Depth normalization for single-cell genomics count data

A. Sina Booeshaghi[1], Ingileif B. Hallgrímsdóttir[2], Ángel Gálvez-Merchán[2], Lior Pachter[2,3,*]

[1]Department of Mechanical Engineering, California Institute of Technology, Pasadena, CA
[2]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA
[3]Department of Computing and Mathematical Sciences, Pasadena, CA
[*]Address correspondence to: lpachter@caltech.edu

Single-cell genomics analysis requires normalization of feature counts that stabilizes variance while accounting for variable cell sequencing depth. We discuss some of the trade-offs present with current widely used methods, and analyze their performance on 526 single-cell RNA-seq datasets. The results lead us to recommend proportional fitting prior to log transformation followed by an additional proportional fitting.

Introduction

A central theme in single-cell RNA-seq "count normalization" is the importance of achieving depth normalization alongside variance stabilization (Vallejos et al. 2017; Evans, Hardin, and Stoebel 2018; Robinson and Oshlack 2010). While variance stabilization has been studied for over 85 years (Bartlett 1936), the question of how to achieve both variance stabilization and depth normalization is unsolved. An important condition that is often overlooked when evaluating normalization and variance-stabilization methods is that structure must be preserved in the data, which is why classic variance stabilizing transformations are monotonic by design .(Doob 1935). This is why the constant transformation, which sets all counts equal to each other and results in a fully variance-stabilized matrix with all cell depths equal, is not a good normalization.

While many methods have been proposed for single-cell RNA-seq normalization (Cole et al. 2019; Tian et al. 2019; You et al. 2021; Lytal, Ran, and An 2020; Borella et al. 2021; Ahlmann-Eltze and Huber 2021; Breda, Zavolan, and van Nimwegen 2021), the approach of equalizing depth for all cells, often to a "size factor" such as ten thousand (CP10k) or one million (CPM), followed by the application of a variance stabilizing transform like log plus one (log1p) is most popular. These methods are implemented in the widely used Seurat[1] and Scanpy (Wolf, Angerer, and Theis 2018) programs, but they do not explicitly model cell depth as a covariate. The recently published sctransform method (Hafemeister and Satija 2019), which has quickly become the most widely used normalization method for single-cell RNA-seq, aims to address the challenge of variance stabilization and depth normalization by transforming data to Pearson residuals derived from a regularized negative binomial regression. This regression-based

[1] The Seurat R toolkit for single cell genomics is unpublished, however the default normalization is described in several "vignettes" on the software website that showcase standard analysis workflows: https://satijalab.org/seurat/articles/get_started.html.

method incorporates sequencing depth as a covariate in a model, rather than utilizing a size factor (Anders and Huber 2010). However, despite the claims in (Hafemeister and Satija 2019),

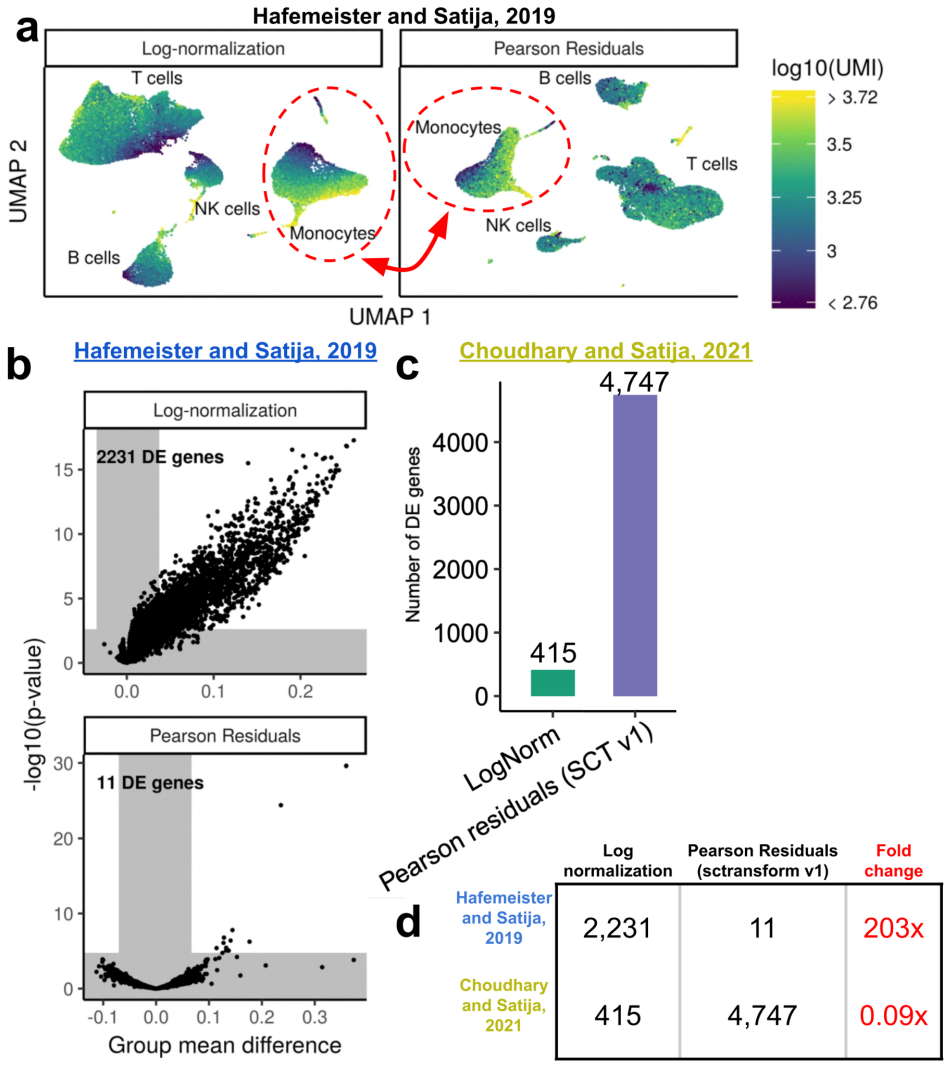

Figure 1: Questions about the efficacy of the sctransform depth normalization. (a) A reproduction of Figure 6 from (Hafemeister and Satija 2019) shows a UMAP generated from the 10x Genomics "33k PBMCs from a Healthy Donor, v1 Chemistry" dataset, where the data has been normalized with the log1pCP10k transform. The figure on the right shows a UMAP generated from the raw data normalized with sctransform. The authors state that "..correlations [between locations of embedded cells and sequencing depth] are strikingly reduced for Pearson residuals [in comparison to log-normalized data" but the difference for Monocytes (circled in red) does not look striking. (b) A differential expression control experiment from (Hafemeister and Satija 2019) showing sctransform greatly reduces false positive genes in comparison to the log transform whereas (c) the opposite is shown in a similar control experiment in (Choudhary and Satija 2022). The figures are all licensed under CC BY 4.0, and have been reproduced from the

papers they were published in with only minor modifications (cropping, the addition of arrows and circles, and addition of number in the plot shown in (c)).

benchmarking of sctransform in (Crowell et al. 2020) shows that the method fails to completely remove the effects of variable depth. The authors show that as a result, sctransform produces “unacceptably high false discovery rates [when used for differential expression]”. Similarly, (Brown et al. 2021) find that sctransform performs poorly (see their Figure 3) and the veracity of the claim in (Hafemeister and Satija 2019) that sctransform “can successfully remove the influence of technical characteristic from downstream analyses” is brought into question by the authors’ own results. Figure 6 of their paper shows a UMAP plot of 33,148 PBMCs that the authors claim displays “a gradient that is correlated with sequencing depth” for log-normalized data, but not for data normalized with sctransform (Fig. 1a).

