Transformation and Preprocessing of Single-Cell RNA-Seq Data

Constantin Ahlmann-Eltze* and Wolfgang Huber*

*Genome Biology Unit, EMBL, Heidelberg, 69117, Germany.

June 24, 2021

Abstract

The count table, a numeric matrix of genes × cells, is a basic input data structure in the analysis of single-cell RNA-seq data. A common preprocessing step is to adjust the counts for variable sampling efficiency and to transform them so that the variance is similar across the dynamic range. These steps are intended to make subsequent application of generic statistical methods more palatable. Here, we describe three transformations (based on the delta method, model residuals, or inferred latent expression state) and compare their strengths and weaknesses. We conclude with an outlook on future needs for the development of transformations for single-cell count data.

Software: An R package implementing the delta method and residual-based variance-stabilizing transformations is available on github.com/const-ae/transformGamPoi.

Contact: constantin.ahlmann@embl.de

Single-cell RNA sequencing count tables are heteroskedastic, which means that counts for highly expressed genes vary more than for lowly expressed genes; accordingly, a change in a gene's counts from 0 to 100 between different cells is more relevant than, say, a change from 1,000 to 1,100. Analyzing heteroskedastic data is challenging because standard statistical methods typically perform best for data with uniform variance. Conversely, on heteroskedastic data, in general:

- generic statistical tests become unreliable,
- least sum of squares regression estimates are unbiased but imprecise, and their standard errors are wrong (Wooldridge, 2013),
- classification and clustering become less accurate.

In Fig. 1, we provide a schematic example. We show the probability mass functions of three Poisson distributions with different means. We see that the standard deviation for the blue distribution ($\mu = 64$) is four times larger than that of the red distribution ($\mu = 4$).

It is important to keep in mind that although a higher mean implies more variance, fold change estimates between two conditions are more precise the higher the involved mean parameters. Fig. 1B shows kernel-smoothed densities of the $\log_2$ fold changes between the counts from the red vs. green, and green vs. blue distributions. The latter is more precise because the coefficient of variation (that is, the standard deviation divided by the mean) of the Poisson distribution decreases with the mean (Appendix B.1).

Statistical approaches that explicitly model the sampling distribution of the data—a theoretically and empirically well-supported and widely used choice is the Gamma-Poisson distribution(Grün et al., 2014; Svensson, 2020; Kharchenko, 2021)—overcome the problem of heteroskedasticity, but the parameter inference of such models can be fiddly and computationally expensive (Townes, 2019; Ahlmann-Eltze and Huber, 2020). Instead, a popular choice is to use variance-stabilizing transformations as a preprocessing step, and subsequently to use the many existing statistical methods that, implicitly or explicitly, assume uniform variance for best performance (Amezquita et al., 2020; Kharchenko, 2021).

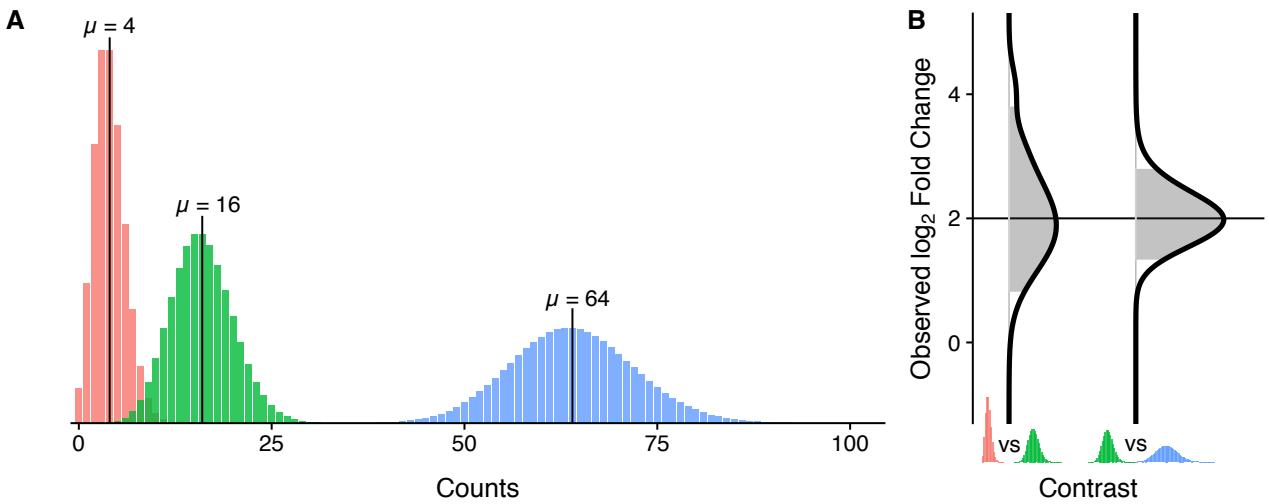


Figure 1: Example of heteroskedastic data. (A) shows the probability mass functions of three Poisson distributions with different means, such that the $\log_2$ fold change between the green and red means, as well as between the blue and green means, is 2. (B) shows the smoothed $\log_2$ fold changes between the red vs. the green, and the green vs. the blue distributions. The shaded areas show the range between the 5% and 95% quantiles.

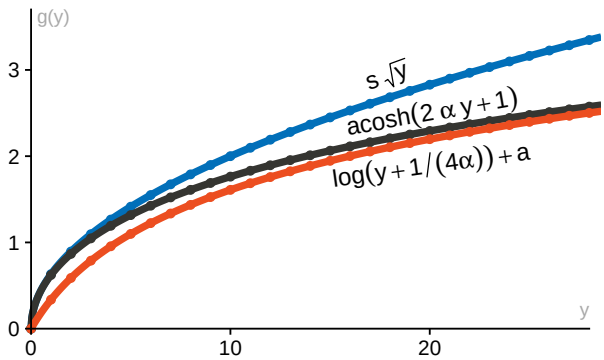


Figure 2: Graph of the three delta method-based variance-stabilizing transformations that are most relevant for count data. The curves are shown for overdispersion parameter $\alpha = 0.1$. We chose the offset $\mathrm{a} = \log(4\alpha)$ in the shifted logarithm and the scaling $\mathrm{s} = 2\sqrt{\alpha}$ in the square-root transformation to match the acosh transformation. The points highlight integer values on the abscissa.

Delta method

Variance-stabilizing transformations based on the delta method promise an easy fix for heteroskedasticity where the variance only depends on the mean. Instead of working with the raw counts Y, we apply a non-linear function $g(Y)$ designed to make the variances (and possibly, higher moments) more similar across the dynamic range (Bartlett, 1947).

