

ADDITIONAL FILE 1

Supplement to: Data-based RNA-seq Simulations by Binomial Thinning

David Gerard

Correspondence:

dgerard@american.edu

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

Full list of author information is
available at the end of the article

Abstract

This document contains theoretical considerations, additional simulation details, and supplementary figures to “Data-based RNA-seq Simulations by Binomial Thinning.”

S1 Theoretical Considerations

S1.1 Target Correlation

Theorem S1 *Let (u_i, z_i) be iid jointly standard normal with correlation ρ for $i = 1, 2, \dots, n$. Let w_j be iid standard normal for $j = 1, 2, \dots, n$. Suppose we match the w_j 's onto the u_i 's by order statistics, resulting in (w_i, u_i) pairs such that the rank of w_i is the same as the rank of u_i . Then $\text{cor}(w_i, z_i) \xrightarrow[n \rightarrow \infty]{} \rho$.*

Proof For a fixed proportion p , we note that $u_{(\lceil np \rceil)}$ and $w_{(\lceil np \rceil)}$ converge in probability to the theoretical p -quantile of the standard normal distribution [1, e.g.]. Since the order statistics converge to the same values, and we match by order statistics, this implies that $u_i - w_i \xrightarrow{P} 0$. Thus, by Slutsky's theorem, we have that $u_i z_i - w_i z_i \xrightarrow{P} 0$.

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

$$\mathbb{E}[|u_i z_i| + |w_i z_i|] \leq \mathbb{E}[u_i^2]^{1/2} \mathbb{E}[z_i^2]^{1/2} + \mathbb{E}[w_i^2]^{1/2} \mathbb{E}[z_i^2]^{1/2} = 2. \quad (\text{S1})$$

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

To place the results of Theorem S1 in context of the matching in Procedure 2, note that the u_i 's are the elements of \mathbf{U} , the z_i 's are the elements of $\hat{\mathbf{Z}}$, the w_i 's are the elements of \mathbf{X}_3 , ρ is the target correlation between the one column in \mathbf{X}_3 and the one column in $\hat{\mathbf{Z}}$, and $\mathbf{\Pi}$ is the permutation matrix that results in the matching of the w_i 's and the u_i 's.

The results of Theorem S1 can be generalized to non-standard normal distributions by appealing to the weak law of large numbers and Slutsky's theorem.

S1.2 Generalizing the Poisson Assumption

For simplicity, we stated a Poisson distribution as the modeling assumption in (1). However, our methods are equally valid under more general conditions. We begin by showing how our methods are valid when using the negative binomial distribution, which is perhaps the most common distribution used to analyze RNA-seq counts [2–4]. To see this, we prove the following simple lemma which, though less well-known than Lemma 1, can still be found in some elementary texts (or at least a version of the following lemma) [exercise 4.32 of 5, e.g.].

Lemma S1 *Suppose $y \sim \text{NB}(\mu, \phi)$, where we are using the parameterization such that $E[y] = \mu$ and $\text{var}(y) = \mu(1 + \mu\phi)$. Also suppose that $\tilde{y}|y \sim \text{Bin}(y, p)$. Then $\tilde{y} \sim \text{NB}(p\mu, \phi)$.*

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

$$\lambda \sim \text{Gamma}(1/\phi, \mu\phi) \quad (\text{S2})$$

$$y|\lambda \sim \text{Poisson}(\lambda), \quad (\text{S3})$$

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

The zero-inflated negative binomial distribution is sometimes used to model single-cell RNA-seq data as it can account for the abundance of zeros observed in such data [6–8]. A random variable y is distributed zero-inflated negative binomial, denoted $y \sim \text{ZINB}(\pi, \mu, \phi)$, if it is generated by the following hierarchical process:

$$z \sim \text{Bern}(1 - \pi), \quad (\text{S4})$$

$$y|z = 0 \sim \delta_0 \quad (\text{S5})$$

$$y|z = 1 \sim \text{NB}(\mu, \phi), \quad (\text{S6})$$

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

Lemma S2 *Suppose $y \sim \text{ZINB}(\pi, \mu, \phi)$ and $\tilde{y}|y \sim \text{Bin}(y, p)$. Then $\tilde{y} \sim \text{ZINB}(\pi, p\mu, \phi)$.*