The figure belies this claim. Contrary to the authors’ assertions, an examination of the plots shows that the Monocytes have a depth gradient with both methods. While this may be due to challenges in interpreting the UMAP embeddings (Chari, Banerjee, and Pachter 2021), it could also be an indication that both methods fail to depth normalize the data. Furthermore, a differential expression benchmark of sctransform in (Hafemeister and Satija 2019) shows that it produces almost no false positives, whereas a similar benchmark in a later paper (Choudhary and Satija 2022) shows the opposite (Fig. 1b, c, d). Aside from questions about depth normalization, it is also unclear whether sctransform is effective at variance stabilization (Ahlmann-Eltze and Huber 2021). These issues raise the question of how effective sctransform, or any other currently used method, is at achieving both depth normalization and variance stabilization.

Furthermore, an analysis of how normalization is used in practice (Supp. Fig. 1), shows that normalization methods are applied in a task-specific manner, resulting in numerous normalizations sometimes being mixed together in a single analysis. For example, sctransform is not, in practice, a single method for computing Pearson residuals from raw counts, but rather a program that implements multiple normalization methods, where each method is used for a different task in the standard Seurat workflow. This highlights the importance of benchmarking the fundamental properties of each normalization technique in a way that is motivated by, and cognizant of, the downstream analysis tasks it may be applied to. In this paper we evaluate several commonly used normalization methods based on how they perform with respect to three criteria that are crucial for common analysis methods: variance stabilization, normalization, and monotonicity of the transformations.

Results

Evaluation criteria

In considering how to evaluate normalization methods, we focused on downstream applications and their respective assumptions. Dimensionality reduction with PCA is an initial step in many analyses that relies on equal gene variances. If variance is not stabilized, genes with a high

variance may have an outsized impact on the singular values solely due to having a high mean (Nguyen and Holmes 2019). Similarly, without depth-normalization, the key step of identifying genes that are differentially expressed between cell types, may yield false-positive genes simply due to certain groups of cells being sampled more deeply than others (Robinson and Oshlack 2010). An additional property of normalization techniques that is important for tasks such as marker gene selection is monotonicity of the transformations, especially for constructing heatmaps or similar visualizations.

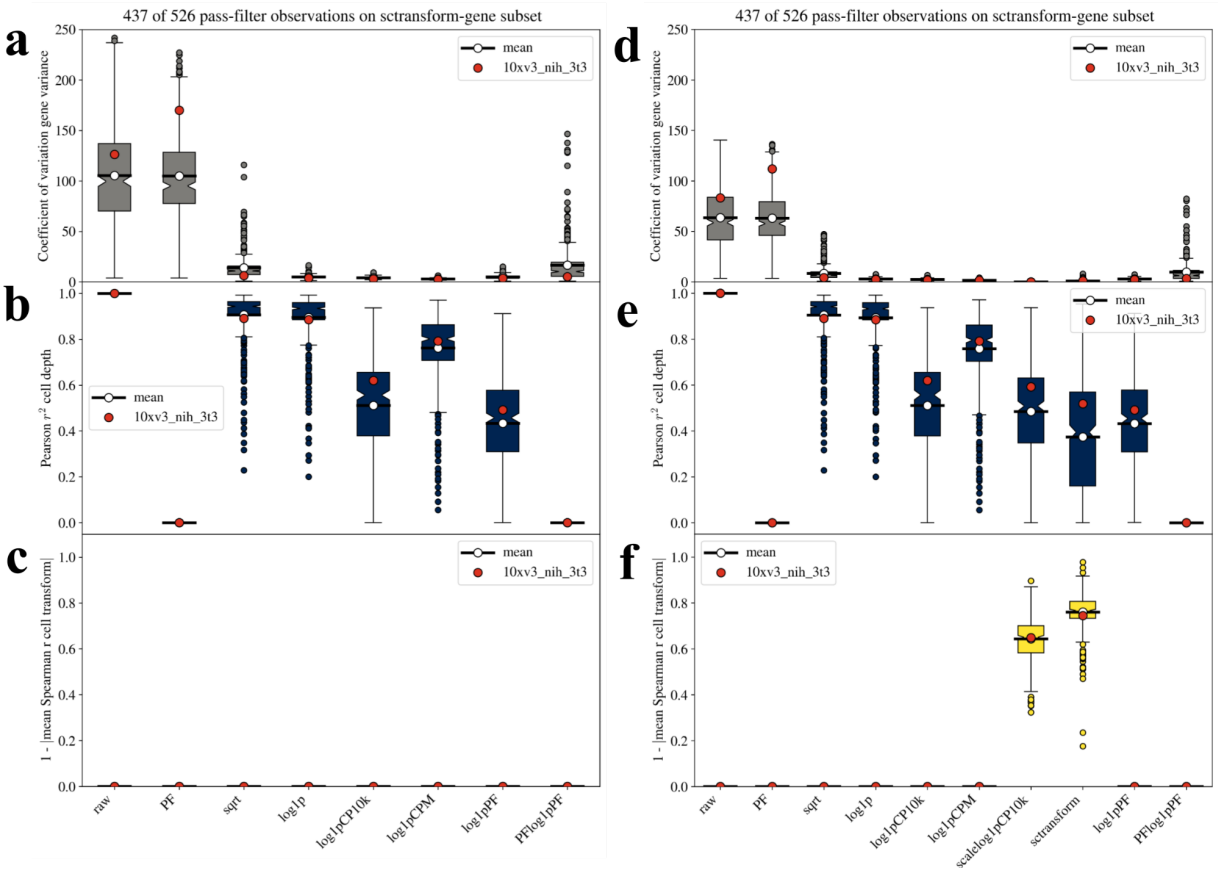

Figure 2: Benchmarking normalization techniques on 437 of 526 datasets passing filter. (a)-(c) demonstrate metrics computed on all genes, a task which is computationally intractable to compute on sctransform and scalelog1pCP10k due to their size. (d)-(f) demonstrate metrics computed on a subset of genes as identified by sctransform's default gene filtering. (Methods). (a) and (d) show the coefficient of variation on the gene variances for each dataset. (b) and (e) show the Pearson r^2 between the raw cell depth and the transformed cell depth. (c) and (f) show one minus the absolute value of the mean Spearman r on the raw vs transformed cell. A bar is plotted to the mean of each distribution (also marked with a red circle). The *10xv3_nih_3t3* dataset is marked with a blue circle.

To assess effectiveness of variance stabilization, we plotted the mean of each gene vs. its variance across cells, and measured the coefficient of variation of the gene variance (CV) after transformation as a scale-independent measure of the effectiveness of variance stabilization.