The Gamma-Poisson distribution implies a quadratic mean-variance relation of $\mathbb{V}\mathrm{ar}[Y] = \mu + \alpha\mu^2$, where μ is the mean and α is the overdispersion (i.e., the additional variation compared to a Poisson distribution). Given this mean-variance relation, we can use the delta method (Dorfman, 1938) to find the variance-stabilizing transformation

$$g(y) = \frac{1}{\sqrt{\alpha}} \operatorname{acosh}(2\alpha y + 1)\,. \tag{1}$$

The shifted log transformation

$$g(y) = \log(y + c) \tag{2}$$

is a good approximation for Eq. (1) if the pseudo-count is $c = \frac{1}{4\alpha}$ (see Fig. 2 and Appendix B.2). The shifted log transformation is the most popular pre-processing method for single-cell data. However, it is conventionally used with pseudo-count $c = 1$ (Butler et al., 2018; Amezquita et al., 2020). Instead, we recommend either using a larger pseudo-count, as α is typically in the range of 0.01 to 0.16 (Suppl. Fig. S1), which implies a choice of c in the range of 25 to 1.6; or directly using the acosh-based transformation, since the approximation deteriorates for $\alpha \ll 0.01$.

One problem with variance-stabilizing transformations based on the delta method are the so-called *size factors*. These parameters, of which there is one per cell, adjust simultaneously for variable cell sizes and for variable efficiency with

which molecules are sampled from the pool of all mRNAs of a cell during the measurement process (Lun et al., 2016). To correct the data for varying size factors, conventionally, the counts are divided by suitably estimated size factors before the variance-stabilizing transformation is applied (Love et al., 2014; Amezquita et al., 2020). However, this operation does not completely remove the confounding effect of the size factors: e.g., in a low-dimensional embedding of the cells, the cells may still separate by size factor instead of possibly more interesting biological differences (Suppl. Fig. S2). Intuitively, the trouble stems from the fact that the division scales large counts from cells with large size factors and small counts from cells with small size factors to the same value, although small counts after scaling are more variable. In Appendix B.3, we explore the problem more formally.

A second problem with variance-stabilizing transformations based on the delta method is, as Warton (2018) points out, that transformations cannot reasonably be expected to stabilize the variance of small counts (compare with Suppl. Fig. S3).

Pearson residuals

Hafemeister and Satija (2019) suggested a different approach to variance stabilization, which promises to address the confounding effect of the size factors and effectively stabilize the variance also for small counts. They use Pearson residuals

$$r = \frac{y - \mu}{\sqrt{\mu + \alpha\mu^2}}, \tag{3}$$

where μ and α come from a Gamma-Poisson generalized linear model fit for each gene i

$$\begin{aligned} Y_{ij} &\sim \text{GammaPoisson}\left(\mu_{ij}, \alpha_i\right) \\ \log(\mu_{ij}) &= \beta_{i0} + \beta_{is}\log(s_j), \end{aligned} \tag{4}$$

where j is the cell index, β_{i0} is the intercept, s_j is the cell-specific size factor, and β_{is} is the corresponding size factor coefficient. Note that the denominator in Eq. (3) is the standard deviation of a Gamma-Poisson random variable with parameters μ and α.

The generalized linear model incorporates the size factors and removes their confounding effect (Suppl. Fig. S2). Furthermore, the transformation, using the gene-wise mean and standard deviation estimate, ensures that also the variances of lowly expressed genes are stabilized (Suppl. Fig. S3).

Although *sctransform* (the implementation of the Pearson residual method provided by Hafemeister and Satija (2019)) performed well in a recent benchmark (Germain et al., 2020), there has been a debate around its statistical model. Lause et al. (2021) argued that neither the estimation of β_{is} nor the estimation of one overdispersion per gene are necessary. Instead, Lause et al. (2021) suggested treating the log-size factors as offsets (i.e., fixing $\beta_{is} = 1$) and fixing the overdispersion to $\alpha = 0.01$, because that is roughly the overdispersion they observed in experiments where an RNA solution is homogeneously encapsulated in droplets. Hafemeister and Satija (2020) responded that estimating a gene-wise coefficient for the size factor "allows sctransform to adapt to artifacts and biases" and that fixing the overdispersion to a small value over-emphasizes the variation of highly abundant housekeeping genes.

Estimating a size factor coefficient per gene or treating it as fixed has little impact on the resulting residuals. Both Lause et al. (2021) and Hafemeister and Satija (2020) state that the resulting residuals are similar. We confirm that the question of how the overdispersion is chosen is more important. In Suppl. Fig. S4, we compare the effect of using the offset model or a fixed overdispersion against the sctransform model across six single-cell datasets. We find that the impact of using the offset model is negligible compared to the choice of the overdispersion.

So how should the overdispersion be estimated or fixed? It turns out that there is no unique correct, or universally optimal answer: it depends on the biological question that the analyst wants to ask. Lause et al. (2021) based their suggestion on the analysis of droplets all loaded from the same RNA solution. This can be considered a technical control experiment, and we confirm that $\alpha = 0.01$ describes the overdispersion for such data well (Suppl. Fig. S1A). However, a technical control experiment is not the only possible reference frame.

To complement the analysis of Lause et al. (2021), we analyzed the overdispersion found in cells from immortalized cell lines, which one can consider biological replicates (Suppl. Fig. S1B). The data from these cells show more overdispersion than that of the droplets with RNA solution,

