]\boldsymbol{x}^p\right\|_\infty \\ &= \max_i\left|\frac{1}{\sigma_i^2}\tilde{\boldsymbol{y}}_i{}^{\mathrm{T}}\left[\boldsymbol{I}_n - \boldsymbol{X}_1\left(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{X}_1\right)^{-1}\boldsymbol{X}_1^{\mathrm{T}}\right]\boldsymbol{x}^p\right|,\end{aligned} \tag{86}$$

where the next to last equality is due to (40), we obtain zero solution.

4.3.2 Type II Penalty

The derivative of $p(\boldsymbol{B}) = \alpha\sum_{i=1}^m \|\boldsymbol{\beta}_i\|$ with respect to $\boldsymbol{B}$ is

$$\frac{\partial p(\boldsymbol{B})}{\partial \boldsymbol{B}} = \alpha\begin{pmatrix}\frac{\partial\|\boldsymbol{\beta}_1\|}{\partial\boldsymbol{\beta}_1^{\mathrm{T}}}\\ \frac{\partial\|\boldsymbol{\beta}_2\|}{\partial\boldsymbol{\beta}_2^{\mathrm{T}}}\\ \vdots\\ \frac{\partial\|\boldsymbol{\beta}_m\|}{\partial\boldsymbol{\beta}_m^{\mathrm{T}}}\end{pmatrix}, \tag{87}$$

where $\frac{\partial\|\boldsymbol{\beta}_i\|}{\partial\boldsymbol{\beta}_i} = \frac{\boldsymbol{\beta}_i}{\|\boldsymbol{\beta}_i\|}$ if $\boldsymbol{\beta}_i \neq \boldsymbol{0}$ and $\left\|\frac{\partial\|\boldsymbol{\beta}_i\|}{\partial\boldsymbol{\beta}_i}\right\| \leq 1$ otherwise [27], [28].

Setting (80) equal to zero yields

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{m}^i - \boldsymbol{X}^{\mathrm{T}}\widetilde{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{m}^i + \alpha\frac{\partial\|\boldsymbol{\beta}_i\|}{\partial\boldsymbol{\beta}_i} = \boldsymbol{0}_{p\times 1}, \tag{88}$$

for $i = 1, 2, \ldots, m$, where $\boldsymbol{m}^i$ is the i-th column of $\boldsymbol{M}$ in (37). The minimizer to $f(\boldsymbol{B})$ is a zero matrix when

$$\alpha \geq \max_i\left\|\boldsymbol{X}^{\mathrm{T}}\widetilde{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{m}^i\right\|. \tag{89}$$

Note that

$$\widetilde{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{m}^i = \widetilde{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Sigma}\left(\boldsymbol{e}_i - \frac{1}{\sum_{\ell=1}^m\frac{1}{\sigma_\ell^2}}\frac{1}{\sigma_i^2}\boldsymbol{1}_m\right) = \widetilde{\boldsymbol{Y}}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{e}_i = \frac{1}{\sigma_i^2}\tilde{\boldsymbol{y}}_i, \tag{90}$$

where the next to last equality is due to (25).

Substituting (90) into (89) yields

$$\alpha_{\max} = \max_i\left\|\frac{1}{\sigma_i^2}\boldsymbol{X}^{\mathrm{T}}\tilde{\boldsymbol{y}}_i\right\|. \tag{91}$$

4.4 Maximum likelihood estimation of $\{\sigma_i^2\}_{i=1}^m$

To solve for $\{\sigma_i^2\}_{i=1}^m$, consider the negative log-likelihood function with $\{\sigma_i^2\}_{i=1}^m$ being unknown parameters as well:

$$\ell\left(\boldsymbol{\beta}_0, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{d}, \{\sigma_i^2\}_{i=1}^m\right) = \sum_{i=1}^m\left[\frac{n}{2}\log(2\pi\sigma_i^2) + \frac{1}{2\sigma_i^2}\sum_{j=1}^n\left(y_{ij} - \beta_{i0} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i - d_j\right)^2\right] \tag{92}$$

Taking the partial derivatives of $\ell(\cdot)$ with respect to d_j and β_{i0} and setting the result to zero, we arrive at

$$d_j = d_j - d_1 = \left(\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)}\right) - \left(\boldsymbol{x}_j - \boldsymbol{x}_1\right)^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}, \tag{93}$$

$$\beta_{i0} = \bar{y}_{i\cdot} - \frac{1}{n}\sum_{j=1}^n\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i - \frac{1}{n}\sum_{j=1}^n d_j = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - \boldsymbol{x}_1^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}, \tag{94}$$

where to derive the second equality we have exploited assumption (58).

The sum of (93) and (94) gives

$$\beta_{i0} + d_j = \bar{y}_{i\cdot} + \bar{y}_{\cdot j}^{(w)} - \bar{y}^{(w)} - \boldsymbol{x}_j^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}. \tag{95}$$

Taking the partial derivative of $\ell(\cdot)$ with respect to $\boldsymbol{\beta}_i$ and setting the result to zero, we have

$$\boldsymbol{\beta}_i = \sum_{j=1}^n\boldsymbol{x}_j y_{ij} - \sum_{j=1}^n\boldsymbol{x}_j\left(\beta_{i0} + d_j\right). \tag{96}$$

Substituting (95) into (96) yields

$$\boldsymbol{\beta}_i = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\left[\sum_{j=1}^n\boldsymbol{x}_j y_{ij} - \frac{1}{\sum_{i=1}^m\frac{1}{\sigma_i^2}}\sum_{i=1}^m\frac{1}{\sigma_i^2}\sum_{j=1}^n\boldsymbol{x}_j y_{ij}\right] + \bar{\boldsymbol{\beta}}^{(w)}, \tag{97}$$

where $\bar{\boldsymbol{\beta}}^{(w)}$ is defined in (60).

Taking the partial derivative of σ_i^2 and setting the result to zero gives

$$\sigma_i^2 = \frac{1}{n}\sum_{j=1}^n\left(y_{ij} - \beta_{i0} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i - d_j\right)^2. \tag{98}$$

Substituting (95) into (98) yields

$$\sigma_i^2 = \frac{1}{n}\sum_{j=1}^n\left(y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)} - \boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{\beta}_i + \boldsymbol{x}_j^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}\right)^2, \tag{99}$$

where $\bar{y}_{i\cdot}$, $\bar{y}_{\cdot j}^{(w)}$ and $\bar{y}^{(w)}$ are defined in (15), (11) and (18), respectively.

Given initial estimates for $\bar{\boldsymbol{\beta}}^{(w)}$ and $\{\sigma_i^2\}_{i=1}^m$, estimates for $\boldsymbol{\beta}_i$ and σ_i^2 can then be iteratively updated using equations (97), (99), and (60) until convergence.

After estimating σ_i^2's, they can then be shrinked (squeezed) toward the common noise variance to obtain robust estimates for σ_i^2, as done in Section 3.4.

Given initial estimates for $\bar{\boldsymbol{\beta}}^{(w)}$ and $\{\sigma_i^2\}_{i=1}^m$, estimates for $\boldsymbol{\beta}_i$ and σ_i^2 can then be iteratively updated using equations (97), (99), and (60) until convergence, as shown in Algorithm 5.

5 Experiments

We evaluate the performance of the proposed algorithm (referred to as ELMSeq, short for extended linear model for RNA-seq data analysis). To save space, we only verify the proposed algorithm for the simple regression model (2). We use the 5th percentile to set the tuning parameter α (see Section 3.3).

We compare our method with the state-of-the-art methods for detecting differential gene expression from RNA-seq data: edgeR-robust [20], [29], DESeq2 [18], and limma-voom [16], [17].

Table 1: Synthetic data generation process and parameters

$\ell_i \sim 2^{\text{unif}(5,10)}$	gene length of gene i
$\beta_{i0} \sim \mathcal{N}(0,1)$	other log scaling factors of gene i
$\beta_i = 0$	log-fold change for non-DE genes
$\beta_i \sim \mathcal{N}(2,1)$	log-fold change for up-regulated DE genes
$\beta_i \sim \mathcal{N}(-2,1)$	log-fold change for down-regulated DE genes
$x_j \sim \mathcal{N}(0,1)$	condition data of sample j
$N_j \sim \text{unif}(2,3) \times 10^6$	library size of sample j
$d_j \sim \mathcal{N}(0,1)$	other log scaling factors of sample j
$\mu_{ij} = N_j \frac{\ell_i}{\sum_{i=1}^m \ell_i} e^{\beta_{i0}+\beta_i x_j + d_j}$	expected RNA-seq read counts of gene i from sample j
$c_{ij} = \lceil e^{\log \mathcal{N}(\mu_{ij}, 0.1)} \rceil$	read counts
$y_{ij} = \log c_{ij}$	log-transformed gene expression

5.1 Simulations on Synthetic Data

We simulate RNA-seq data with a total of $m = 1000$ genes and $n = 15$ samples. The data generation is described in Table 1.