Depth normalization was assessed by plotting, for each cell, the total raw cell counts vs. the total transformed cell counts. Since the total abundance of a gene per cell may not be measured with respect to an absolute scale, we computed the r^2 correlation with raw cell depth as a proxy for the extent to which raw cell counts were reflected in the transformed data. Finally, for each cell, we computed the Spearman rank correlation between cells prior to, and after, transformation to measure deviations from a monotonic transformation. These three metrics allow for quantifying the trends observed in the three plots and offer a measure of the effectiveness of each normalization technique.

To verify that these metrics are reasonable for benchmarking normalization methods, we first examined cells from a NIH/3T3 mouse cell line dataset published in (Svensson 2020) and studied in (Ahlmann-Eltze and Huber 2021). We found that these metrics, which we computed for each normalization technique, were concordant with the analysis performed in (Ahlmann-Eltze and Huber 2021), and provided useful summaries of the performance of different normalization techniques (Supp. Fig. 2).

Benchmarks of 526 datasets

In recognition of the fact that the patterns we observed in *10xv3_NIH_3T3* were not necessarily representative of other datasets, we analyzed a further 525 datasets of which 437 passed quality control (Supp. Fig. 3.1 - 3.526, Methods). We evaluated eight normalization techniques; in addition to sctransform, we selected seven other methods based on their use in popular single-cell RNA-seq analysis packages, as well as a novel method we decided to investigate after examining initial results (see Methods). The most widely used approach for depth normalization and variance stabilization is depth normalization of cell counts to ten thousand counts (CP10k), followed by variance stabilization of the gene counts with the log(x+1) transform (denoted by log1p, with the combined procedures denoted log1pCP10k). This is the default in the Seurat and Scanpy packages. Seurat and Scanpy also recommend an additional scaling step (scalelog1pCP10k) for some analyses. Scaling consists of two steps: centering gene expression values by subtracting the mean expression of each gene, and equalizing gene variances by dividing the counts for each gene by the standard deviation (computed across cells). We also benchmarked a method that has been adapted for single-cell RNA-seq from bulk RNA-seq, namely cell depth normalization to the mean cell depth, followed by log1p (log1pPF). This “proportional fitting” approach, our name for the method because the first step constitutes one step of iterative proportional fitting (Edwards Deming and Stephan 1940), is similar to log1pCP10k (Love, Huber, and Anders 2014), and is the method underlying the Monocle single-cell analysis package (Cao et al. 2019). We also tested the square root transformation that forms a part of the scprep package default transformation[2], as well as a log1pCPM, which is a popular option in Seurat, and is similar to log1pCP10k but with a scaling factor of one million rather than ten thousand. Finally, we included a benchmark of PF for completeness.

[2] The scprep package is unpublished but is documented here: https://github.com/KrishnaswamyLab/scprep

Our benchmarks revealed high variability in the extent of variance stabilization for any given method (Fig. 2a); to the extent that even though one method might be better at stabilizing variance than another, on one dataset it may produce worse results than the inferior method on another. For example, the sqrt transformation results in a CV of 1.61 for *GSM3738540* (Supp. Fig. 3.400.2) whereas the log1p transformation yields a higher CV of 6.78 for *GSM3396184* (Supp. Fig. 3.243.2). Some datasets are also particularly sensitive to the method used. The sqrt transformation gives a CV of 9.45 for *GSM3178783* (Supp. Fig. 3.178.2) and 46.53 for *GSM3396177* (Supp. Fig. 3.236.2), whereas the log1p transformation gives consistent results for these datasets with 5.77 *GSM3178783* and 5.8 for *GSM3396177*; interestingly there is even a slight reversal in behavior. This highlights the importance of large-scale benchmarking for evaluating normalization methods.

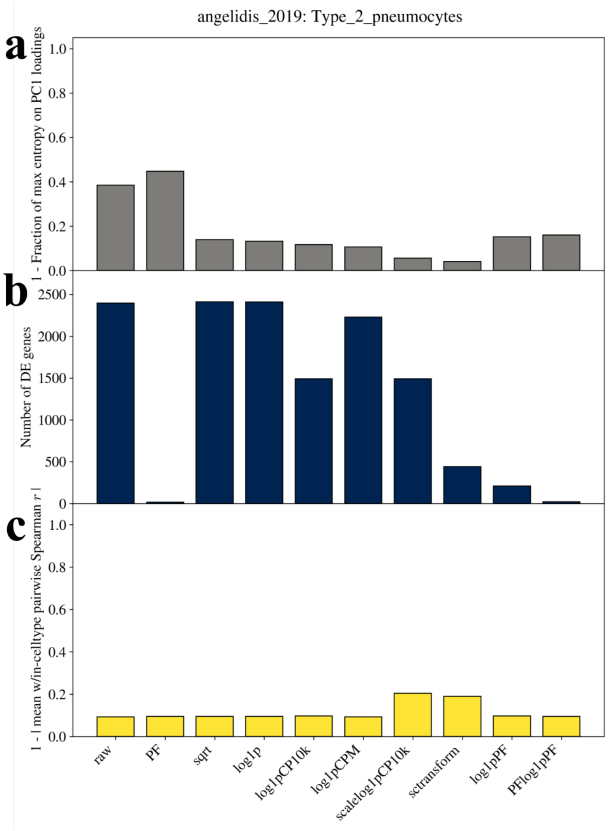

Figure 3: Cell-type level metrics. Three metrics are computed for cells within the *Type 2 pneumocytes* from *angelidis_2019* for all normalization methods. (a) The fraction entropy of the PC1 loadings for all genes as a fraction of the max entropy. (b) The number of false-positive DE genes. (c) The absolute value of the mean within-cell-type-pairwise Spearman r.

The sctransform method subsets the genes analyzed (see Methods), so to compare sctransform to other methods we redid the analysis of each method with respect to the sctransform selected genes (Fig. 2d); we found the results to be qualitatively consistent with the full analysis using all genes. In terms of depth normalization, we found that even methods that claim to normalize for depth, e.g. sctransform, do not succeed in completely removing depth effects and retain

information about depth in the normalized data (Fig. 2e). Popular normalization methods such as log1pCP10k are similar in terms of removing the effects of depth on downstream analysis (Fig. 2b, 2e). The sctransform normalized cells, for example, exhibit similar cell-depth correlation (r^2 = 0.37) as log1pPF cells (r^2 = 0.43) on average, with some sctransform normalized datasets exhibiting very high depth correlation (Supp. Fig. 3.520.2). Finally, while most transformations are monotonic, we find that sctransform scrambles the rank order of genes in individual cells (Fig. 2c, f), a straightforward result of the normalization procedure that can negatively influence downstream analyses if not taken into consideration.

Variance stabilization

Our analysis of current normalization methods shows that they exhibit a stark tradeoff between variance stabilization and depth normalization. To understand the implications of each normalization technique we analyzed data from (Angelidis et al. 2019), as studied in (Ahlmann-Eltze and Huber 2021).