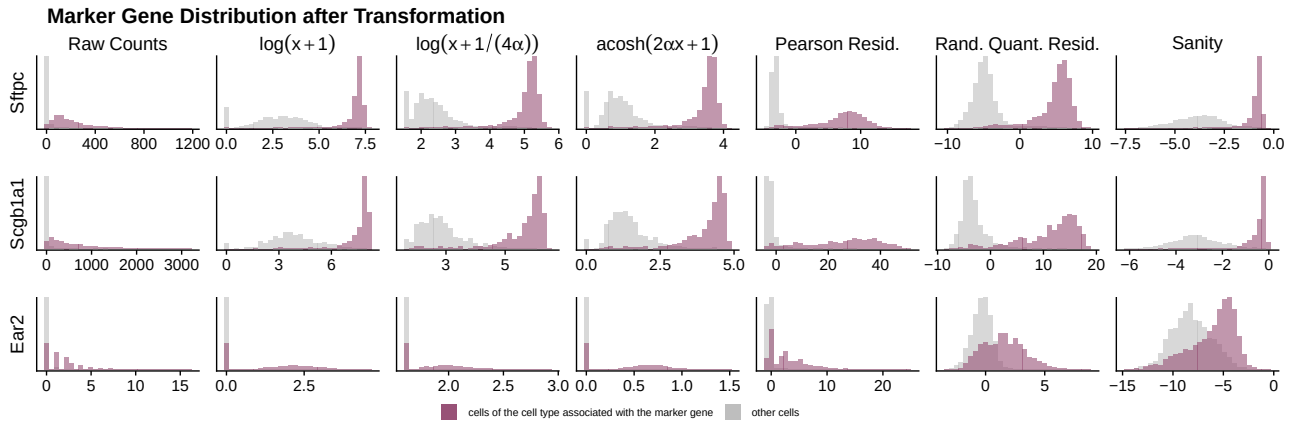


Figure 3: Histograms of the raw gene counts and the transformed values, for six different transformation approaches, for three cell-type marker genes in a mouse lung dataset (Angelidis et al., 2019). The color of the bars indicates whether a cell is of the cell type associated with the marker gene (Sftpc for type II pneumocytes, Scgb1a1 for club and goblet cells, Ear2 for alveolar macrophages). For visual clarity, we down-sampled the other cells (grey) to match the number of those from the marked cell type.

and the overdispersion differs from gene to gene. This is not surprising, as even in an ostensibly homogeneous cell population, there are real biological differences between cells, e.g., cell cycle stage.

For the analyst, the question remains how to set the overdispersion.

- If any variation larger than the one expected due to Poisson sampling is considered interesting, it is natural to fix the overdispersion to $\alpha = 0$ or, allowing for some slack, to a small value like $\alpha = 0.01$ as Lause et al. (2021) suggested.

- If the interest lies in genes whose variation is higher than that in the majority of genes of similar expression level, a robust approach is that of Hafemeister and Satija (2019), who fit a trend line through the mean-overdispersion relation.

- If one wants to level any gene-wise overdispersion differences, e.g., if the interest lies in expression patterns of genes across cells, irrespective of each gene's absolute variability, one could use the gene-wise maximum likelihood overdispersion estimates.

An important drawback of the Pearson residuals is that they fail to stabilize the variance if a gene's true expression strongly differs between cell subpopulations, as shown in Fig. 3. The figure shows the expression pattern of three cell type marker genes after applying different variance-stabilizing transformations. Unlike the delta method-based, non-linear variance-stabilizing transformations, the Pearson residuals fail to reduce the variance within the high-expression subpopulations, because the Pearson residuals are a linear transformation per gene (Eq. (3)). This means that while Pearson residuals successfully rescale the data from different genes relative to each other, heteroskedasticity in the data of a gene across cells remains and may obstruct tasks like clustering, mixture modelling or differential expression analysis.

An alternative is to combine the idea of delta method-based variance-stabilizing transformations with the generalized linear model-residual approach by using non-linear residuals. We suggest using, for example, randomized quantile residuals (Dunn and Smyth, 1996). (Suppl. Fig. S5 shows how they are constructed.) Same as Pearson residuals, randomized quantile residuals stabilize the variance for small counts (Suppl. Fig. S3), but also stabilize the variance for one gene across cells (Fig. 3).

Latent expression state

An alternative approach, which is not directly concerned with finding a variance stabilizing transformation, aims to infer the latent expression state for each cell and gene. This is the idea used in differential expression tools like *edgeR* and *DESeq2* (Robinson et al., 2009; Love et al., 2014). It was recently developed further by Breda et al. (2021), who suggest using it as a data trans-

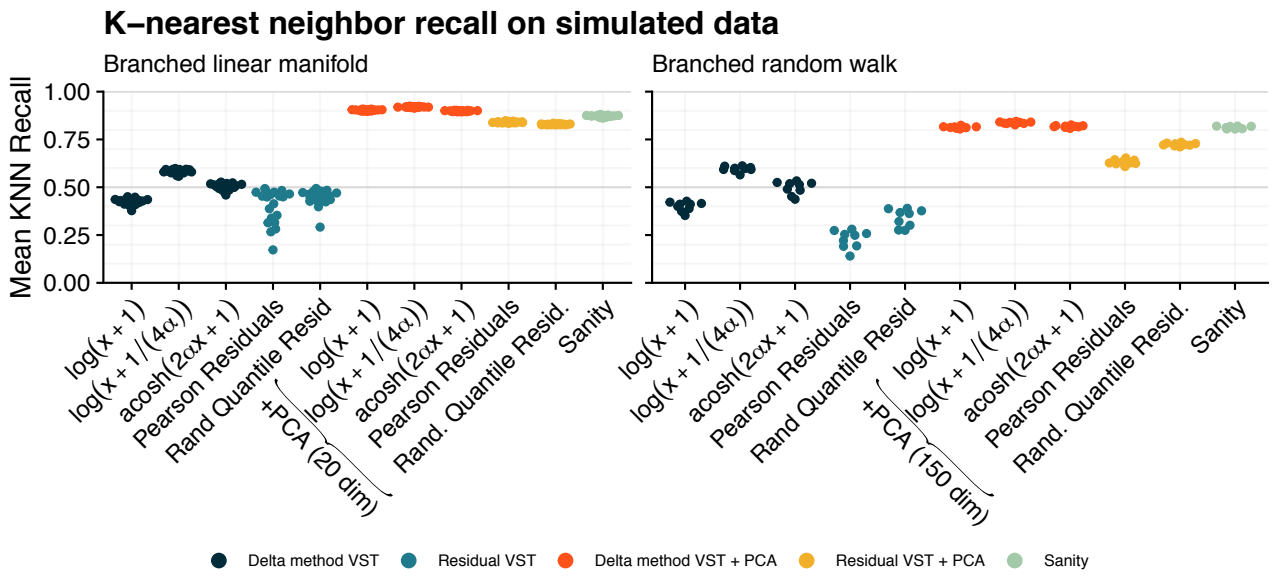


Figure 4: Bee swarm plot of the method performance for the eleven transformation methods discussed in this paper. We measure the performance as the percentage of correctly identified 100 nearest neighbors for each cell. We simulated two branching datasets: (1) each branch is a linear interpolation between two points, (2) each branch is a random walk (i.e., using the same kind of dataset as Breda et al. (2021)). On the linear manifold dataset, we used the first 20 principal components; on the random walk dataset, we used 150. The simulated overdispersion for the linear manifold dataset was $\alpha = 0.01$ and for the random walk $\alpha = 0$. For the transformations, we chose a mismatched overdispersion of $\alpha = 0.05$ for both datasets.

formation.