We first examine whether the proposed algorithm can accurately estimate the log-fold changes (or slopes) β_i's. For ease of illustration, we set the true slopes for DE ones as $\beta_i = \pm 2$ instead of $\beta_i \sim \mathcal{N}(\pm 2, 1)$.

We start with 300 DE genes and 700 non-DE genes. Among DE genes 50% are up-regulated while the remaining 50% are down-regulated. The fitted $\{\beta_i\}_{i=1}^m$ using ELMSeq are plotted in Figure 1(a). We see that the estimated slopes are centered around the true ones: the estimated β_i's of the DE genes are centered around ± 2, while those of the non-DE genes are close to zero. In Figure 1(b) and Figure 1(c), we increase the percent of up-regulated DE genes to 70% and 90%, rspectively. Our method still accurately retrieves all non-zero β_i's while shrinking all other β_i's to zero.

In Figure 1(d-f), we increase the number of DE genes to 500, among which 50%, 70% or 90% are up-regulated while others are down-regulated. Our method still achieves accurate estimates. In Figure 1(g-h), we further increase the number of DE genes to 700 among which 50% or 70% are up-regulated, for which our method still achieves accurate estimates when. Only when we simulate with 700 DE genes among which 90% are up-regulated, our method fails to distinguish between DE and non-DE genes since the estimated regression coefficients of the latter are not zero either [Figure 1(i)]. A theoretical explanation of Figure 1(i) has been provided in the supplementary material.

Using a different gene expression measure such as CPM, RPKM or TPM values computed with formulas in (1) yields essentially the same result.

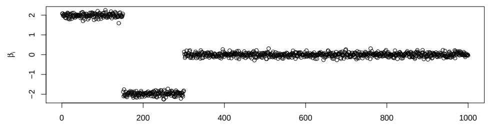
Using the algorithm in Algorithm (1), we estimate the regression coefficient $\hat{\beta}_i$ for each gene i. We decide there is a linear relationship between the predictor variable x_j and the expression data y_{ij} if $\hat{\beta}_i \neq 0$. The larger $\left|\hat{\beta}_i\right|$ is, the stronger the relationship. We then sort the genes in descending order of their $\left|\hat{\beta}_i\right|$ and vary the threshold to construct the receiver operating characteristic (ROC) curve and to calculate the area under the ROC curve (AUC).

The AUCs for DE gene detection using all four methods are summarized in Table 2. We see that the ELMSeq performs better than or comparbly to other three methods, regardless of how many genes are differentially expressed and whether they are expressed in a symmetric manner or not. In challenging cases where a large proportion of genes are differentially expressed in an asymmetric manner (e.g., 50% DE genes among which 90% are up-regulated or 70% DE genes among which 70% are up-regulated), the performance gain of the ELMSeq over completing methods is more significant.

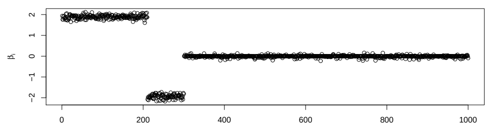
In Table 3, we decrease the log-fold change of the DE genes as $\beta_i \sim \mathcal{N}(\pm 0.2, 0.1)$ while keeping all other data generation parameters (including the noise level) the same as those in Table 2. We see that all methods suffer a degradation in AUC performance; but again, the ELMSeq consistently perform better than or comparably to all other methods.

Note that when more samples are available, the performance gain of the ELMSeq over completing methods becomes even more significant. The results for various sample sizes $n = 5, 8, 25, 50, 100$ are provided in the supplementary materials (Figs. S1–S5 for genes with high expression profiles $\beta_i \sim \mathcal{N}(\pm 2, 1)$ and Figs. S6–S10 for genes with low expression profiles $\beta_i \sim \mathcal{N}(\pm 0.2, 0.1)$).

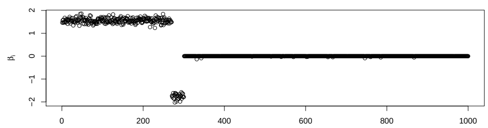
We also performed simulations with the multiple linear regression model in Section 4, and the preliminary results are similar to that obtained for the simple regression model. Note that unlike the simple regression model and type I penalized multiple linear regression model, the type II penalized multiple linear regression model does not allow to define up- and down-regulated genes as multiple regression coefficients are tested simultaneously.



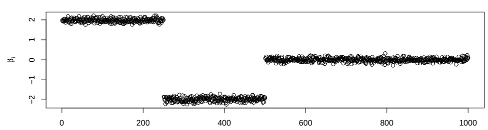
(a) 300 DE genes, among which 50% are up-regulated



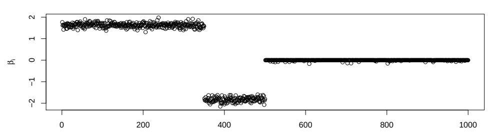
(b) 300 DE genes, among which 70% are up-regulated



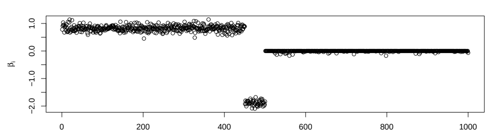
(c) 300 DE genes, among which 90% are up-regulated



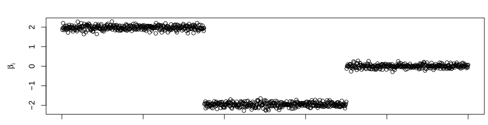
(d) 500 DE genes, among which 50% are up-regulated



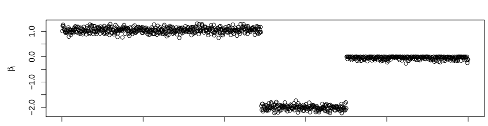
(e) 500 DE genes, among which 70% are up-regulated



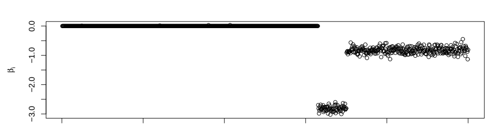
(f) 500 DE genes, among which 90% are up-regulated



(g) 700 DE genes, among which 50% are up-regulated



(h) 700 DE genes, among which 70% are up-regulated



(i) 700 DE genes, among which 90% are up-regulated

Figure 1: Estimated β_i in the simple linear regression model from simulated RNASeq data, where the number of genes is $m = 1000$ and number of samples is $n = 15$. The number of DE genes varies from 300 to 700, and the percentage of up-regulated DE genes varies from 50% to 90%. Along the horizontal axis, from left to right: up-regulated genes ($\beta_i = 2$), down-regulated genes ($\beta_i = -2$) and non-DE genes ($\beta_i = 0$).

Table 2: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. Number of samples: $n = 15$, log-fold change for DE genes: $\beta_i \sim \mathcal{N}(\pm 2, 1)$, and noise level: $\sigma_i = 0.1$. The table shows the percent of DE genes (DE %), percent of up-regulated genes among the DE genes (Up %), as well as the mean AUCs for all four methods measured using 10 simulated replicates. The standard errors of the mean AUCs are given in parentheses.

DE (%)	Up (%)	edgeR	DESeq2	voom	ELMSeq
10	50	0.9903	0.6068	0.991	**0.9914**
		(0.0016)	(0.0807)	(0.0018)	**(0.0017)**
10	70	0.9935	0.4527	0.9941	**0.9943**
		(0.0021)	(0.0638)	(0.0021)	**(0.0021)**
10	90	0.9869	0.6878	0.9875	**0.9897**
		(0.0028)	(0.0637)	(0.0024)	**(0.0022)**
30	50	0.9898	0.5508	**0.99**	**0.99**
		(0.001)	(0.0883)	**(0.001)**	**(0.001)**
30	70	0.9891	0.7946	0.9897	**0.991**
		(0.0014)	(0.064)	(0.0014)	**(0.0011)**
30	90	0.9788	0.6114	**0.9796**	0.9795
		(0.0023)	(0.0805)	**(0.0022)**	(0.0014)
50	50	0.9917	0.429	0.9916	**0.9917**
		(8e-04)	(0.0797)	(8e-04)	**(8e-04)**
50	70	0.9748	0.4923	0.9754	**0.9826**
		(0.0026)	(0.081)	(0.0026)	**(0.0015)**
50	90	0.8717	0.4697	0.8801	**0.9662**
		(0.0133)	(0.0667)	(0.0119)	**(0.002)**
70	50	0.9907	0.5572	0.9915	**0.9923**
		(9e-04)	(0.1027)	(8e-04)	**(7e-04)**
70	70	0.8564	0.5307	0.8696	**0.9591**
		(0.018)	(0.0588)	(0.0148)	**(0.0034)**
70	90	0.3375	**0.4808**	0.3204	0.4718
		(0.0108)	**(0.0192)**	(0.0154)	(0.0124)

Table 3: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table 2 except that the log-fold changes for DE genes decrease to: $\beta_i \sim \mathcal{N}(\pm 0.2, 0.1)$.