Interestingly, there has been much focus on variance stabilization, perhaps because variance stabilization has a long history dating back to (Bartlett 1936). A relationship between expression levels of a gene and its variance can mask biological variation and affect data analysis methods such as PCA as a result of technical artifacts (e.g. sampling). Highly expressed genes may dominate PCA components, regardless of biologically meaningful variation. When analyzing *angelidis_2019* we found that PF, like the raw counts, was not variance stabilized resulting in non-uniform PC loadings corresponding to low entropy for genes (Fig. 3a), with PC loadings increasing with increasing gene mean (Supp. Fig. 4). sctransform had the highest entropy, a finding that can be explained by the heuristic clipping procedure performed on the gene variances (Choudhary and Satija 2022).

To address this problem, approximations to variance stabilizing transforms, such as log1p or sqrt are used, often in conjunction with a depth normalization step such as PF, CP10k, and CPM. The effect of each of these transformations on the mean-variance relationship can be seen in Supp. Fig. 3.519.2.[3] Variance stabilizing transforms like log1p and sqrt reduce the CV of the genes from *angelidis_2019* by a factor of 29.1 and 7.9 respectively (from 98.8 to 3.4 and 12.5). The addition of depth normalization step does not greatly affect the CV for log1pPF (3.0). Therefore normalization techniques that include a variance stabilization step will greatly reduce the effects that highly expressed, and thus highly variable, genes have on PC components.

The log transformation is often used with a pseudocount, and the size of the pseudocount can be seen to reflect assumptions about the extent of overdispersion (Ahlmann-Eltze and Huber 2021; Booeshaghi and Pachter 2021). For negative binomial data, the overdispersion is the constant α in a quadratic mean (μ) - variance (σ^2) relationship of $\sigma^2 = \mu + \alpha\mu^2$. Depth normalizations prior to logarithmic transformation with a pseudocount of 1 therefore reflect assumptions about the overdispersion as reflected in the size factor. As pointed out in

[3] Different plotting styles can lead to very different interpretations of the effect of each normalization procedure on the mean-variance relationship (Supp. Fig. 5).

(Ahlmann-Eltze and Huber 2021), a large size factor represents an assumption of high overdispersion. For example, (Ahlmann-Eltze and Huber 2021) show that scaling counts with a size factor of one million by computing CPM in a scRNAseq dataset with an average of 5,000 counts per cell, is equivalent to using a pseudo-count of 0.005.[4] This amounts to assuming an overdispersion of $\alpha = 50$. This calculation is based on a variance stabilizing approximation derived by the Delta method that yields a pseudocount of $1/4\alpha$; interestingly, there is some disagreement over the denominator of the pseudocount as $1/2\alpha$ *(Anscombe 1948)* is frequently preferred. In a simulation study (see Methods), we found that in the range of relevant overdispersion parameters, $1/4\alpha$ provides a slightly better variance stabilizing transform than $1/2\alpha$ (Supp. Fig 6).

Our results (Fig. 3) reflect the different assumptions about overdispersion underlying the use of log1pPF, log1pCP10k, or log1pCPM depth-normalization, however they also show that a smaller CV is not necessarily indicative of better variance stabilization. For example, the CPM assumption of overdispersion, that is at least two orders of magnitude larger than present in biological datasets, results in overcorrection and the removal of biological variation (Ahlmann-Eltze and Huber 2021) and results in the smallest CV in *angelidis_2019* of 1.7. The sqrt transformation did not perform as well at stabilizing the variance as log1p, which is not surprising given the overdispersion (relative to the Poisson distribution) of single-cell RNA-seq data. As noted previously, many methods display a linear relationship between gene mean and gene variance for cells with very low counts. This phenomenon is well known and is a consequence of Theorem 1 of (Warton 2018). The sctransform method is an exception, because when the program computes the Pearson residuals, the standard deviation for each gene is artificially required to be at least *nzmedian*/5 where *nzmedian* is the median number of counts for each gene computed over non-zero cells. (Choudhary and Satija 2022).

The log1pPF method, which stabilizes the variance after depth normalization, performs well in all metrics, a result which is consistent with the findings of (Ahlmann-Eltze and Huber 2021). Overall, while some methods achieve better variance stabilization than others since they better match the overdispersion characteristics of biological data (e.g. log1p vs. sqrt), even sqrt is effective at achieving an absolute reduction in the coefficient of variation of variance, which explains its adequacy in scprep. Similarly, while log1pCP10k is preferable to log1pCPM, the use of log1pCPM does not preclude obtaining some meaningful results in analysis (Chen et al. 2021). Indeed, all current variance stabilization procedures are heuristics that ignore the fact that Poisson and negative binomial distributions may arise due to biophysical stochasticity in bursty transcription and RNA degradation (Amrhein, Harsha, and Fuchs 2019; Jahnke and Huisinga 2007). The development of “mechanistically justified normalization” is a pressing challenge for single-cell RNA-seq analysis.

[4] (Ahlmann-Eltze and Huber 2021) made a minor error when performing this calculation and erroneously reported an overdispersion of 250.

Implicit use of cell depth

Current depth normalization procedures that are applied alongside variance stabilization procedures implicitly assume that differences in cell-count depth is a technical artifact due to sampling differences between cells, rather than the result of different numbers of RNA molecules in different cells. While this assumption may be flawed, in the absence of effective data and procedures for assessing variation in the amount of RNA between cells, normalization for cell sequencing depth is essential. This is because standard statistical tests that are employed in Seurat and Scanpy, such as the t-test and wilcoxon rank-sum, do not explicitly model cell-depth as a technical covariate.

To investigate the effect that poor depth-normalization can have on analysis, we selected two subsets of cells from *Type 2 pneumocytes* with high and low depth respectively (see Methods). An analysis of the number of differential expressed genes detected after transformation with different methods shows that poor depth normalization can lead to many false positives (Fig. 3b). Interestingly, the default normalization used for differential expression analysis in Seurat and Scanpy (CP10k) finds 1,490 false-positive DE genes, about seven times more than log1pPF (Love, Huber, and Anders 2014), which is used in (Cao et al. 2019), and has been recently recommended again (Ahlmann-Eltze and Huber 2021). In comparison, sctransform finds 442 DE genes, about twice as many as log1pPF and about three times fewer than log1pCP10k.[5]

Depth normalization is also important for identifying clusters with biologically meaningful gene-expression patterns. Standard clustering techniques first construct a cell-cell distance matrix based on a distance metric. Next, a k-nearest neighbor graph is constructed from the distance matrix with tools such as *annoy* (Bernhardsson 2018). Finally, graph-partition methods, e.g. Louvain (Blondel et al. 2008) or Leiden (Traag, Waltman, and van Eck 2019), identify "communities" of cells in this graph that exhibit similar expression patterns.[6] In the absence of proper depth normalization, cell-cell distances, computed with metrics like the l_1 distance can be correlated with cell depth (Supp. Fig. 7a). For *Type 2 Pneumocytes* in *angelidis_2019*, the cell-cell distances were correlated with cell depth when normalized with sctransform (0.71) and scalelog1pCP10k (0.54) but not with PFlog1pPF (0.06). Proper depth normalization ensures that the k-nearest neighbor graph can be built with a distance metric that is cell-depth independent and results in cell communities that exhibit similar gene expression patterns.