Breda et al. (2021) posit that each cell is characterized by a latent expression state, for which we observe a corrupted picture through the mRNA counts. To account for the uncertainty of the inferred expression state, they choose a Bayesian inference approach. Given a count matrix, their method *Sanity* infers a matrix of posterior distributions, which they represent using two matrices of real numbers: the distributions' means and standard deviations. The inferred posterior for a specific gene and cell is a function of the observed count, the cell's size factor, and the gene's overall expression across all cells. A larger size factor implies more precision in the inference of the latent state of that cell. The gene's expression pattern is used to regularize the inference. Breda et al. (2021) show that the regularization suppresses Poisson noise and that their method does a better job estimating the variance per gene on simulated data and data without biological signal.

Furthermore, Breda et al. (2021) include a benchmark that shows that Sanity is the best method for identifying the k nearest neighbors of a cell. However, we find that the delta method-based and residual-based variance-stabilizing transformations perform similarly well if we reduce the dimensions of the input data using principal component analysis (PCA) before searching for the k nearest neighbors (Fig. 4). The dimension reduction has the effect of averaging out uncorrelated noise, and serves a similar purpose as the regularization step of Sanity. However, unlike Sanity, the PCA-based approach requires the choice of the number of dimensions, which can greatly affect the performance (Suppl. Fig. S6).

A limitation of Sanity is that it is slow compared to the other transformations (Suppl. Fig. S7). In our analysis, Sanity needed $1,000 - 10,000\times$ more CPU time. The exact performance difference, of course, depends on the size of the dataset, the number of nearest neighbors, and the number of dimensions used in PCA, but in general, Sanity's k nearest neighbor search scales quadratically with the number of cells, because Sanity estimates all cell-by-cell distances. In contrast, the other transformations can be combined with approximate nearest neighbor search algorithms like random projection trees (Dasgupta and Freund, 2008), which scale linearly with the number of cells.

Discussion

We have described and compared three conceptually different preprocessing approaches for single-cell data. We find that the popular shifted log transformation in combination with principal component analysis performs well. We present theoretical evidence for using the related acosh transformation or using a larger pseudo-count $c = 1/(4\alpha)$ for the shifted logarithm. However, in the benchmark, we find only a slight performance benefit for the two alternatives.

The residual-based variance-stabilizing transformation approach first suggested by Hafemeister and Satija (2019) has nice theoretical properties. It stabilizes the variance across all genes and is not affected by variations of the size factor. However, the linear nature of the Pearson residuals-based transformation reduces its suitability for comparisons of the data of a gene across cells (such as differential expression analysis between cell subpopulations, or visualization)—there is no variance stabilization across cells, only across genes. As an alternative, we considered using non-linear residuals like randomized quantile residuals. However, in our benchmark, neither method excelled at identifying the k nearest neighbors.

The recent proposal by Breda et al. (2021) to use the inferred latent expression state as a transformation is appealing because it is biologically interpretable and does not need any tunable parameters. Sanity performs well at identifying the k nearest neighbors. It has two potential downsides: first, the fact that Sanity outputs not just one, but two values per gene and cell (mean and standard deviation) requires corresponding downstream processing, or conversely complicates feeding its output into generic methods that expect one number per gene and cell. Second, Sanity's inference approach is computationally expensive: in our applications, $1,000 - 10,000\times$ slower than the alternative transformation approaches.

The results of our analysis differ from previously reported results. Lause et al. (2021) benchmarked different gene selection and transformation approaches and claimed that the Pearson residuals-based transformation outperforms alternative approaches. They used a dataset with known cell types (Zheng et al., 2017) and added a synthetic rare cell type population by copying the expression data for 50 B cells and injecting 10 genes exclusively expressed in this population. The Pearson residuals-based gene selection and transformation successfully distinguished this synthetic B cell population from the real B cells. In contrast, the square root-based gene selection and transformation combination (i.e., the closest equivalent to our delta method-based transformations) failed to distinguish the synthetic from the real B cells because none of the 10 synthetic marker genes were among the 2,000 selected highly variable genes. Lause et al. (2021) compared the methods using the average F1-score across cell types, which is sensitive to poor performance in one cell type and thus shows a strong benefit to using Pearson residuals. However, in terms of accuracy (mean of correctly classified cells) or F1-score weighted by cell type size, the square root-based gene selection and transformation outperform the Pearson residuals.

The results from Lause et al. (2021) do not show that the Pearson residual variance-stabilizing transformation necessarily outperforms alternative transformations, but they stress the danger of prematurely removing important genes. In our benchmark, we avoided this problem by using all genes instead of selecting only highly variable ones. Of course, this increases the runtime of the PCA step, but that is rarely the computational bottleneck.

There has been considerable development in the space of preprocessing methods for single-cell RNA-seq data. Somewhat to our surprise, the shifted logarithm still performs among the best for preprocessing, but crucially only if combined with a dimensionality reduction method like PCA and an appropriate number of latent dimensions. Thus, in the future, we expect that new methods will shift away from simple variance stabilization to transformations that work well specifically in combination with PCA.

Ultimately, the approach of "preprocessing" (i.e., size-factor normalization and transformation) and subsequent application of generic statistical models has fundamental limitations, and we expect greater innovation from statistical models that integrate the biases and sampling phenomena in the measurement process with the biological effects (clusters, gradients, trajectories, differential expression, ...) of interest.

Availability

An R package that provides convenient methods for the delta method and residual-based variance-stabilizing transformations is available on github.com/const-ae/transformGamPoi. The code to generate the figures is available on github.com/const-ae/transformGamPoi-Paper. Our forked version of sctransform is available on branch offset_with_flexible_theta of github.com/const-ae/sctransform/. All datasets used in this manuscript are listed in Appendix C.

Acknowledgments

The authors thank Dr. Simon Anders for extensive discussions about variance-stabilizing transformations and how to benchmark preprocessing methods.

Funding

This work has been supported by the EMBL International PhD Programme and by the German Federal Ministry of Education and Research (CompLS project SIMONA under grant agreement no. 031L0263A).

References

10X Genomics (2017). Data from 10X Genomics website on PBMC cells. https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc4k.