Proof It's sufficient to note that

$$\tilde{y}|z = 0 \sim \delta_0, \text{ and} \quad (\text{S7})$$

$$\tilde{y}|z = 1 \sim \text{NB}(p\mu, \phi). \quad (\text{S8})$$

\square

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

Lemma S3 *Let $\pi_0, \pi_1, \dots, \pi_M$ and $\tau_1, \tau_2, \dots, \tau_L$ be non-negative mixing proportions such that*

$$\sum_{m=0}^M \pi_m + \sum_{\ell=1}^L \tau_\ell = 1. \quad (\text{S9})$$

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

$$f(y) = \pi_0 \delta_0(y) + \sum_{m=1}^M \pi_m \text{NB}(y|\mu_m, \phi_m) + \sum_{\ell=1}^L \tau_\ell \text{Bin}(y|\frac{\nu_\ell}{n_\ell}, n_\ell), \quad (\text{S10})$$

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

$$\mathbb{E}[\tilde{y}] = p \mathbb{E}[y], \text{ and} \quad (\text{S11})$$

$$f(\tilde{y}) = \pi_0 \delta_0(\tilde{y}) + \sum_{m=1}^M \pi_m \text{NB}(\tilde{y}|p\mu_m, \phi_m) + \sum_{\ell=1}^L \tau_\ell \text{Bin}(\tilde{y}|\frac{p\nu_\ell}{n_\ell}, n_\ell). \quad (\text{S12})$$

Proof Equation (S11) is just a consequence of the law of total expectation. To prove (S12), note that if $y \sim \text{NB}(\mu_m, \phi_m)$ then $\tilde{y} \sim \text{NB}(p\mu_m, \phi_m)$ and if $y \sim \text{Bin}(\nu_\ell/n_\ell, n_\ell)$ then $\tilde{y} \sim \text{Bin}(p\nu_\ell/n_\ell, n_\ell)$. The proof follows by conditioning on the latent mixing group. \square

S2 Additional Simulations

S2.1 Correlation Estimator

We explored the effects of changing the target correlation on the true correlation. We varied the sample size, $N \in \{6, 10, 20, 500\}$, and the target correlations between \mathbf{z} and the two columns in $\mathbf{\Pi X}_3$, $\mathbf{r} \in \{(0, 0), (0.5, 0), (0.9, 0), (0.5, 0.5)\}$. We did not implement simulations for $\mathbf{r} = (0.5, 0.9)$ because the resulting correlation matrix would not be positive definite. Under each unique combination of simulation parameter settings, we iterative drew $\mathbf{z} \in \mathbb{R}^N$ from a standard normal. We also drew $\mathbf{X}_3 \in \mathbb{R}^{N \times 2}$ according to two schemes:

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

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

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

S2.2 Single Cell Factor Analysis Simulations

In this section, we evaluate the same five factor analysis methods from Section 3.3 using a single cell RNA-seq dataset. We used a dataset of Peripheral Blood Mononuclear Cells (PBMC) from 10X Genomics [9], providing the same filters as the tutorial for Seurat [10, 11], available at https://satijalab.org/seurat/v3.1/pbmc3k_tutorial.html. Specifically, we removed cells that have unique feature counts either over 2500 or less than 200, and we removed cells that have more than 5% mitochondrial counts. After filtering cells, we kept the 4000 features with the greatest cell-to-cell variation in the dataset, where variation was measured after normalizing each cell. This resulted in a dataset of 4000 features on 2638 cells.

To simulate single-cell RNA-seq data, we took the filtered PBMC data, assumed model (9), then generated data using `seqgendiff` such that (19) holds. We simulated the components of \mathbf{x}_3 and the non-zero components of \mathbf{b}_3 from independent normal distributions. We varied the following parameters of the simulation study:

- 1 The signal strength: the standard deviation of the loadings (the b_{3g} 's) was set to one of $\{0.4, 0.8\}$, with higher standard deviations corresponding to higher signal.
- 2 The sparsity: the proportion of loadings (the b_{3g} 's) that are 0 was set to one of $\{0, 0.9\}$, and
- 3 The target correlations of the added factor with the first unobserved factor: $r \in \{0, 0.5\}$.