The interpretation of PCA requires confidence that explained variation is biological rather than technical (Lun 2018) and in the absence of depth normalization, PCA components can correlate strongly with cell depth (Supp. Fig. 7b). Normalization techniques that do not include a final depth-normalization step, like sqrt, log1p, and log1pCP10k, demonstrate a high correlation with

[5] The use of Pearson residuals for differential expression has been both recommended (example 1, 2019; example 2, 2022) and discouraged (example, 2021) by the authors of sctransform.
[6] The Scanpy workflow runs Louvain and Leiden clustering with the same neighborhood graph as UMAP. This is not the case for Seurat's RunUmap (used for UMAP) and FindNeighbors (used for clustering) which utilize different default distance metrics (cosine vs. Euclidean distance) and different numbers of neighbors (30 vs. 20) respectively.

PC1. Techniques that end with a depth normalization step like PF and PFlog1pPF, and techniques that model cell depth as a covariate, exhibit lower correlation with PC1 with the former exhibiting almost no relation to PC1. In the absence of depth normalization, subsequent analyses that rely on PCA, such as clustering or UMAP, may produce results that are affected by technical artifacts, rather than reflecting biological structure (Fig. 1a).

Finding markers versus differential expression

Classic variance stabilizing transformations such as the logarithm or square root functions are monotonic, a property that is rarely highlighted, but of crucial importance. For instance, monotonicity of the transformation applied to single-cell RNA-seq counts is crucial for the task of finding marker genes. The term “marker gene identification” is frequently used interchangeably with “differential expression” (Dumitrascu et al. 2021), but the two tasks are not the same. Differential expression, which is the identification of genes exhibiting significantly different expression between groups of cells, is a needed step for marker gene identification, however the latter demands more: good marker genes for a group of cells are not only statistically differential with respect to other cells, but also specifically expressed (i.e. not present in high abundance in other cells).

One popular approach for finding marker genes is manual inspection of heatmaps, because in principle these can allow for identifying genes that not only distinguish among cell types, but that are also exceptionally highly expressed within cell types (Bonnycastle et al. 2020). The accurate depiction of gene expression in heatmaps is challenging due to the wide range of gene expression in typical experiments. To address this problem, programs such as Seurat and Scanpy scale the gene expression values across cells, by normalizing them to have mean zero and variance 1, and then clip extreme values. These values are visualized using a continuous color scale.

The use of heatmaps requires some care, because the relative expression of two marker genes within a cell type becomes meaningless as each gene is centered and scaled, effectively scrambling gene expression within each cell (Supp. Fig. 8). For example in the *angelidis_2019* dataset, *Syce2* is a DE gene for *Red Blood Cells* that switches ranking within the *Eosinophils* cell type, from rank 76 to rank 41 out of 96 top DE genes, after the heatmap scaling procedure (Methods). This scaling procedure, coupled with the lack of monotonicity of the sctransform or scalelog1pCP10k transformations of Seurat or Scanpy can make finding marker genes from heatmap visualizations challenging.

Use of monotonic transformations for normalizations results in cells within a cell type exhibiting higher pairwise Spearman r on their gene expression; a feature of monotonic transformations such as log1pPF but not non-monotonic transformations such as sctransform (Fig. 3c, Supp. Fig. 9). By avoiding an initial scrambling of genes within cells, further heatmap scaling procedures can then be applied to create two heatmaps (Supp Fig. 10) that more faithfully represent gene expression ranking. The first heatmap scales values across genes and the second across cells.

Scalable normalization

Compute resource constraints imposes practical limits on matrix operations (Lun 2020, n.d.). One issue that arises in the context of normalization is that some methods transform sparse matrices into dense matrices that can surpass standard RAM availability. For example, we found that the scalelog1pCP10k matrix *ERX2756720* was 219 times larger than the log1p sparse matrix. Memory and speed requirements (Supp. Fig. 11) can inhibit scalable computation on increasingly large scRNA-seq datasets and drive higher cloud-computing costs (Supplementary Table 1 of (Melsted et al. 2021))). In contrast, sparse matrices have been used for high-performance computing for a long time (Orchard-Eays 1956; Markowitz 1957), and can drastically reduce the memory overhead required to perform memory-intensive computations. While recently developed “sketching” procedures (Hao et al. 2022) that subsample matrix operations for scalable computation may provide workarounds for dense matrices, we believe that sparsity will remain an important consideration for normalization transformations for the foreseeable future.

The PFlog1pPF heuristic

The Seurat and Scanpy workflows offer users the ability to choose different matrix types for different analysis tasks. This is a good design decision, in principle, because different tasks make different assumptions on the count matrix. However, without clear guidelines or appropriate defaults, matrix managers like the Seurat and AnnData objects can confuse users and make analysis error-prone. A single normalization technique resulting in a single (sparse) matrix can make data sharing and reproducibility more straightforward.

While depth normalization is achieved perfectly with proportional fitting (PF), the addition of a log1p transform in log1pPF does reintroduce some depth heterogeneity (Fig. 2b). The importance of depth normalization therefore motivated us to explore adding an additional proportional fitting step to log1pPF. We hypothesized that an additional round of proportional fitting might achieve depth equalization without drastically affecting variance stabilization. We tested this method (PFlog1pPF) and found that to be the case on *10xv3_nih_3T3* (Supp. Fig. 3.2.2), *angelidis_2019* Supp. Fig. 3.519.2, and the other benchmark datasets (Supp. Fig. 3).

We observed that PFlog1pPF (Supp. Fig. 12) can be seen to only slightly decrease variance stabilization (Fig. 2a) while ensuring depth normalization and monotonicity. With the addition of a PF step, gene variance CV suffers only slightly making PFlog1pPF comparable to sqrt with the additional benefit of full depth normalization of PF resulting in almost no false-positive differentially expressed genes (Fig. 3b). PFlog1pPF also recapitulates cell-type marker gene expression for *angelidis_2019* and is consistent with other normalization techniques tested in (Ahlmann-Eltze and Huber 2021) (Supp. Fig. 13). Additionally, PCA components computed on PFlog1pPF have similar loadings to log1pPF (Fig 3a) and within-celltype pairwise gene expression rankings are better preserved than sctransform and scalelog1pCP10k (Fig. 3c) both of which exhibit high concordance (Supp. Fig. 14).

Discussion

Count normalization is a crucial first step in all scRNAseq analysis that, in principle, comprises a single step in a standard workflow. However in practice normalization is a collection of techniques, data representations, analysis types, and visualizations that interact with each other in non-obvious and frequently undocumented ways. In Seurat and Scanpy, the analysis software used for the majority of scRNAseq analysis, some normalization implementations can also limit users by requiring large amounts of memory. Thus, while users frequently think of normalization as a single data transformation step in analyses, it is often not; the software engineering choices made by developers of the tools used can affect analyses in unpredictable, and sometimes unintended ways.