10X Genomics (2018). Data from 10X Genomics website on HEK293T and NIH3T3 cells. https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/hgmm_5k_v3.

Ahlmann-Eltze, C. and Huber, W. (2020). glmGamPoi: Fitting gamma-Poisson generalized linear models on single cell count data. *Bioinformatics*.

Amezquita, R. A., Lun, A. T., Becht, E., Carey, V. J., Carpp, L. N., Geistlinger, L., Marini, F., Rue-Albrecht, K., Risso, D., Soneson, C., et al. (2020). Orchestrating single-cell analysis with Bioconductor. *Nature Methods*, 17(2):137–145.

Angelidis, I., Simon, L. M., Fernandez, I. E., Strunz, M., Mayr, C. H., Greiffo, F. R., Tsitsiridis, G., Ansari, M., Graf, E., Strom, T.-M., et al. (2019). An atlas of the aging lung mapped by single cell transcriptomics and deep tissue proteomics. *Nature Communications*, 10(1):1–17.

Baron, M., Veres, A., Wolock, S. L., Faust, A. L., Gaujoux, R., Vetere, A., Ryu, J. H., Wagner, B. K., Shen-Orr, S. S., Klein, A. M., et al. (2016). A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Systems*, 3(4):346–360.

Bartlett, M. S. (1947). The use of transformations. *Biometrics*, 3(1):39.

Breda, J., Zavolan, M., and van Nimwegen, E. (2021). Bayesian inference of gene expression states from single-cell RNA-seq data. *Nature Biotechnology*, pages 1–9.

Butler, A., Hoffman, P., Smibert, P., Papalexi, E., and Satija, R. (2018). Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411–420.

Dasgupta, S. and Freund, Y. (2008). Random projection trees and low dimensional manifolds. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 537–546.

Dorfman, R. (1938). A note on the δ-method for finding variance formulae. *Biometric Bulletin*.

Dunn, P. K. and Smyth, G. K. (1996). Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5(3):236–244.

Germain, P.-L., Sonrel, A., and Robinson, M. D. (2020). pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biology*, 21(1):1–28.

Grün, D., Kester, L., and Van Oudenaarden, A. (2014). Validation of noise models for single-cell transcriptomics. *Nature Methods*, 11(6):637–640.

Hafemeister, C. and Satija, R. (2019). Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20(1):1–15.

Hafemeister, C. and Satija, R. (2020). Analyzing scRNA-seq data with the sctransform and offset models. https://satijalab.org/pdf/sctransform_offset.pdf.

Kharchenko, P. V. (2021). The triumphs and limitations of computational methods for scRNA-seq. *Nature Methods*.

Klein, A. M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D. A., and Kirschner, M. W. (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201.

Lause, J., Berens, P., and Kobak, D. (2021). Analytic pearson residuals for normalization of single-cell RNA-seq UMI data. *bioRxiv*.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550.

Lun, A. T. (2020). What transformation should we use? https://ltla.github.io/SingleCellThoughts/general/transformation.html. Accessed: 2021-06-01.

Lun, A. T., Bach, K., and Marioni, J. C. (2016). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, 17(1):1–14.

Messmer, T., von Meyenn, F., Savino, A., Santos, F., Mohammed, H., Lun, A. T. L., Marioni, J. C., and Reik, W. (2019). Transcriptional heterogeneity in naive and primed human pluripotent stem cells at single-cell resolution. *Cell Reports*, 26(4):815–824.

Osorio, D., Yu, X., Yu, P., Serpedin, E., and Cai, J. J. (2019). Single-cell RNA sequencing of a European and an African lymphoblastoid cell line. *Scientific data*, 6(1):1–8.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2009). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.

Svensson, V. (2020). Droplet scRNA-seq is not zero-inflated. *Nature Biotechnology*, 38(2):147–150.

Svensson, V., Natarajan, K. N., Ly, L.-H., Miragaia, R. J., Labalette, C., Macaulay, I. C., Cvejic, A., and Teichmann, S. A. (2017). Power analysis of single-cell RNA-sequencing experiments. *Nature Methods*, 14(4):381–387.

Tian, L., Dong, X., Freytag, S., Lê Cao, K.-A., Su, S., JalalAbadi, A., Amann-Zalcenstein, D., Weber, T. S., Seidi, A., Jabbari, J. S., et al. (2019). Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nature Methods*, 16(6):479–487.

Townes, F. W. (2019). Generalized principal component analysis. *arXiv*, abs/1907.02647.

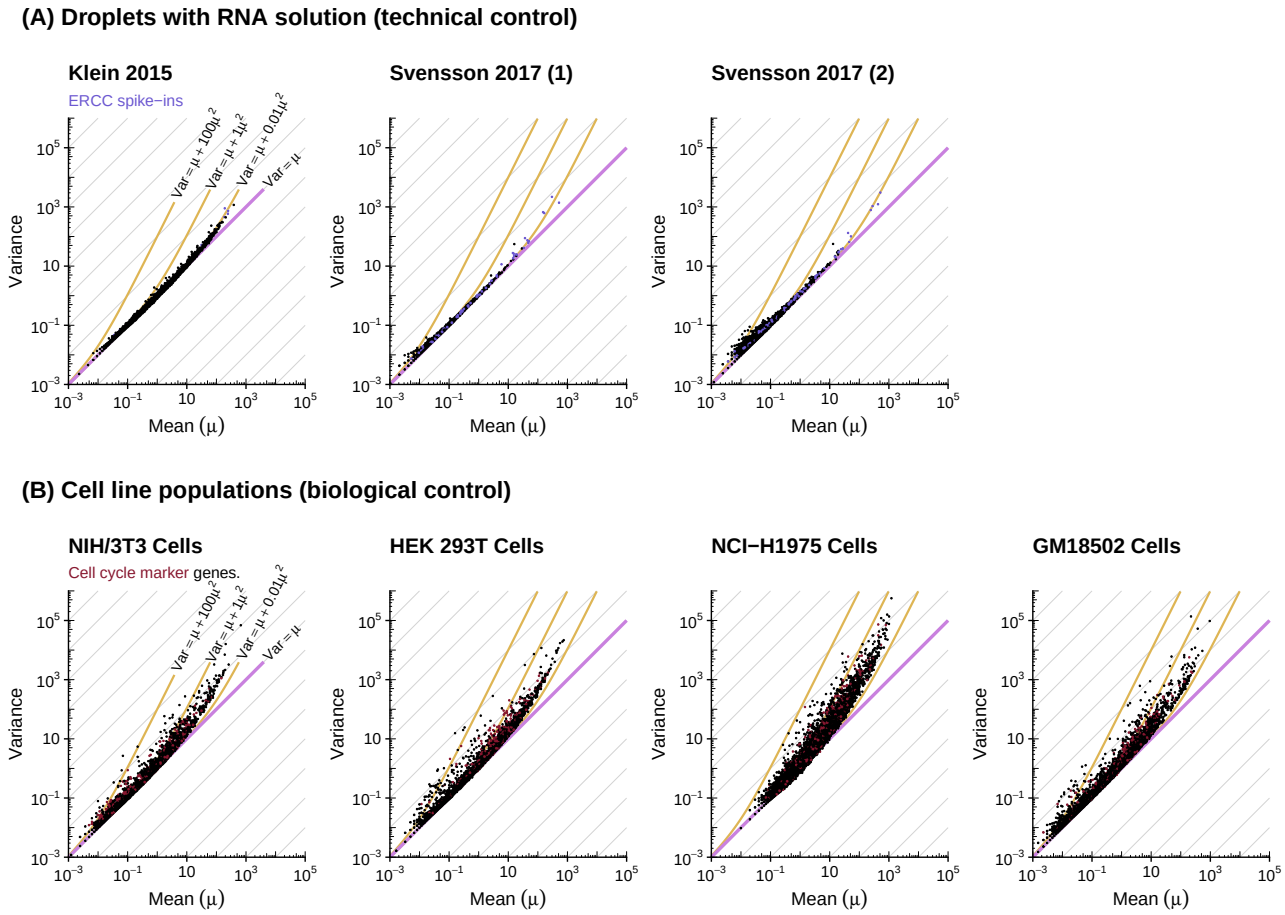
Warton, D. I. (2018). Why you cannot transform your way out of trouble for small counts. *Biometrics*, 74(1):362–368.