DE (%)	Up (%)	edge	DESeq2	voom	ELMSeq
10	50	0.8055	0.5241	0.8224	**0.8232**
		(0.0089)	(0.0142)	(0.0095)	**(0.0095)**
10	70	0.8086	0.4846	0.8212	**0.8234**
		(0.009)	(0.0126)	(0.0095)	**(0.0101)**
10	90	0.7867	0.5078	0.7955	**0.8024**
		(0.0084)	(0.0084)	(0.0104)	**(0.0106)**
30	50	0.8087	0.497	**0.8158**	0.8157
		(0.005)	(0.0119)	**(0.0054)**	(0.0054)
30	70	0.7848	0.5471	0.7949	**0.8013**
		(0.0052)	(0.0211)	(0.0052)	**(0.0055)**
30	90	0.7398	0.5329	0.7505	**0.773**
		(0.0059)	(0.0181)	(0.0059)	**(0.0054)**
50	50	0.8143	0.4931	0.8265	**0.8268**
		(0.0061)	(0.0137)	(0.0049)	**(0.0051)**
50	70	0.7611	0.5061	0.7704	**0.7752**
		(0.0054)	(0.0155)	(0.0054)	**(0.0056)**
50	90	0.6451	0.5017	0.6503	**0.6793**
		(0.006)	(0.0102)	(0.0059)	**(0.0025)**
70	50	0.8149	0.5231	0.8261	**0.8267**
		(0.0022)	(0.0273)	(0.003)	**(0.0028)**
70	70	0.7271	0.5093	0.7354	**0.7388**
		(0.0074)	(0.01)	(0.0086)	**(0.0083)**
70	90	0.5449	0.5158	**0.5505**	0.5434
		(0.0066)	(0.0089)	**(0.0081)**	(0.0069)

5.2 An application to a real RNA-Seq dataset

We further evaluate our algorithm on a prostate adenocarcinoma (PRAD) RNA-Sequencing dataset published as part of The Cancer Genome Atlas (TCGA) project [30]. The RNA-Seq datasets of 20531 genes from 187 samples were downloaded from the TCGA data portal (https://tcga-data.nci.nih.gov/tcga). We desire to identify genes that are associated with pre-operative prostate-specific antigen (PSA), an important risk factor for prostate cancer. The gene expression data were preprocessed by the TCGA consortium. Tissue samples from 333 PRAD patients were sequenced using the Il-lumina sequencing instruments. The raw sequencing reads were processed and analyzed using the SeqWare Pipeline 0.7.0 and MapspliceRSEM workflow 0.7 developed by the University of North Carolina, and then aligned to the human reference genome using MapSplice [31]. The gene expression distributions of all samples are normalized to have the same 75th percentile expression values (1,000).

Using the algorithm in Algorithm 1, we obtain the estimated between-sample normalization factors $\hat{d}_j$'s and regression coefficient $\hat{\beta}_i$ for each gene i. We then substitute $\hat{d}_j$'s into model (2), and for each gene i compute the p-value by testing the null hypothesis that the slope of the regression line is equal to zero, i.e., $\beta_i = 0$. We determine a gene is differentially expressed if the p value associated with its linear regression model is less than $0.05/m$. Here the threshold $0.05/m$ is determined using the Bonferroni correction to adjust for multiple significant tests and to achieve a desired family-wise error rate of 0.05. The relations between the sets of differentially expressed genes selected by edgeR, DESeq2, limma-voom and ELMSeq are depicted in Fig. 2.

Nine genes are uniquely detected by ELMSeq: *RIC3*, *ALDH1A2*, *BCL11A*, *CDH3*, *DIRAS3*, *EPHA5*, *CEACAM1*, *PRSS16*, and *AJAP1*. For most of these genes, evidence has also been reported in the literature on their association with prostate cancer. For example, the genes *ALDH1a2* [32] and *CEACAM1* [33] are reported to be tumor suppressors in prostate cancer: underexpression of these genes promote prostate cancer cell proliferation.

Twelve genes are detected by all four methods: *KANK4*, *RHOU*, *TPT1*, *SH2D3A*, *EEF1A1P9*, *ZCWPW1*, *ZNF454*, *RACGAP1*, *PTPLA*, *POC1A*, *AURKA* and *TIMM17A*. The common genes detected by three methods are: six genes *CDK1*, *FAM111B*, *MLF1IP*, *PRC1*, *DTL*, *RAD54B* by edgeR, DESeq2, and limma-voom, three genes *SH3RF2*, *ATCAY* and *PCP4* by edgeR, DESeq2 and ELMSeq, three genes *FERMT1*, *FOXA3* and *LRAT* by edgeR, limma-voom and ELMSeq, and one gene *IPO9* by DESeq2, limma-voom and ELMSeq. For most of these genes, evidence has also been reported in the literature on their association with prostate cancer. For example, the silencing of gene RHOU decreases the invasion, proliferation and motility of prostate cancer cells [34].

6 Discussion

A unified statistical model is proposed for joint between-sample normalization and DE detection of RNA-seq data. The sample-specific normalization factors are modeled as

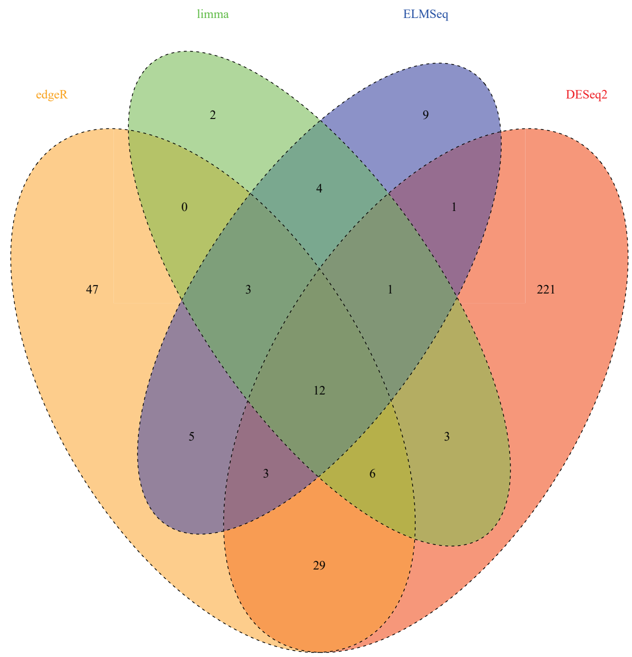


Figure 2: Venn diagram showing the relation between the set of differentially expressed genes detected by edgeR, DESeq2, limma-voom and ELMSeq.

unknown parameters and jointly estimated together with DE detection. As a result, the model is robust against normalization errors and is independent of the units (i.e., counts, CPM/RPM, RPKM/FPKM or TPM) in which gene expression levels are summarized.

For the model with a single treatment condition, we introduce the L1 penalty to the linear regression model. The L1 penalty favors sparse solutions (forces some coefficients to be exactly zero). This is desirable since many genes are not differentially expressed. From a Bayesian point of view, the lasso penalty corresponds to a Laplace (double exponential centred at zero) prior over the regression coefficients. By contrast, existing methods do not exploit the sparsity information. We also extend the simple linear regression model to multiple linear regression model to accommodate multiple treatment conditions. Two types of penalty functions are introduced. In the first one only one covariate is of interest while all other covariates are treated as confounding factors. We are interested in testing whether that specific covariate is associated with differential expression. In the second case all covariates are of interest (there are no confounding covariates) and we are interested in testing whether any covariate affects the differential expression of a gene.

Simulation studies show that the proposed methods always perform better than or comparably to existing methods in terms of AUC. The performance gain increases with a larger sample size or higher signal to noise ratio, and is more significant when a large proportion of genes are differentially expressed in an asymmetric manner.

The R codes of the algorithms described in the paper are available for download at http://www-personal.umich.edu/~jianghui/lr-ADMM/.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

[1] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-Seq." *Nat Methods*, vol. 5, no. 7, pp. 621–628, Jul 2008.