Despite the complexity of normalization in practice, much work on scRNAseq has focused on statistical details that, while important, are not necessarily the primary determinants of results. For example, the debate over whether gene-specific over-dispersion parameters should be used when computing Pearson residuals (Hafemeister and Satija 2019; Lause, Berens, and Kobak 2021; Hafemeister and Satija 2020; Choudhary and Satija 2022) ignores the fact that Pearson residuals are not the result of a monotonic transformation, and they create dense matrices that can lead to significant analysis limitations (Borella et al. 2021). These problems have significant implications for common tasks such as finding marker genes, as discussed above. Newer methods that explicitly couple statistical methods with software engineering considerations are needed; we examined several recent publications proposing new ideas but restricted the paper to widely used methods common in existing workflows (Brown et al. 2021; Breda, Zavolan, and van Nimwegen 2021; Borella et al. 2021; Bacher et al. 2017). A detailed analysis and review of these methods is an important next step. Furthermore, normalization should ideally include modeling of transcriptional dynamics so as to be able to evaluate the contribution of technical noise to count data (Gorin and Pachter 2021).

We have argued that a single, sparse, variance-stabilized and depth-normalized matrix on which all analysis and visualizations are performed can simplify current workflows. The PFlog1PF heuristic we have proposed is a monotonic transform on the raw counts that results in a fully depth normalized matrix and offers variance stability similar to sqrt. Importantly, we have shown that for downstream analysis, PFlog1pPF effectively stabilizes variance for PCA, produces low false-positive DE genes, and has the same within cell-type Spearman correlation as unnormalized matrices. Having said that, we believe it is an interesting challenge to develop more principled approaches that achieve depth normalization and variance stabilization while preserving sparsity and respecting monotonicity

Regardless of the normalization transformation that is applied, our work shows that assessment of data quality and normalization effectiveness is crucial in practice. Measures such as the overdispersion, coefficient of variation of the transformed-gene variances, and raw to transformed cell-depth Pearson correlation ought to be collected as part of standard quality control of experiments. It's also crucial that practitioners understand the assumptions implicit in the normalizations applied, and the implications for interpretation of results, such as whether variation is technical or biological.

Methods

Preprocessing
Raw matrices were filtered by removing cells beneath a selected knee-plot threshold. The knee plot and threshold used for each dataset are reported in the dataset folders. Datasets for which the average count per cell was less than 818.46 (the average count per cell in *angelidis_2019*) were not used in Fig. 1.

Collecting metadata
Dataset metadata was collected with the ffq program version 0.2.1 available at https://github.com/pachterlab/ffq by running `ffq -l 2 -o DATASETID_metadata.json DATASETID`. 18 out of the 526 datasets processed did not have metadata associated with their dataset ID.

Normalizing matrices
We applied seven normalization methods to the cell-filtered matrix: PF, sqrt, log1p, log1pCP10k, log1pPF, log1pCPM, PFlog1pPF. The normalization transformations were computed by running the `norm_sparse.sh` script.

We then ran `norm_sctransform.sh` on the original cell-filtered matrix to generate the sctransform matrix. The sctransform function was called with var_features_n=number_of_genes_in_dataset, vst_flavor="v2", and default parameters. In order to perform a uniform analysis, we filtered the original cell-filtered matrix to the set of genes returned by sctransform- since sctransform has a built-in gene filtering step.

We then ran `norm_sparse.sh` to create the seven normalized matrices mentioned above, and finally ran `norm_cp10k_log_scale.sh` to create the scalelog1pCP10k matrix.

Running sctransform
We performed all of our benchmarks of sctransform with v2. The sctransform v1 regression model has been shown to be overspecified (Lause, Berens, and Kobak 2021) and has been superseded by v2. We opted to benchmark sctransform v2 over analytical Pearson residuals as the latter's validation consisted of comparing two dimensional PCA and UMAP embeddings to compare and contrast methods.

In order to run sctransform v2, a one-line modification was made to pysctransform.py (in the develop branch), namely casting `params[“order”]` as a numpy array with `npy.asarray(params["order"])` in line 333. This modification fixed an issue described in https://github.com/saketkc/pySCTransform/issues/4#issue-912930103 which was causing the pip-installed version of pySCTransform not to work. An additional modification to the pySCTransform code allowed for the corrected counts matrix to be returned- line 759 `return (vst_out["residuals"], vst_out["corrected_counts"])`.

Computing dataset metrics

For each normalization method we computed three metrics: the coefficient of variation on the transformed-gene variances (CV), the Pearson r^2 correlation between the transformed-cell counts and the raw cell counts, and the average Spearman r between the transformed-cell counts and the raw cell counts. The CV was computed by calculating the variance for each gene, across all cells, and then calculating the variance and mean across all genes, and dividing the two. The Pearson r^2 was computed by summing the transformed cell counts and running sklearn.linear_model.LinearRegression().fit() followed by score() on the transformed cell counts and the raw cell counts. The average Spearman r was computed by first performing paired stats.spearmanr on all transformed-raw cell pairs and then taking the mean.

Computing cell-type metrics

Cell-type metrics were computed on cells from the *Type 2 pneumocytes* in the *angelidis_2019* dataset. For each normalization method, `sklearn.decomposition.PCA()` was run with n_components=1 and svd_solver="full" and the absolute value of the loadings were l_1-normalized. The entropy was computed with `scipy.stats.entropy()` and the max entropy was computed with `np.log(ngenes)`. Additionally, the Pearson r2 was computed on PC1, derived from PCA on the normalized matrix, and raw-cell depth.

To compute the number of false-positive DE gene genes, we performed differential expression on two groups of cells: 500 cells with the highest raw-cell count and 500 cells with the lowest raw-cell count. Then, for each normalization method, we performed differential expression as previously described (Booeshaghi et al. 2021). The number of differentially expressed genes with a corrected p-value less than 0.01 were recorded.

To compute the average pairwise-Spearman gene-rank correlation, we first found the smallest non-zero difference in counts between entries in each normalization matrix. We added a random number between zero and one-fourth of this minimum to each gene vector to break ties. After adjusting the matrix counts, pairwise-Spearman correlations were calculated on all cells and the average was computed.

To compute the correlation between pairwise-difference in cell depth and pairwise l_1 distance, for each matrix we subsampled to 1,000 cells and then computed all pairwise differences in cell depth by running `sklearn.metrics.pairwise_distances` with metric="l1" on the cell sums. Then we computed the pairwise l_1 distances in the same manner but on with the entire gene vectors. Lastly, `sklearn.linear_model.LinearRegression.fit()` and `score()` were used to compute the Pearson correlation.

Computing matrix metrics

The following matrix-level metrics were computed for each matrix, on both all genes and those subset by sctransform: the number of cells (ncells), the number of genes (ngenes), the number of non-zero entries in the matrix (nvals), the fraction of non-zero entries (density), the average depth per cell (avg_per_cell), the average depth per gene (avg_per_gene), the minimum depth per cell (min_cell), the maximum depth per cell (max_cell), the total number of counts in the

matrix (total_count), the empirical overdispersion (overdispersion). These metrics were computed with `metrics_matrix.sh`.

Creating multi-panel normalization figure

For each dataset and normalization, the following three plots were made: 1. A scatterplot of the transformed gene variance vs raw gene mean, 2. a scatterplot of the transformed cell depth vs raw cell depth, and 3. a histogram of the distribution of transformed-to-raw cell Spearman rank correlations. To make visualization easier, a min-max procedure was performed to scale the x and y axes of plot 2 where the min cell depth was subtracted from each cell and the result was divided by the max cell depth. These figures were made on all genes, for normalizations that were computationally tractable, and on the gene subset by sctransform for all normalizations.