Wooldridge, J. M. (2013). *Introductory econometrics: A modern approach*. Cengage learning.

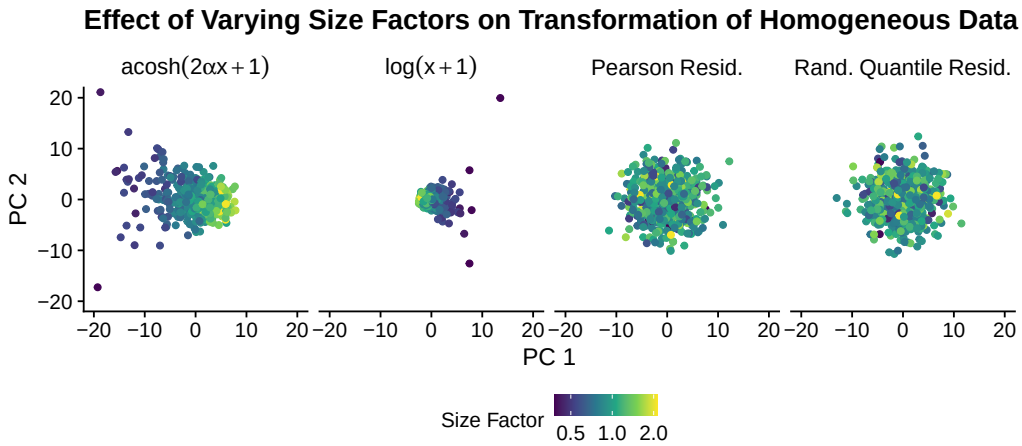
Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142.

Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., et al. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):1–12.

A Supplementary Figures



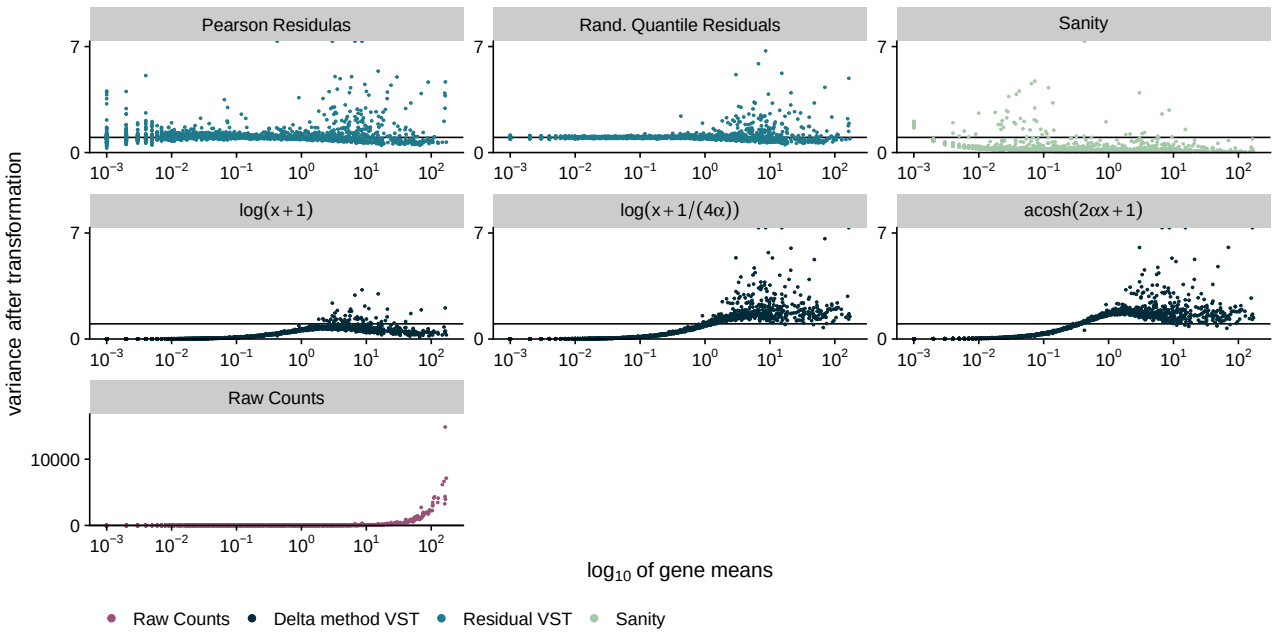
Suppl. Figure S1: Scatter plot on the log-log scale of the mean and variance per gene for technical and biological control experiments. (A) shows three datasets where endogenous RNA plus a known concentration of the External RNA Control Consortium (ERCC) spike-in standard has been captured in droplets so that the variations of gene's counts per droplet are purely statistical. The best overdispersion fit for genes with a mean of more than 1 were $\alpha = 0.006$, 0.011, and 0.015, respectively. (B) shows four immortalized cell line populations that are ostensibly homogeneous (cells from one mouse cell line and three human cell lines). The best overdispersion fit for genes with a mean of more than 1 were $\alpha = 0.12$, 0.07, 0.16, and 0.17, respectively. Cell cycle genes (gene ontology term GO:0007049) are highlighted in red; for these, we expect elevated variance even in a homogeneous cell population.
The diagonal line with slope 1 (purple) corresponds to the mean-variance relation of a Poisson distribution. The yellow lines indicate quadratic mean-variance relations with different coefficients for the quadratic term (corresponding to Gamma-Poisson distributions). To limit contributions of the sequencing coverage on the variance, only cells between the median and 1.3× the median of the size factor are shown.