We fixed the number of genes to be 1000 and the number of cells to be 500. Thus, we had eight different simulation settings under evaluation. We ran 100 replications for each simulation setting, each time fitting one of the five factor analysis methods. We applied all factor analysis methods to the \log_2 -counts, after adding half a pseudo-count. The number of factors was estimated using parallel analysis [12].

We used the same three metrics to evaluate the different factor analysis methods as in Section 3.3. The results are presented in Figures S18, S19, and S20. We have the following conclusions:

- 1 PEER generally performed poorly, only excelling in estimating the loadings in high-signal and low-sparsity regimes.
- 2 ICA had difficulty estimating the factors in high sparsity, high correlation regimes.
- 3 PCA, SSVD, and *flash* all performed comparably.

Based on these conclusions, we would suggest researchers explore PCA, SSVD, or *flash*.

S2.3 SimSeq Simulations

We ran the **SimSeq** simulation software [13] using the GTEx muscle data [14]. **SimSeq** requires the presence of a group indicator variable to create differentially expressed genes. So we used the individual's sex as this indicator variable. We simulated data so that 90% of genes are non-differentially expressed, using 10000 genes and a sample of size 10 individuals each replication (5 for each group). For each of 500 repetitions, we calculated the false positive rate and the power when using DESeq2 [4], edgeR [15], and voom-limma [16].

Since the performance of **SimSeq** depends on the quality of the indicator variable used, we estimated the proportion of non-null genes between sexes using voom-limma-eBayes [16] and *ash* [17]. *Ash* estimated that 98.2% of genes are differentially expressed, indicating that sex is an appropriate indicator variable to use for **SimSeq**. Many of these observed associations are likely due to unwanted variation [18]. But for the purposes of the **SimSeq** simulation method, it is only important to have an indicator variable that is marginally associated with gene expression.

SimSeq is unable to control the strength of the signal between two groups. So for accurate comparison with **seqgendiff**, we needed to estimate the distribution of effect sizes. We used *ash* to estimate the standard deviation of effect sizes in the GTEx muscle data, resulting in an estimate of 0.334. We then ran **seqgendiff** using a sample size of 10 (5 for each of two groups), 10000 genes, setting 90% of the genes to be null, and drawing the effect sizes of the non-null genes from a normal distribution with mean 0 and standard deviation 0.334. We did this for 500 repetitions, each time calculating the false positive rate and power when using DESeq2, edgeR, and voom-limma.

The results are presented in Figure S16. There, we see that false positive rates and powers are distributionally similar when using either **seqgendiff** or **SimSeq**. Thus, when **seqgendiff** uses effect sizes approximately similar to the indicator variable that **SimSeq** uses, the results are about the same.

References

1. Arnold BC, Balakrishnan N, Nagaraja HN. A first course in order statistics. vol. 54. Siam; 1992.
2. Robinson MD, Smyth GK. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*. 2007 09;23(21):2881–2887.
3. Robinson MD, Smyth GK. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*. 2007 08;9(2):321–332.
4. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*. 2014 Dec;15(12):550.
5. Casella G, Berger RL. Statistical inference. vol. 2. Duxbury Pacific Grove, CA; 2002.
6. Miao Z, Deng K, Wang X, Zhang X. DEsingle for detecting three types of differential expression in single-cell RNA-seq data. *Bioinformatics*. 2018 04;34(18):3223–3224.
7. Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert JP. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature communications*. 2018;9(1):284.
8. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications*. 2019;10(1):390.
9. Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*. 2017;8:14049.
10. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*. 2018;36(5):411.
11. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck III WM, et al. Comprehensive Integration of Single-Cell Data. *Cell*. 2019;.
12. Buja A, Eyuboglu N. Remarks on parallel analysis. *Multivariate behavioral research*. 1992;27(4):509–540.
13. Benidt S, Nettleton D. SimSeq: a nonparametric approach to simulation of RNA-sequence datasets. *Bioinformatics*. 2015 02;31(13):2131–2140.
14. GTEx Consortium. Genetic effects on gene expression across human tissues. *Nature*. 2017;550(7675):204.
15. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*. 2009 11;26(1):139–140.