[2] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nat Biotechnol*, vol. 28, no. 5, pp. 511–515, May 2010.

[3] H. Jiang and W. H. Wong, "Statistical inferences for isoform expression in RNA-Seq," *Bioinformatics*, vol. 25, no. 8, pp. 1026–1032, Apr 2009.

[4] J. Salzman, H. Jiang, and W. H. Wong, "Statistical modeling of RNA-Seq data," *Statistical Science*, vol. 26, no. 1, pp. 62–83, 2011.

[5] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics." *Nat Rev Genet*, vol. 10, no. 1, pp. 57–63, Jan 2009.

[6] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, G. Guernec, B. Jagla, L. Jouneau, D. Laloe, C. Le Gall, B. Schaeffer, L.ffer, S. Le Crom, M. Guedj, F. Jaffrezic, and F. S. C. , "A comprehensive evaluation of normalization methods for illumina high-throughput RNA sequencing data analysis," *Brief Bioinform*, vol. 14, no. 6, pp. 671–683, Nov 2013.

[7] F. Rapaport, R. Khanin, Y. Liang, M. Pirun, A. Krek, P. Zumbo, C. E. Mason, N. D. Socci, and D. Betel, "Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data," *Genome Biology*, vol. 14, no. 9, p. R95, Sep 2013.

[8] M. D. Robinson and A. Oshlack, "A scaling normalization method for differential expression analysis of RNA-seq data," *Genome Biol*, vol. 11, no. 3, p. R25, 2010.

[9] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias." *Bioinformatics*, vol. 19, no. 2, pp. 185–193, Jan 2003.

[10] G. K. Smyth, "Limma: linear models for microarray data," in *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer, 2005, pp. 397–420.

[11] J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit, "Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments." *BMC Bioinformatics*, vol. 11, p. 94, 2010.

[12] S. Anders and W. Huber, "Differential expression analysis for sequence count data," *Genome Biol*, vol. 11, no. 10, p. R106, 2010.

[13] B. Li, V. Ruotti, R. M. Stewart, J. A. Thomson, and C. N. Dewey, "RNA-Seq gene expression estimation with read mapping uncertainty." *Bioinformatics*, vol. 26, no. 4, pp. 493–500, Feb 2010.

[14] A. Oshlack, M. J. Wakefield *et al.*, "Transcript length bias in RNA-seq data confounds systems biology," *Biol Direct*, vol. 4, no. 1, p. 14, 2009.

[15] J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad, "RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays," *Genome research*, vol. 18, no. 9, pp. 1509–1517, 2008.

[16] C. W. Law, Y. Chen, W. Shi, and G. K. Smyth, "Voom: precision weights unlock linear model analysis tools for RNA-seq read counts," *Genome Biol*, vol. 15, no. 2, p. R29, 2014.

[17] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth, "limma powers differential expression analyses for RNA-sequencing and microarray studies," *Nucleic Acids Research*, vol. 43, no. 7, p. e47, Jan 2015.

[18] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2," *Genome Biology*, vol. 15, no. 12, p. 550, 2014.

[19] H. Jiang and J. Salzman, "A penalized likelihood approach for robust estimation of isoform expression," *Statistics and Its Interface*, vol. 8, no. 4, pp. 437–445, 2015.

[20] X. Zhou, H. Lindsay, and M. D. Robinson, "Robustly detecting differential expression in RNA sequencing data using observation weights," *Nucleic acids research*, vol. 42, no. 11, pp. e91–e91, 2014.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[22] H. Jiang and T. Zhan, "Unit-free and robust detection of differential expression from RNA-Seq data," *Statistics in Biosciences*, vol. 9, no. 1, pp. 178–199, 2017.

[23] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[24] H. Ji and W. H. Wong, "TileMap: create chromosomal map of tiling array hybridizations," *Bioinformatics*, vol. 21, no. 18, pp. 3629–3636, 2005.

[25] H. Ji and X. S. Liu, "Analyzing 'omics data using hierarchical models," *Nature biotechnology*, vol. 28, no. 4, pp. 337–340, 2010.

[26] G. Smyth, "Statistical applications in genetics and molecular biology," *Linear models and empirical Bayes methods for assessing differential expression in microarray experiments*, 2004.

[27] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[28] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[29] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: a bioconductor package for differential expression analysis of digital gene expression data." *Bioinformatics*, vol. 26, pp. 139–140, Jan 2010.

[30] C. G. A. R. Network, "The molecular taxonomy of primary prostate cancer," *Cell*, vol. 163, pp. 1011–1025, Nov. 2015.

[31] K. Wang, D. Singh, Z. Zeng, S. J. Coleman, Y. Huang, G. L. Savich, X. He, P. Mieczkowski, S. A. Grimm, C. M. Perou, J. N. MacLeod, D. Y. Chiang, J. F. Prins, and J. Liu, "Mapsplice: accurate mapping of RNA-seq reads for splice junction discovery," *Nucleic acids research*, vol. 38, p. e178, Oct. 2010.

[32] H. Kim, J. Lapointe, G. Kaygusuz, D. E. Ong, C. Li, M. van de Rijn, J. D. Brooks, and J. R. Pollack, "The retinoic acid synthesis gene ALDH1a2 is a candidate tumor suppressor in prostate cancer," *Cancer research*, vol. 65, no. 18, pp. 8118–8124, 2005.

[33] C. Busch, T. A. Hanssen, C. Wagener, and B. Öbrink, "Down-regulation of CEACAM1 in human prostate cancer: correlation with loss of cell polarity, increased proliferation rate, and gleason grade 3 to 4 transition," *Human pathology*, vol. 33, no. 3, pp. 290–298, 2002.

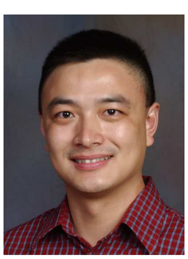
[34] S. Alinezhad, R.-M. Väänänen, J. Mattsson, Y. Li, T. Tallgrén, N. T. Ochoa, A. Bjartell, M. Åkerfelt, P. Taimen, P. J. Boström *et al.*, "Validation of novel biomarkers for prostate cancer progression by the combination of bioinformatics, clinical and functional studies," *PloS one*, vol. 11, no. 5, p. e0155901, 2016.

Kefei Liu received B.Sc. in mathematics and Ph.D. in Electronic Engineering from Wuhan University in 2006 and City University of Hong Kong in 2013, respectively. He is currently a Postdoctoral Research Fellow at the Center for Computational Biology and Bioinformatics, Indiana University School of Medicine. Before joining IU, he worked as a Postdoctoral Research Associate at The Biodesign Institute of Arizona State University and the Department of Computational Medicine and Bioinformatics of University of Michigan. His current research interests include machine learning, optimization, tensor decompositions and their applications in biomedical data analysis.

Jieping Ye received the Ph.D. degree in computer science from the University of Minnesota, Twin Cities, MN, USA, in 2005.

He is an Associate Professor of Department of Computational Medicine and Bioinformatics and Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, MI, USA. His research interests include machine learning, data mining, and biomedical informatics. Dr. Ye has served as Senior Program Committee/Area Chair/Program Committee Vice Chair of many conferences including NIPS, ICML, KDD, IJCAI, ICDM, SDM, ACML, and PAKDD. He serves as a PC Co-Chair of SDM 2015. He serves as an Associate Editor for IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING and IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and serves as an Action Editor for Data Mining and Knowledge Discovery. He won the NSF CAREER Award in 2010. His papers have been selected for the outstanding student paper at ICML in 2004, the KDD best research paper honorable mention in 2010, the KDD best research paper nomination in 2011 and 2012, the SDM best research paper runner up in 2013, the KDD best research paper runner up in 2013, and the KDD best student paper award in 2014.

Yang Yang is a Ph.D. candidate at Beihang University. Now he is a visiting student under the supervision of distinguished professor Philip S. Yu at the University of Illinois at Chicago. He got his bachelor's and masters degree from Xidian University. His research interests are social network analysis, machine learning, and complex networks.



Li Shen holds a B.S. degree from Xian Jiao Tong University, an M.S. degree from Shanghai Jiao Tong University, and a Ph.D. degree from Dartmouth College, all in Computer Science. He is an Associate Professor of Radiology and Imaging Sciences at Indiana University School of Medicine. His research interests include medical image computing, bioinformatics, data mining, network science, systems biology, brain imaging genomics, and brain connectomics.