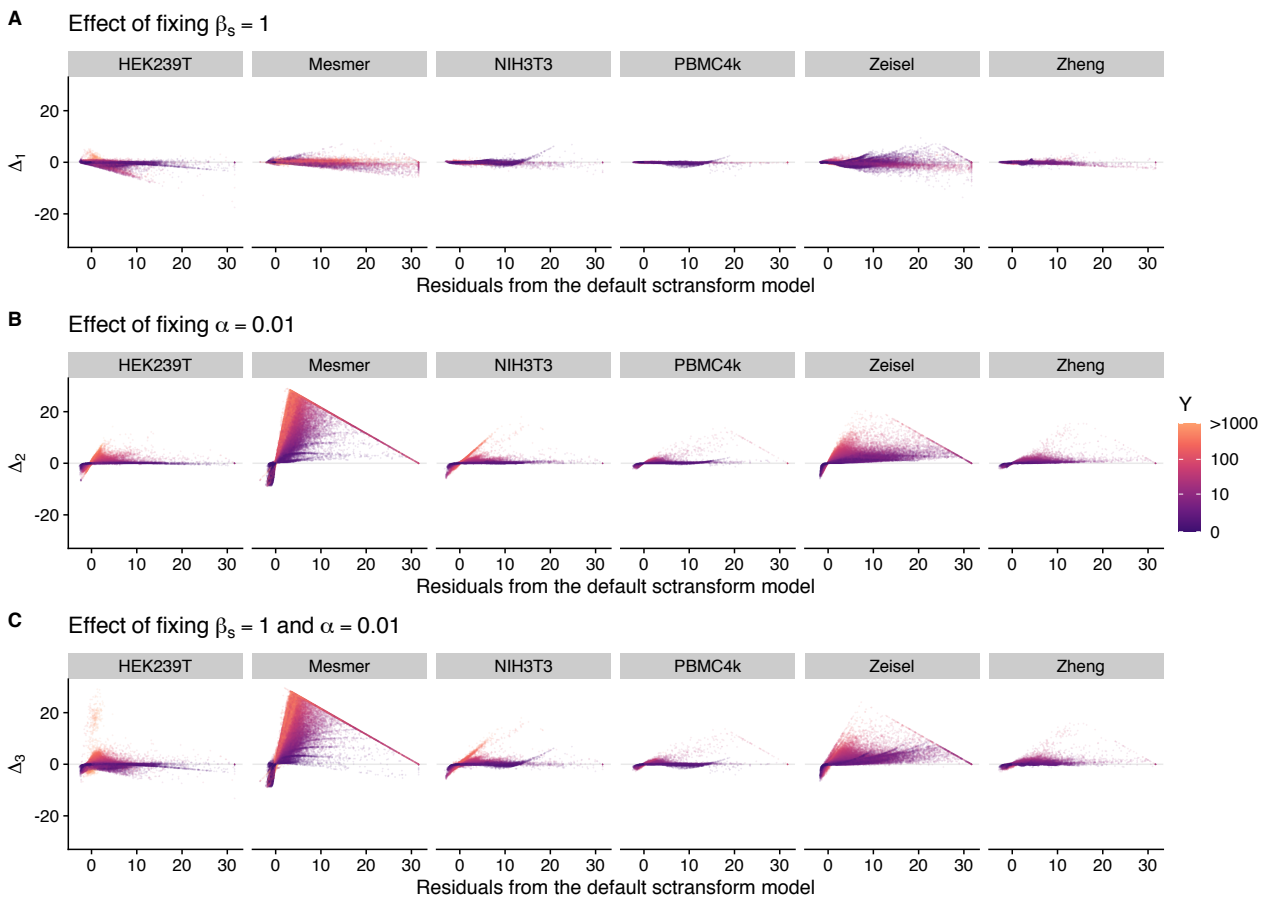
Suppl. Figure S2: Plots of the first two principal components of homogeneous data with size factors that vary across cells. We simulated 500 cells and 4000 genes according to the following model

$$
\begin{aligned}
Y_{ij} &\sim \text{GammaPoisson}(\mu_i s_j, \alpha_i) \\
\log(\mu_i) &\sim \text{Normal}(4, 2.6) \\
100\alpha &\sim \chi^2(5) \\
\log(s_j^*) &\sim \text{Normal}(4, 0.3) \\
s_j &= s_j^* / \text{mean}(s_j^*).
\end{aligned}
\tag{5}
$$

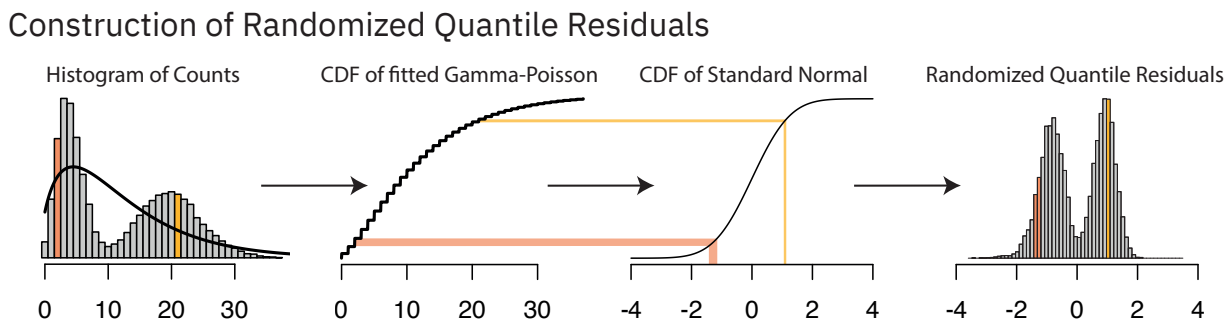
This figure was inspired by Lun (2020).