Plotting styles for the gene mean-variance relationship

In order to consistently visualize variance stabilization of normalization procedures against each other, we plotted all transformations on a log-log axis with the x and y-axis limits set equal. We also plotted the identity line y equals x to illustrate the asymptotic behavior of the mean-variance relationship for genes with small mean.

Pseudocount simulation

We simulated negative binomial count data for 8,000 genes, $g_1, g_2, \ldots, g_{8000}$, as follows: we first drew the mean expression for each gene from an exponential distribution with mean 3, obtaining $\mu_1, \mu_2, \ldots, \mu_{8000}$. We considered the overdispersion parameters $\gamma = 0.3, 0.5, 1, 1.5, 2, 3, 4, 5$. This spans a larger range than is evident in typical single-cell RNA-seq experiments, but is informative. For each gene we generated gene counts for 10,000 cells from a negative binomial distribution with mean μ_i and overdispersion γ_k to form a 10,000 cells x 8,000 genes count matrix. We then filtered this data to remove genes with average count less than 4. Two hundred simulations were performed for each parameter setting.

Generating heatmaps

The top 100 expressed genes were found for each cell type in *angelidis_2019*. Then a *cell type x gene* matrix was made by averaging the expression of all cells within a cell type on the set of top 100 genes for that cell type. Lastly, the genes within each cell type were ranked from lowest to highest expressed using `scipy.stats.rankdata()` and the matrix of ranks was plotted on a heatmap.

To create the cell and gene-scaled *cell x gene* heatmaps, the top 96 DE genes for all cell types were selected and the *cell x gene* matrix on those 96 genes was scaled to unit variance and zero mean using `sklearn.preprocessing.scale()` across the cells to create the gene-scaled heatmap, and across the genes to create the cell-scaled heatmap. To find genes that switch rank, we first rank the raw gene expression within a cell type for the top marker genes, and then compare gene ranks to scaled (mean zero and variance one) gene expression ranks.

Data and code availability

All data and code to reproduce the figures and results in the paper are available at https://github.com/pachterlab/BHGP_2022.

Acknowledgements

This project started with an investigation of normalization of orthogonal barcoding tags. We thank John Thompson and Linda Hsieh-Wilson for helpful initial discussions related to that problem. We thank Tara Chari for helpful insights on normalization and clustering. Lambda Moses helped with reviewing the Seurat source code.

Author contributions

A.S.B., and L.P. developed the project idea. A.G.M. pre-processed the datasets. A.S.B. performed the analysis. A.S.B. and A.G.M. compiled the supplementary material. A.S.B. drafted the paper. I.B.H. performed the overdispersion simulation. A.S.B., I.B.H., A.G.M., and L.P. wrote, reviewed, and edited the paper.

Competing interests

The authors declare no competing interests.

References

Ahlmann-Eltze, Constantin, and Wolfgang Huber. 2021. “Transformation and Preprocessing of Single-Cell RNA-Seq Data.” *bioRxiv*. https://doi.org/10.1101/2021.06.24.449781.

Amrhein, Lisa, Kumar Harsha, and Christiane Fuchs. 2019. “A Mechanistic Model for the Negative Binomial Distribution of Single-Cell mRNA Counts.” *bioRxiv*. https://doi.org/10.1101/657619.

Anders, Simon, and Wolfgang Huber. 2010. “Differential Expression Analysis for Sequence Count Data.” *Nature Precedings*, March, 1–1.

Angelidis, Ilias, Lukas M. Simon, Isis E. Fernandez, Maximilian Strunz, Christoph H. Mayr, Flavia R. Greiffo, George Tsitsiridis, et al. 2019. “An Atlas of the Aging Lung Mapped by Single Cell Transcriptomics and Deep Tissue Proteomics.” *Nature Communications* 10 (1): 963.

Anscombe, F. J. 1948. “The Transformation of Poisson, Binomial and Negative-Binomial Data.” *Biometrika* 35 (3/4): 246–54.

Bacher, Rhonda, Li-Fang Chu, Ning Leng, Audrey P. Gasch, James A. Thomson, Ron M. Stewart, Michael Newton, and Christina Kendziorski. 2017. “SCnorm: Robust Normalization of Single-Cell RNA-Seq Data.” *Nature Methods* 14 (6): 584–86.

Bartlett, M. S. 1936. “The Square Root Transformation in Analysis of Variance.” *Supplement to the Journal of the Royal Statistical Society* 3 (1): 68–78.

Bernhardsson, Erik. 2018. “Annoy: Approximate Nearest Neighbors in C++/Python.” https://pypi.org/project/annoy/.

Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. “Fast Unfolding of Communities in Large Networks.” *Journal of Statistical Mechanics* 2008 (10): P10008.

Bonnycastle, Lori L., Derek E. Gildea, Tingfen Yan, Narisu Narisu, Amy J. Swift, Tyra G. Wolfsberg, Michael R. Erdos, and Francis S. Collins. 2020. “Single-Cell Transcriptomics from Human Pancreatic Islets: Sample Preparation Matters.” *Biology Methods & Protocols* 5 (1): bpz019.

Booeshaghi, A. Sina, and Lior Pachter. 2021. “Normalization of Single-Cell RNA-Seq Counts by Log(x + 1)* or log(1 + X).” *Bioinformatics* , March. https://doi.org/10.1093/bioinformatics/btab085.

Booeshaghi, A. Sina, Zizhen Yao, Cindy van Velthoven, Kimberly Smith, Bosiljka Tasic, Hongkui Zeng, and Lior Pachter. 2021. “Isoform Cell-Type Specificity in the Mouse Primary Motor Cortex.” *Nature* 598 (7879): 195–99.

Borella, Matteo, Graziano Martello, Davide Risso, and Chiara Romualdi. 2021. “PsiNorm: A Scalable Normalization for Single-Cell RNA-Seq Data.” *Bioinformatics* , September. https://doi.org/10.1093/bioinformatics/btab641.

Breda, Jérémie, Mihaela Zavolan, and Erik van Nimwegen. 2021. “Bayesian Inference of Gene Expression States from Single-Cell RNA-Seq Data.” *Nature Biotechnology* 39 (8): 1008–16.

Brown, Jared, Zijian Ni, Chitrasen Mohanty, Rhonda Bacher, and Christina Kendziorski. 2021. “Normalization by Distributional Resampling of High Throughput Single-Cell RNA-Sequencing Data.” *Bioinformatics* , June. https://doi.org/10.1093/bioinformatics/btab450.

Cao, Junyue, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M. Ibrahim, Andrew J. Hill, Fan Zhang, et al. 2019. “The Single-Cell Transcriptional Landscape of Mammalian Organogenesis.” *Nature* 566 (7745): 496–502.

Chari, Tara, Joeyta Banerjee, and Lior Pachter. 2021. “The Specious Art of Single-Cell Genomics.” *bioRxiv*. https://doi.org/10.1101/2021.08.25.457696.