Hui Jiang is an Assistant Professor in the Department of Biostatistics at University of Michigan. He received his Ph.D. in Computational and Mathematical Engineering from Stanford University in 2009. He received his B.S. and M.S. in Computer Science from Peking University. Before joining the University of Michigan in 2011, he was a postdoctoral scholar in the Department of Statistics and Genome Technology Center at Stanford University. He is interested in developing statistical and computational methods for the analysis of large-scale biological data generated using modern high-throughput technologies.

A Unified Model for Joint Normalization and Differential Gene Expression Detection in RNA-Seq data: Supplementary Material

Algorithm 3 ADMM algorithm for fitting the multiple linear regression model with type I penalty

Input: Log-transformed gene expression measurements: $\{\{y_{ij}\}_{i=1}^{m}\}_{j=1}^{n}$, predictor variables: $\{\boldsymbol{x}_j\}_{j=1}^{n}$ and estimated noise variance: $\{\sigma_i^2\}_{i=1}^{m}$.

Transform data. Normalize each column of $\boldsymbol{X}$ to zero mean and unit norm: ▷ The scaling step is performed only for consistency with the developed algorithm in Algorithm 1.

$$\tilde{x}_{jk} = \frac{x_{jk} - \bar{x}_{\cdot k}}{\sqrt{\sum_{j=1}^{n}\left(x_{jk} - \bar{x}_{\cdot k}\right)^2}}, \text{ with } \bar{x}_{\cdot k} \coloneqq \frac{1}{n}\sum_{j=1}^{n} x_{jk}, \quad k = 1, 2, \ldots, p.$$

Center y_{ij} to zero mean over row index i and column index j: $\tilde{y}_{ij} = y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)}$, where $\bar{y}_{i\cdot}$, $\bar{y}_{\cdot j}^{(w)}$ and $\bar{y}^{(w)}$ are defined in (15), (11) and (18), respectively.

Set the penalty parameter to $\rho = 1$ [21]; select the tuning parameter α according to Section 3.3.

Initialization: $k = 0$; randomly initialize $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^0$, $i = 1, 2, \ldots, m$, $\boldsymbol{\delta}_0 = \boldsymbol{\delta}_0^0$, and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^0$.

repeat

  for $i = 1, 2, \ldots, m$ **do**

    Let $\boldsymbol{Q}$ and $\boldsymbol{v}_i$ be partitioned as

$$\boldsymbol{Q} \coloneqq \frac{1}{\sigma_i^2}\tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}} + \frac{\rho}{\sigma_i^4}\frac{1}{\left(\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}\right)^2}\boldsymbol{I}_p = \begin{pmatrix} \boldsymbol{Q}_{11} & \boldsymbol{q} \\ \boldsymbol{q}^{\mathrm{T}} & q_{pp} \end{pmatrix},$$

$$\boldsymbol{v}_i \coloneqq \frac{1}{\sigma_i^2}\left(\sum_{j=1}^{n}\tilde{\boldsymbol{x}}_j\tilde{y}_{ij} + \tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}}\boldsymbol{\delta}_0^k\right) - \frac{\rho}{\sigma_i^2}\frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\left(\frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\sum_{\ell\neq i}\frac{1}{\sigma_\ell^2}\boldsymbol{\beta}_\ell^k - \boldsymbol{\delta}_0^k + \frac{\boldsymbol{\lambda}^k}{\rho}\right) = \begin{pmatrix} \boldsymbol{v}_i^- \\ v_{ip} \end{pmatrix}.$$

    Update $\boldsymbol{\beta}_i$:

$$\boldsymbol{\beta}_i^{k+1} = \begin{pmatrix} \boldsymbol{\beta}_i^{-k+1} \\ \beta_{ip}^{k+1} \end{pmatrix},$$

where

$$\beta_{ip}^{k+1} = \frac{1}{q_{pp} - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{Q}_{11}^{-1}\boldsymbol{q}}T_\alpha\left[v_{ip} - \boldsymbol{q}^{\mathrm{T}}\boldsymbol{Q}_{11}^{-1}\boldsymbol{v}_i^-\right], \quad \boldsymbol{\beta}_i^{-k+1} = \boldsymbol{Q}_{11}^{-1}\left(\boldsymbol{v}_i^- - \boldsymbol{q}\beta_{ip}^{k+1}\right).$$

  end for

  Update $\boldsymbol{\delta}_0$:

$$\boldsymbol{\delta}_0^{k+1} = \left(\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\cdot\tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}} + \rho\boldsymbol{I}_p\right)^{-1}\boldsymbol{\lambda}^k + \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i^{k+1}$$

  Update $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho\left(\frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i^{k+1} - \boldsymbol{\delta}_0^{k+1}\right)$$

  $k \leftarrow k + 1$;

until convergence or maximum number of iterations is reached.

Output: $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^k$, $i = 1, 2, \ldots, m$, $\bar{\boldsymbol{\beta}}^{(w)} = \boldsymbol{\delta}_0^k$, and

$$\beta_{i0} = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - \tilde{\boldsymbol{x}}_1^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}, \; i = 1, 2, \ldots, m$$

$$d_1 = 0, \quad d_j = \left(\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)}\right) - \left(\tilde{\boldsymbol{x}}_j - \tilde{\boldsymbol{x}}_1\right)^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}, \; j = 2, \ldots, n$$

Recover the original parameter space:

$$\beta_{i0}' = \beta_{i0} - \sum_{k=1}^{p}\frac{\beta_{ik}\bar{x}_{\cdot k}}{\sqrt{\sum_{j=1}^{n}\left(x_{jk} - \bar{x}_{\cdot k}\right)^2}}, \quad \beta_{ik}' = \frac{\beta_{ik}}{\sqrt{\sum_{j=1}^{n}\left(x_{jk} - \bar{x}_{\cdot k}\right)^2}}, \quad i = 1, 2, \ldots, m.$$

Algorithm 4 ADMM algorithm for fitting the multiple linear regression model with type II penalty

Input: Log-transformed gene expression measurements: $\{\{y_{ij}\}_{i=1}^{m}\}_{j=1}^{n}$, predictor variables: $\{\boldsymbol{x}_j\}_{j=1}^{n}$ and estimated noise variance: $\{\sigma_i^2\}_{i=1}^{m}$.

Transform data. Normalize each column of $\boldsymbol{X}$ to zero mean and unit norm:

$$\tilde{x}_{jk} = \frac{x_{jk} - \bar{x}_{\cdot k}}{\sqrt{\sum_{j=1}^{n}\left(x_{jk} - \bar{x}_{\cdot k}\right)^2}}, \text{ with } \bar{x}_{\cdot k} \coloneqq \frac{1}{n}\sum_{j=1}^{n} x_{jk}, \quad k = 1, 2, \ldots, p.$$

Center y_{ij} to zero mean over row index i and column index j:

$$\tilde{y}_{ij} = y_{ij} - \bar{y}_{i\cdot} - \bar{y}_{\cdot j}^{(w)} + \bar{y}^{(w)},$$

where $\bar{y}_{i\cdot}$, $\bar{y}_{\cdot j}^{(w)}$ and $\bar{y}^{(w)}$ are defined in (15), (11) and (18), respectively.

Compute the eigendecomposition of $\tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}}$:

$$\tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathrm{T}}.$$

Set the penalty parameter to $\rho = 1$ [21]; select the tuning parameter α according to Section 3.3.

Initialization: $k = 0$; randomly initialize $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^0$, $i = 1, 2, \ldots, m$, $\boldsymbol{\delta}_0 = \boldsymbol{\delta}_0^0$, and $\boldsymbol{\lambda} = \boldsymbol{\lambda}^0$.

repeat

  for $i = 1, 2, \ldots, m$ **do**

    Denote

$$\boldsymbol{v}_i = \frac{1}{\sigma_i^2}\left(\sum_{j=1}^{n}\tilde{\boldsymbol{x}}_j\tilde{y}_{ij} + \tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}}\boldsymbol{\delta}_0^k\right) - \frac{\rho}{\sigma_i^2}\frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\left(\frac{1}{\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}}\sum_{\ell\neq i}\frac{1}{\sigma_\ell^2}\boldsymbol{\beta}_\ell^k - \boldsymbol{\delta}_0^k + \frac{\boldsymbol{\lambda}^k}{\rho}\right).$$