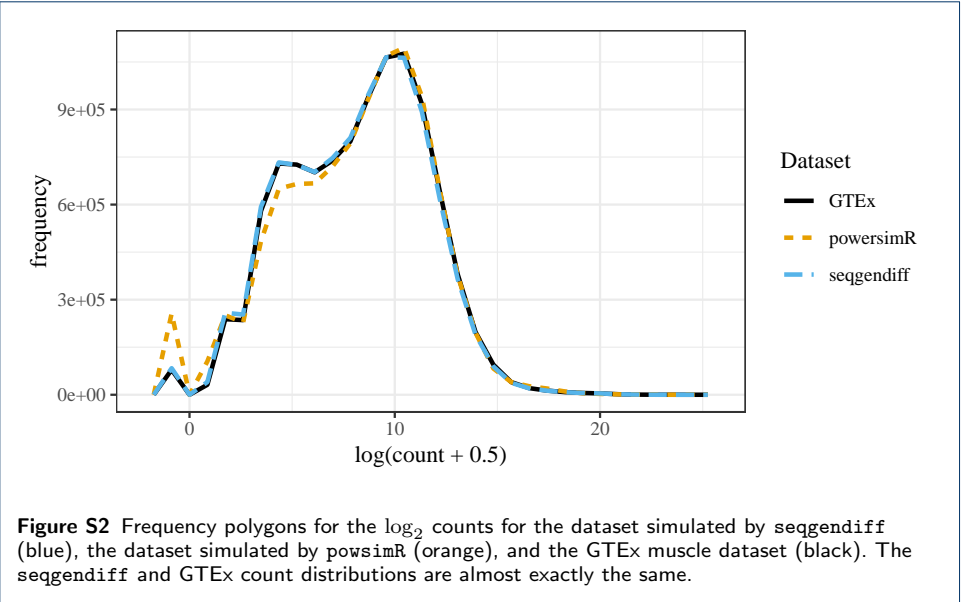
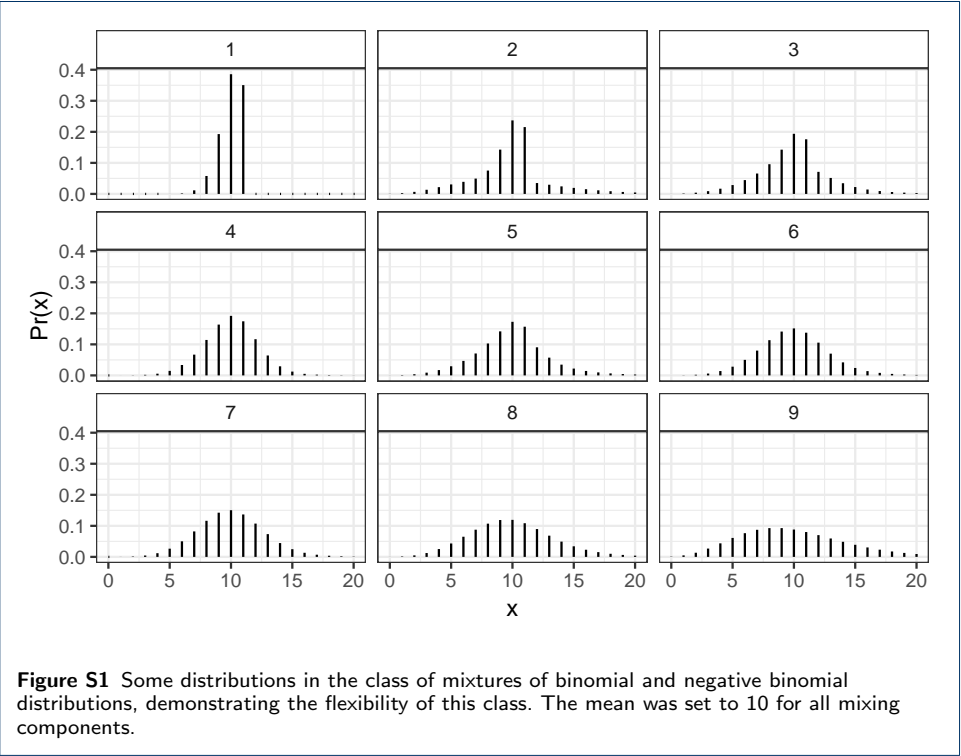
16. Law CW, Chen Y, Shi W, Smyth GK. Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome biology*. 2014;15(R29).