Chen, Wanqiu, Yongmei Zhao, Xin Chen, Zhaowei Yang, Xiaojiang Xu, Yingtao Bi, Vicky Chen,

et al. 2021. “A Multicenter Study Benchmarking Single-Cell RNA Sequencing Technologies Using Reference Samples.” *Nature Biotechnology* 39 (9): 1103–14.

Choudhary, Saket, and Rahul Satija. 2022. “Comparison and Evaluation of Statistical Error Models for scRNA-Seq.” *Genome Biology* 23 (1): 27.

Cole, Michael B., Davide Risso, Allon Wagner, David DeTomaso, John Ngai, Elizabeth Purdom, Sandrine Dudoit, and Nir Yosef. 2019. “Performance Assessment and Selection of Normalization Procedures for Single-Cell RNA-Seq.” *Cell Systems* 8 (4): 315–28.e8.

Crowell, Helena L., Charlotte Soneson, Pierre-Luc Germain, Daniela Calini, Ludovic Collin, Catarina Raposo, Dheeraj Malhotra, and Mark D. Robinson. 2020. “Muscat Detects Subpopulation-Specific State Transitions from Multi-Sample Multi-Condition Single-Cell Transcriptomics Data.” *Nature Communications* 11 (1): 6077.

Doob, J. L. 1935. “The Limiting Distributions of Certain Statistics.” *The Annals of Mathematical Statistics* 6 (3): 160–69.

Dumitrascu, Bianca, Soledad Villar, Dustin G. Mixon, and Barbara E. Engelhardt. 2021. “Optimal Marker Gene Selection for Cell Type Discrimination in Single Cell Analyses.” *Nature Communications* 12 (1): 1186.

Edwards Deming, W., and Frederick F. Stephan. 1940. “On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals Are Known.” *The Annals of Mathematical Statistics* 11 (4): 427–44.

Evans, Ciaran, Johanna Hardin, and Daniel M. Stoebel. 2018. “Selecting between-Sample RNA-Seq Normalization Methods from the Perspective of Their Assumptions.” *Briefings in Bioinformatics* 19 (5): 776–92.

Gorin, Gennady, and Lior Pachter. 2021. “Length Biases in Single-Cell RNA Sequencing of Pre-mRNA.” *bioRxiv*. https://doi.org/10.1101/2021.07.30.454514.

Hafemeister, Christoph, and Rahul Satija. 2019. “Normalization and Variance Stabilization of Single-Cell RNA-Seq Data Using Regularized Negative Binomial Regression.” *Genome Biology* 20 (1): 296.

———. 2020. “Analyzing scRNA-Seq Data with the Sctransform and Offset Models.” https://satijalab.org/pdf/sctransform_offset.pdf.

Hao, Yuhan, Tim Stuart, Madeline Kowalski, Saket Choudhary, Paul Hoffman, Austin Hartman, Avi Srivastava, et al. 2022. “Dictionary Learning for Integrative, Multimodal, and Scalable Single-Cell Analysis.” *bioRxiv*. https://doi.org/10.1101/2022.02.24.481684.

Jahnke, Tobias, and Wilhelm Huisinga. 2007. “Solving the Chemical Master Equation for Monomolecular Reaction Systems Analytically.” *Journal of Mathematical Biology* 54 (1): 1–26.

Lause, Jan, Philipp Berens, and Dmitry Kobak. 2021. “Analytic Pearson Residuals for Normalization of Single-Cell RNA-Seq UMI Data.” *Genome Biology* 22 (1): 258.

Love, Michael I., Wolfgang Huber, and Simon Anders. 2014. “Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2.” *Genome Biology* 15 (12): 550.

Lun, Aaron. 2018. “Overcoming Systematic Errors Caused by Log-Transformation of Normalized Single-Cell RNA Sequencing Data.” *bioRxiv*. https://doi.org/10.1101/404962.

———. 2020. “What Transformation Should We Use?” January 20, 2020. https://ltla.github.io/SingleCellThoughts/general/transformation.html.

———. n.d. *GitHub Issue Comment LTLA / scRNAseq - Aaron Lun on “Seurat Versions?”* Github. Accessed April 20, 2022. https://github.com/LTLA/scRNAseq/issues/15#issuecomment-650648478.

Lytal, Nicholas, Di Ran, and Lingling An. 2020. “Normalization Methods on Single-Cell RNA-Seq Data: An Empirical Survey.” *Frontiers in Genetics* 11 (February): 41.

Markowitz, Harry M. 1957. “The Elimination Form of the Inverse and Its Application to Linear Programming.” *Management Science* 3 (3): 255–69.

Melsted, Páll, A. Sina Booeshaghi, Lauren Liu, Fan Gao, Lambda Lu, Kyung Hoi Joseph Min,

Eduardo da Veiga Beltrame, Kristján Eldjárn Hjörleifsson, Jase Gehring, and Lior Pachter. 2021. “Modular, Efficient and Constant-Memory Single-Cell RNA-Seq Preprocessing.” *Nature Biotechnology* 39 (7): 813–18.

Nguyen, Lan Huong, and Susan Holmes. 2019. “Ten Quick Tips for Effective Dimensionality Reduction.” *PLoS Computational Biology* 15 (6): e1006907.

Orchard-Eays, Wm. 1956. “An Efficient Form of Inverse for Sparse Matrices.” In *Proceedings of the 1956 11th ACM National Meeting*, 154–57. ACM ’56. New York, NY, USA: Association for Computing Machinery.

Robinson, Mark D., and Alicia Oshlack. 2010. “A Scaling Normalization Method for Differential Expression Analysis of RNA-Seq Data.” *Genome Biology* 11 (3): R25.

Svensson, Valentine. 2020. “Droplet scRNA-Seq Is Not Zero-Inflated.” *Nature Biotechnology* 38 (2): 147–50.

Tian, Luyi, Xueyi Dong, Saskia Freytag, Kim-Anh Lê Cao, Shian Su, Abolfazl JalalAbadi, Daniela Amann-Zalcenstein, et al. 2019. “Benchmarking Single Cell RNA-Sequencing Analysis Pipelines Using Mixture Control Experiments.” *Nature Methods* 16 (6): 479–87.

Traag, V. A., L. Waltman, and N. J. van Eck. 2019. “From Louvain to Leiden: Guaranteeing Well-Connected Communities.” *Scientific Reports* 9 (1): 5233.

Vallejos, Catalina A., Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C. Marioni. 2017. “Normalizing Single-Cell RNA Sequencing Data: Challenges and Opportunities.” *Nature Methods* 14 (6): 565–71.

Warton, David I. 2018. “Why You Cannot Transform Your Way out of Trouble for Small Counts.” *Biometrics* 74 (1): 362–68.

Wolf, F. Alexander, Philipp Angerer, and Fabian J. Theis. 2018. “SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis.” *Genome Biology* 19 (1): 15.

You, Yue, Luyi Tian, Shian Su, Xueyi Dong, Jafar S. Jabbari, Peter F. Hickey, and Matthew E. Ritchie. 2021. “Benchmarking UMI-Based Single-Cell RNA-Seq Preprocessing Workflows.” *Genome Biology* 22 (1): 339.