    Update $\boldsymbol{\beta}_i$:

    if $\|\boldsymbol{v}_i\| \leq \alpha$ **then**

      $\boldsymbol{\beta}_i^{k+1} = \boldsymbol{0}$;

    else

$$\boldsymbol{\beta}_i^{k+1} = \arg\min_{\boldsymbol{\beta}_i}\frac{1}{2}\left\|\boldsymbol{Z}_i\boldsymbol{\beta}_i - \boldsymbol{b}_i\right\|^2 + \alpha\left\|\boldsymbol{\beta}_i\right\|,$$

where

$$\boldsymbol{Z}_i = \left[\frac{1}{\sigma_i^2}\boldsymbol{D} + \frac{\rho}{\sigma_i^4}\frac{1}{\left(\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}\right)^2}\boldsymbol{I}_p\right]^{1/2}\boldsymbol{U}^{\mathrm{T}}, \quad \boldsymbol{b}_i = \left[\frac{1}{\sigma_i^2}\boldsymbol{D} + \frac{\rho}{\sigma_i^4}\frac{1}{\left(\sum_{\ell=1}^{m}\frac{1}{\sigma_\ell^2}\right)^2}\boldsymbol{I}_p\right]^{-1/2}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{v}_i.$$

The above problem can be solved via the coordinate descent algorithm described in Section 4.2.

    end if

  end for

  Update $\boldsymbol{\delta}_0$:

$$\boldsymbol{\delta}_0^{k+1} = \left(\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\cdot\tilde{\boldsymbol{X}}^{\mathrm{T}}\tilde{\boldsymbol{X}} + \rho\boldsymbol{I}_p\right)^{-1}\boldsymbol{\lambda}^k + \frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i^{k+1}$$

  Update $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho\left(\frac{1}{\sum_{i=1}^{m}\frac{1}{\sigma_i^2}}\sum_{i=1}^{m}\frac{1}{\sigma_i^2}\boldsymbol{\beta}_i^{k+1} - \boldsymbol{\delta}_0^{k+1}\right)$$

  $k \leftarrow k + 1$;

until convergence or maximum number of iterations is reached.

Output: $\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^k$, $i = 1, 2, \ldots, m$, $\bar{\boldsymbol{\beta}}^{(w)} = \boldsymbol{\delta}_0^k$, and

$$\beta_{i0} = \bar{y}_{i\cdot} + \bar{y}_{\cdot 1}^{(w)} - \bar{y}^{(w)} - \tilde{\boldsymbol{x}}_1^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}, \ i = 1, 2, \ldots, m$$

$$d_1 = 0, \quad d_j = \left(\bar{y}_{\cdot j}^{(w)} - \bar{y}_{\cdot 1}^{(w)}\right) - \left(\tilde{\boldsymbol{x}}_j - \tilde{\boldsymbol{x}}_1\right)^{\mathrm{T}}\bar{\boldsymbol{\beta}}^{(w)}, \ j = 2, \ldots, n$$

Recover the original parameter space:

$$\beta_{i0}' = \beta_{i0} - \sum_{k=1}^{p}\frac{\beta_{ik}\bar{x}_{\cdot k}}{\sqrt{\sum_{j=1}^{n}\left(x_{jk} - \bar{x}_{\cdot k}\right)^2}}, \quad \beta_{ik}' = \frac{\beta_{ik}}{\sqrt{\sum_{j=1}^{n}\left(x_{jk} - \bar{x}_{\cdot k}\right)^2}}, \quad i = 1, 2, \ldots, m.$$

Algorithm 5 Maximum likelihood estimation of $\{\sigma_i^2\}_{i=1}^m$ for multiple linear regression model

Input: Log-transformed gene expression measurements: $\{\{y_{ij}\}_{i=1}^m\}_{j=1}^n$ and predictor variables: $\{\boldsymbol{x}_j\}_{j=1}^n$.

1: Center $\{\boldsymbol{x}_j\}_{j=1}^n$ to zero mean:

$$\boldsymbol{x}_j \leftarrow \boldsymbol{x}_j - \bar{\boldsymbol{x}}, \text{ with } \bar{\boldsymbol{x}} \coloneqq \frac{1}{n}\sum_{j=1}^n \boldsymbol{x}_j.$$

2: *Initialization*: $\bar{\boldsymbol{\beta}}^{(w)} = 0$, $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_m^2 = 1$.

3: **repeat**

4: Update $\boldsymbol{\beta}_i$, $i = 1, 2, \ldots, m$, according to (97);

5: Update σ_i^2, $i = 1, 2, \ldots, m$, according to (99);

6: Update $\bar{\boldsymbol{\beta}}^{(w)}$ according to (60);

7: **until** convergence or maximum number of iterations is reached.

Output: $\hat{\sigma}_i^2 = \sigma_i^2$, $i = 1, 2, \ldots, m$.

Theoretical Explanation of Figure 1(i)

The negative log-likelihood function part in the objective (7) is not uniquely identifiable with respect to β_i's because we can simply add any constant c to all the β_i's and subtract cx_j from all the d_j's, while having the same fit.

Denote the true log-fold-change in expression of gene i as β_i^* and the minimizer to the objective function (7) as $\hat{\beta}_i$, $i = 1, 2, \ldots, m$. In the absence of noise, a necessary condition for $\hat{\beta}_i = \beta_i^*$, $i = 1, 2, \ldots, m$, is

$$\left\|\boldsymbol{\beta}^*\right\|_1 = \sum_{i=1}^m \left|\beta_i^*\right| \leq \sum_{i=1}^m \left|\beta_i^* + c\right| = \left\|\boldsymbol{\beta}^* + c\right\|_1 \tag{100}$$

for any constant $c \neq 0$. Otherwise, $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + c_0$ will be the minimizer of (7), where

$$c_0 = \arg\min_c \sum_{i=1}^m \left|\beta_i^* + c\right|. \tag{101}$$

Assume

non-DE genes	$\beta_i = 0$
up-regulated genes:	$\beta_i \sim \mathcal{N}(\mathrm{fc_{up}}, 1)$
down-regulated genes:	$\beta_i \sim \mathcal{N}(\mathrm{fc_{down}}, 1)$

where $\mathrm{fc_{up}}$ and $\mathrm{fc_{down}}$ are the mean log fold-change in expression of the up- and down-regulated DE genes, respectively.

Let $\mathrm{up}\%$ and $\mathrm{down}\%$ be the percent of up- and down-regulated DE genes respectively. It follows that the percent of non-DE genes is $(1 - \mathrm{up}\% - \mathrm{down}\%)$. The left-hand side and right-hand side of inequality (100) can be approximated as

$$\frac{1}{m}\left\|\boldsymbol{\beta}^*\right\|_1 \approx \mathrm{up}\% \left|\mathrm{fc_{up}}\right| + \mathrm{down}\% \left|\mathrm{fc_{down}}\right|, \tag{102}$$

$$\begin{aligned}\frac{1}{m}\left\|\boldsymbol{\beta}^* + c\right\|_1 \approx\ & \mathrm{up}\% \left|\mathrm{fc_{up}} + c\right| + \mathrm{down}\% \left|\mathrm{fc_{down}} + c\right| \\ & + (1 - \mathrm{up}\% - \mathrm{down}\%)\left|c\right|.\end{aligned} \tag{103}$$

where we have ignored the irrelevant coefficient α.

It can be shown that inequality (100) holds only if $\max(\mathrm{up}\%, \mathrm{down}\%) \leq 50\%$, which is a necessary condition under which we can correctly estimate β_i's.

In Figure 1(i), the percent of up- and down-regulated DE genes are respectively

$$\mathrm{up}\% = 70\% \times 90\% = 63\%,\ \mathrm{down}\% = 70\% \times (1 - 90\%) = 7\%$$

In this case, we have $\max(\mathrm{up}\%, \mathrm{down}\%) = 63\% > 50\%$ and thus inequality (100) does not hold. Consequently, the solution $\hat{\beta}_i$ for every gene $i = 1, 2, \ldots, m$ is a shifted version of β_i^*, i.e., $\hat{\beta}_i = \beta_i^* + c_0$ where c_0 is defined in (101).