17. Stephens M. False discovery rates: a new deal. *Biostatistics*. 2016 10;18(2):275–294.

18. Leek JT, Scharpf RB, Bravo HC, Simcha D, Langmead B, Johnson WE, et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*. 2010;11(10):733–739.

19. Smyth GK. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*. 2004;3(1).

Supplementary Figures



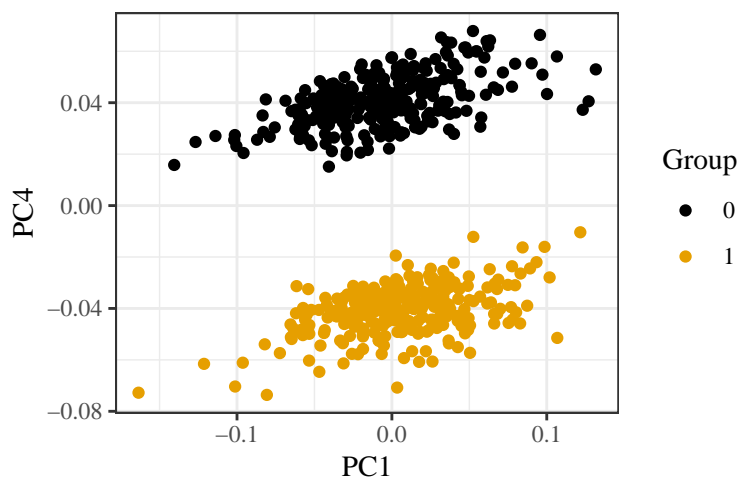


Figure S3 First principle component (x -axis) versus the fourth principle component (y -axis) in the seggendiff dataset. The fourth principle component seems capture group membership.

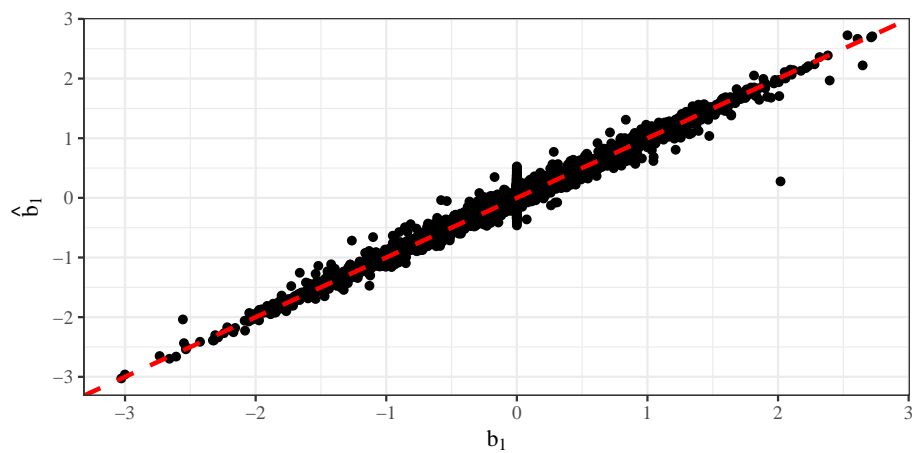


Figure S4 True coefficient values (x -axis) versus their corresponding estimates (y -axis) in the seggendiff dataset. Estimates were obtained using the voom-limma pipeline [16, 19].