Table S1: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. Number of samples: $n = 5$, log-fold change for DE genes: $\beta_i \sim \mathcal{N}(\pm 2, 1)$, and noise level: $\sigma_i = 0.1$. The table shows the percent of DE genes (DE %), percent of up-regulated genes among the DE genes (Up %), as well as the mean AUCs for all four methods measured using 10 simulated replicates. The standard errors of the mean AUCs are given in parentheses.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	**0.9921**	0.5582	0.9921	0.992
		(0.0016)	(0.0888)	(0.0015)	(0.0016)
10	70	0.9889	0.7292	**0.9893**	0.9892
		(0.0015)	(0.0801)	**(0.0015)**	(0.0015)
10	90	0.9882	0.6401	0.9882	**0.9888**
		(0.0019)	(0.1046)	(0.0019)	**(0.0018)**
30	50	0.9898	0.4654	0.9902	**0.9903**
		(0.0017)	(0.104)	(0.0015)	**(0.0013)**
30	70	0.9878	0.5882	**0.9882**	0.9881
		(0.0014)	(0.1023)	**(0.0012)**	(0.0012)
30	90	0.9885	0.7714	**0.989**	0.9806
		(0.0017)	(0.0636)	**(0.0016)**	(0.0016)
50	50	0.9926	0.5916	0.993	**0.9931**
		(8e-04)	(0.0865)	(9e-04)	**(8e-04)**
50	70	0.9812	0.5552	0.9816	**0.9835**
		(0.003)	(0.0846)	(0.003)	**(0.0015)**
50	90	0.9143	0.6331	0.9166	**0.9657**
		(0.0077)	(0.0815)	(0.0075)	**(0.0025)**
70	50	0.988	0.6866	0.9883	**0.9907**
		(0.0012)	(0.0791)	(0.0013)	**(9e-04)**
70	70	0.9039	0.7063	0.9088	**0.9604**
		(0.0064)	(0.0862)	(0.0057)	**(0.0013)**
70	90	0.4804	0.4447	**0.5326**	0.4692
		(0.0314)	(0.0173)	**(0.0426)**	(0.0148)

Table S2: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S1 except that the number of samples is $n = 8$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.9846	0.3798	**0.9855**	0.9854
		(0.0035)	(0.0611)	**(0.0036)**	(0.0036)
10	70	0.9924	0.4826	0.9926	**0.9929**
		(0.0014)	(0.0711)	(0.0015)	**(0.0014)**
10	90	0.9923	0.6162	0.9926	**0.9932**
		(0.0021)	(0.071)	(0.0019)	**(0.002)**
30	50	0.9897	0.5664	**0.9907**	0.9906
		(0.0016)	(0.0864)	**(0.0017)**	(0.0017)
30	70	0.9898	0.7494	0.9905	**0.9913**
		(0.0014)	(0.062)	(0.0013)	**(0.0012)**
30	90	0.9822	0.5462	**0.9831**	0.9788
		(0.0017)	(0.0905)	**(0.0019)**	(0.0018)
50	50	0.9919	0.5588	0.9922	**0.9923**
		(8e-04)	(0.0758)	(7e-04)	**(7e-04)**
50	70	0.9797	0.5008	0.9801	**0.9854**
		(0.002)	(0.0965)	(0.0019)	**(0.0013)**
50	90	0.904	0.4788	0.9048	**0.9639**
		(0.0087)	(0.0746)	(0.0084)	**(0.0012)**
70	50	0.9892	0.5846	0.9899	**0.9918**
		(0.0012)	(0.0915)	(0.001)	**(5e-04)**
70	70	0.8437	0.4263	0.8699	**0.9598**
		(0.0269)	(0.0529)	(0.013)	**(0.0036)**
70	90	0.3639	**0.4783**	0.3971	0.4549
		(0.0183)	**(0.0162)**	(0.0342)	(0.0098)

Table S3: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S1 except that the number of samples is $n = 25$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.9882	0.509	0.9889	**0.9889**
		(0.0026)	(0.1007)	(0.0024)	**(0.0023)**
10	70	0.9928	0.4839	**0.9932**	0.993
		(0.0022)	(0.0774)	**(0.0019)**	(0.0021)
10	90	0.9897	0.4427	0.9901	**0.9909**
		(0.0022)	(0.0762)	(0.0022)	**(0.0023)**
30	50	**0.9897**	0.7324	0.989	0.989
		(0.0018)	(0.0847)	(0.0017)	(0.0017)
30	70	0.9849	0.6607	0.9859	**0.9891**
		(0.0012)	(0.0652)	(0.0011)	**(0.001)**
30	90	0.9734	0.6233	0.9735	**0.9803**
		(0.0032)	(0.073)	(0.0032)	**(0.0012)**
50	50	0.9917	0.6281	**0.9922**	0.9921
		(8e-04)	(0.0941)	**(9e-04)**	(9e-04)
50	70	0.9674	0.4515	0.9681	**0.9848**
		(0.0041)	(0.0611)	(0.0043)	**(0.0014)**
50	90	0.8408	0.5486	0.8532	**0.9655**
		(0.0099)	(0.0538)	(0.0085)	**(0.0015)**
70	50	0.9892	0.5593	0.9898	**0.9914**
		(0.0014)	(0.0862)	(0.0014)	**(0.001)**
70	70	0.8496	0.4371	0.8696	**0.9639**
		(0.0161)	(0.0446)	(0.0123)	**(0.0031)**
70	90	0.3291	**0.4821**	0.3107	0.476
		(0.0051)	**(0.0148)**	(0.0048)	(0.0147)

Table S4: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S1 except that the number of samples is $n = 50$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.9888	0.5022	0.9887	**0.9889**
		(0.0024)	(0.0836)	(0.0026)	**(0.0025)**
10	70	0.9874	0.4791	0.987	**0.9875**
		(0.0046)	(0.0608)	(0.0045)	**(0.0045)**
10	90	0.9842	0.5424	0.9852	**0.988**
		(0.0031)	(0.0569)	(0.0036)	**(0.0024)**
30	50	**0.9901**	0.5436	0.9897	0.99
		(0.0012)	(0.0772)	(0.0011)	(0.001)
30	70	0.9843	0.5553	0.9859	**0.9922**
		(0.0018)	(0.0624)	(0.0018)	**(0.001)**
30	90	0.971	0.4762	0.9718	**0.9807**
		(0.003)	(0.0406)	(0.003)	**(0.0013)**
50	50	0.9912	0.457	0.9914	**0.992**
		(8e-04)	(0.0545)	(9e-04)	**(8e-04)**
50	70	0.9544	0.4791	0.9551	**0.9813**
		(0.0025)	(0.0351)	(0.0024)	**(0.0016)**
50	90	0.8312	0.4675	0.8467	**0.9696**
		(0.0158)	(0.0316)	(0.0131)	**(0.0019)**
70	50	0.99	0.5387	0.9899	**0.9911**
		(9e-04)	(0.0729)	(9e-04)	**(0.001)**
70	70	0.84	0.4268	0.8598	**0.963**
		(0.0158)	(0.0323)	(0.0114)	**(0.0023)**
70	90	0.3124	**0.5109**	0.2997	0.4335
		(0.005)	**(0.0106)**	(0.0044)	(0.021)

Table S5: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S1 except that the number of samples is $n = 100$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.9912	0.4659	0.9912	**0.9912**
		(0.0015)	(0.0616)	(0.0019)	**(0.0017)**
10	70	0.9856	0.5762	0.9875	**0.9888**
		(0.003)	(0.0688)	(0.0032)	**(0.0028)**
10	90	0.9865	0.456	0.9868	**0.9892**
		(0.0032)	(0.0482)	(0.0027)	**(0.0025)**
30	50	0.9925	0.5247	**0.9928**	0.9924
		(0.001)	(0.0468)	**(0.0011)**	(0.0012)
30	70	0.9785	0.4166	0.9785	**0.9882**
		(8e-04)	(0.0627)	(0.001)	**(0.0011)**
30	90	0.9433	0.581	0.9441	**0.9806**
		(0.004)	(0.0626)	(0.004)	**(0.0022)**
50	50	0.9908	0.4967	0.9912	**0.9918**
		(0.0012)	(0.0699)	(0.0013)	**(0.0012)**
50	70	0.9492	0.5253	0.9501	**0.9817**
		(0.0062)	(0.0564)	(0.0062)	**(0.0012)**
50	90	0.7752	0.4987	0.8031	**0.971**
		(0.0184)	(0.0329)	(0.013)	**(0.0021)**
70	50	0.9901	0.5089	0.9906	**0.992**
		(8e-04)	(0.05)	(9e-04)	**(7e-04)**
70	70	0.82	0.5221	0.8453	**0.9633**
		(0.0116)	(0.032)	(0.0086)	**(0.0027)**
70	90	0.3082	**0.498**	0.294	0.4511
		(0.0024)	**(0.0074)**	(0.0016)	(0.0041)

Table S6: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. Number of samples: $n = 5$, log-fold change for DE genes: $\beta_i \sim \mathcal{N}(\pm 0.2, 0.1)$, and noise level: $\sigma_i = 0.1$. The table shows the percent of DE genes (DE %), percent of up-regulated genes among the DE genes (Up %), as well as the mean AUCs for all four methods measured using 10 simulated replicates. The standard errors of the mean AUCs are given in parentheses.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.8314	0.5755	0.8357	**0.8358**
		(0.0075)	(0.0415)	(0.0072)	**(0.0074)**
10	70	0.7948	0.6377	0.8071	**0.8119**
		(0.0096)	(0.0323)	(0.0095)	**(0.009)**
10	90	0.7914	0.6058	0.8025	**0.8099**
		(0.0097)	(0.0405)	(0.0095)	**(0.0106)**
30	50	0.812	0.5429	0.8153	**0.8161**
		(0.0063)	(0.0347)	(0.0055)	**(0.0049)**
30	70	0.7883	0.5633	0.7942	**0.7996**
		(0.0063)	(0.0325)	(0.0066)	**(0.006)**
30	90	0.7482	0.6117	0.7515	**0.7688**
		(0.0073)	(0.0271)	(0.0068)	**(0.0048)**
50	50	0.8156	0.5871	**0.8242**	0.824
		(0.0059)	(0.0282)	**(0.0057)**	(0.0057)
50	70	0.7676	0.5648	0.7743	**0.7865**
		(0.0062)	(0.0247)	(0.005)	**(0.0048)**
50	90	0.6523	0.5523	0.6582	**0.6746**
		(0.0049)	(0.0199)	(0.0054)	**(0.0081)**
70	50	0.8042	0.5752	0.8106	**0.8128**
		(0.0039)	(0.0347)	(0.0038)	**(0.0039)**
70	70	0.7339	0.5832	0.7367	**0.7375**
		(0.0055)	(0.0291)	(0.0053)	**(0.0045)**
70	90	0.5437	0.5268	0.5422	**0.5509**
		(0.0066)	(0.0102)	(0.0069)	**(0.007)**

Table S7: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S6 except that the number of samples is $n = 8$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.7946	0.4908	0.801	**0.8016**
		(0.0071)	(0.0133)	(0.0085)	**(0.0086)**
10	70	0.795	0.5039	0.8017	**0.8038**
		(0.0106)	(0.0112)	(0.0093)	**(0.0088)**
10	90	0.7884	0.5522	0.7942	**0.8015**
		(0.0095)	(0.0197)	(0.0099)	**(0.0088)**
30	50	0.8043	0.5232	0.8122	**0.8125**
		(0.0039)	(0.0274)	(0.0047)	**(0.0049)**
30	70	0.795	0.5747	0.8048	**0.811**
		(0.0035)	(0.021)	(0.0035)	**(0.0025)**
30	90	0.7329	0.5318	0.7412	**0.7558**
		(0.0088)	(0.0235)	(0.0086)	**(0.0092)**
50	50	0.8063	0.5365	0.8161	**0.8172**
		(0.0039)	(0.0147)	(0.0041)	**(0.0038)**
50	70	0.7696	0.5375	0.7787	**0.7862**
		(0.0055)	(0.0263)	(0.0049)	**(0.0048)**
50	90	0.6609	0.4991	0.6651	**0.6914**
		(0.0068)	(0.0109)	(0.0052)	**(0.0049)**
70	50	0.8014	0.5437	0.8137	**0.8153**
		(0.0056)	(0.0252)	(0.0049)	**(0.0048)**
70	70	0.7308	0.4998	0.7372	**0.7414**
		(0.0039)	(0.0106)	(0.0038)	**(0.0046)**
70	90	0.5382	0.5043	0.5418	**0.5446**
		(0.0056)	(0.0065)	(0.005)	**(0.0072)**

Table S8: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S6 except that the number of samples is $n = 25$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.8038	0.4962	0.8127	**0.8128**
		(0.0102)	(0.0144)	(0.0091)	**(0.0093)**
10	70	0.7984	0.4976	0.8052	**0.8073**
		(0.0088)	(0.0136)	(0.0094)	**(0.009)**
10	90	0.7861	0.4976	0.8016	**0.8103**
		(0.0088)	(0.0152)	(0.0081)	**(0.0073)**
30	50	0.8014	0.5307	0.811	**0.8123**
		(0.0049)	(0.0158)	(0.0051)	**(0.005)**
30	70	0.7905	0.5166	0.7974	**0.8048**
		(0.0039)	(0.0111)	(0.0031)	**(0.0039)**
30	90	0.7283	0.5088	0.7437	**0.7624**
		(0.0055)	(0.0099)	(0.0075)	**(0.007)**
50	50	0.799	0.5278	0.8129	**0.8135**
		(0.005)	(0.0138)	(0.0044)	**(0.0043)**
50	70	0.7656	0.4929	0.78	**0.7895**
		(0.0048)	(0.0095)	(0.0039)	**(0.0048)**
50	90	0.6426	0.5041	0.6482	**0.6805**
		(0.0062)	(0.0054)	(0.0064)	**(0.0078)**
70	50	0.7952	0.5245	**0.8068**	0.8068
		(0.004)	(0.016)	**(0.0042)**	(0.004)
70	70	0.7323	0.5018	**0.7417**	0.7361
		(0.0069)	(0.0068)	**(0.0064)**	(0.0066)
70	90	0.5564	0.4937	**0.558**	0.5572
		(0.0072)	(0.0088)	**(0.0089)**	(0.0106)

Table S9: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S6 except that the number of samples is $n = 50$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.8002	0.4991	0.8045	**0.8068**
		(0.0073)	(0.0126)	(0.0063)	**(0.0058)**
10	70	0.789	0.4932	0.8069	**0.81**
		(0.0141)	(0.0055)	(0.0121)	**(0.0123)**
10	90	0.7788	0.4851	0.7941	**0.8024**
		(0.0083)	(0.0088)	(0.0111)	**(0.0093)**
30	50	0.7935	0.5044	0.808	**0.8086**
		(0.0048)	(0.0058)	(0.0052)	**(0.0049)**
30	70	0.7912	0.4995	0.8029	**0.809**
		(0.0063)	(0.0044)	(0.0056)	**(0.005)**
30	90	0.7312	0.4944	0.7373	**0.765**
		(0.006)	(0.0051)	(0.0067)	**(0.0072)**
50	50	0.8098	0.5023	0.8205	**0.822**
		(0.0056)	(0.0064)	(0.0047)	**(0.0044)**
50	70	0.7625	0.5037	0.7732	**0.7851**
		(0.004)	(0.0067)	(0.0035)	**(0.0049)**
50	90	0.6468	0.5118	0.6527	**0.6887**
		(0.0048)	(0.0048)	(0.0049)	**(0.0043)**
70	50	0.8026	0.5028	**0.8141**	0.8135
		(0.0032)	(0.0066)	**(0.0041)**	(0.004)
70	70	0.735	0.4913	**0.7451**	0.7384
		(0.0064)	(0.0094)	**(0.006)**	(0.0068)
70	90	0.5382	0.5019	0.5404	**0.5432**
		(0.007)	(0.0075)	(0.0072)	**(0.0091)**

Table S10: AUC comparison of edgeR-robust, DESeq2, limma-voom and ELMSeq in log-normally distributed data. The data generation parameters are the same as those in Table S6 except that the number of samples is $n = 100$.

DE (%)	Up (%)	edgeR - robust	DESeq2	limma - voom	ELMSeq
10	50	0.8086	0.4995	0.8215	**0.8217**
		(0.0066)	(0.0111)	(0.0071)	**(0.0072)**
10	70	0.7822	0.4936	0.798	**0.8017**
		(0.0117)	(0.0112)	(0.0105)	**(0.0109)**
10	90	0.8017	0.4998	0.8138	**0.8246**
		(0.0101)	(0.0083)	(0.0102)	**(0.0085)**
30	50	0.8101	0.4975	0.8212	**0.8213**
		(0.0061)	(0.0051)	(0.0065)	**(0.0065)**
30	70	0.7783	0.4921	0.792	**0.7968**
		(0.0036)	(0.0084)	(0.0037)	**(0.0039)**
30	90	0.7175	0.5052	0.7268	**0.7549**
		(0.0066)	(0.0062)	(0.0056)	**(0.0069)**
50	50	0.8006	0.5024	0.8108	**0.8121**
		(0.0047)	(0.007)	(0.0041)	**(0.004)**
50	70	0.763	0.4895	0.7753	**0.7835**
		(0.0078)	(0.003)	(0.0078)	**(0.0059)**
50	90	0.6362	0.5	0.645	**0.68**
		(0.0053)	(0.0041)	(0.0049)	**(0.0054)**
70	50	0.8056	0.4962	0.8161	**0.8167**
		(0.0056)	(0.0078)	(0.005)	**(0.0053)**
70	70	0.7418	0.495	**0.7524**	0.7451
		(0.0046)	(0.0058)	**(0.0052)**	(0.0045)
70	90	0.5455	0.5111	0.5416	**0.5481**
		(0.0051)	(0.0091)	(0.0059)	**(0.0051)**