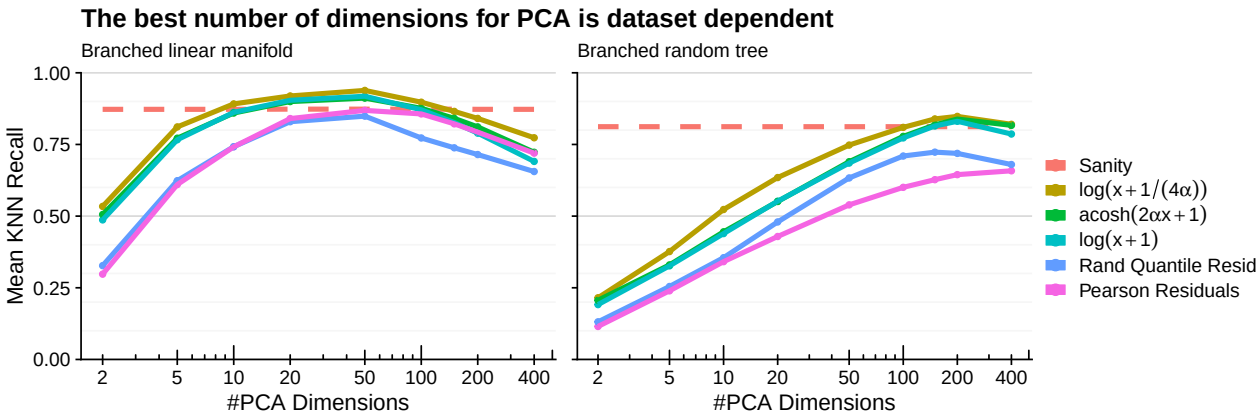
Suppl. Figure S3: Scatter plots of the variance per gene of the raw counts and after applying the six different transformations. The x-axis shows the logarithmized mean of the raw counts per gene; the y-axis shows the variance per gene after applying the transformations. We sampled 1,000 cells and 1,852 genes from the NIH/3T3 mouse cell line dataset. We chose the genes so that they uniformly cover the log mean expression space. We set $\alpha = 0.12$ using the estimate from Suppl. Fig. S1. The horizontal line highlights the target variance of 1.



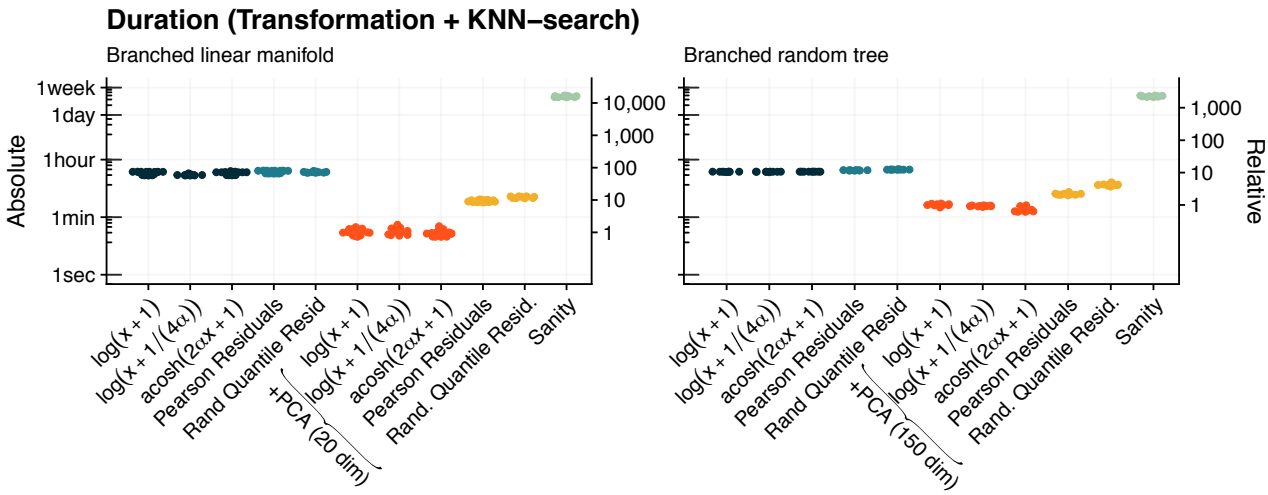
Suppl. Figure S4: Scatter plots to assess the importance of estimating β_s and/or α in Eq. (4). The x-axis shows the Pearson residuals calculated with *sctransform*'s default model (where both β_s and α are estimated); the larger the residual, the more the data point is an outlier. The y-axis shows the difference between the residuals from sctransform's default model and the residuals from a fit with fixed $\beta_s = 1$ and/or $\alpha = 0.01$; the extremer the difference, the more impact fixing that parameter has on the result. In (A), Δ_1 is the difference between sctransform's default model and the offset model ($\beta_s = 1$). In (B), Δ_2 is the difference between sctransform's default model and the fixed dispersion model ($\alpha = 0.01$). In (C), Δ_3 is the difference between sctransform's default model and the model suggested by Lause et al. (2021) ($\beta_s = 1$ and $\alpha = 0.01$).
The facets show 6 different single-cell datasets. From each, we sampled 3,000 genes and 1,000 cells. Each point is colored by the observed count. To fit the offset mode without fixing α, we forked sctransform and extended the provided offset routine from Hafemeister and Satija (2020) to allow estimation of α from the data. The diagonal line visible in the Mesmer, Zeisel, and Zheng data is an artifact from sctransform limiting the maximum value for the residual to $\sqrt{n}$.



Suppl. Figure S5: Schematic representation of how randomized quantile residuals are constructed. In the first step, a Gamma-Poisson distribution (black line) is fitted to the observed counts. Then, the quantiles of the Gamma-Poisson distribution are matched with the quantiles of a standard normal distribution by comparing their respective cumulative density functions (CDFs). This obtains a mapping from the raw count scale to a new, continuous scale. The two colored bars (orange for $y = 2$, yellow for $y = 21$) exemplify this mapping. The non-linear nature of the CDFs ensures that small counts are mapped to a broader range than large counts. This helps to stabilize the variance on the residual scale. Furthermore, the randomization within the mapping sidesteps the discrete nature of the counts.



Suppl. Figure S6: Line plot of the performance (mean recall of the 100 nearest neighbors) depending on the number of dimensions used for the principal component analysis (PCA). The performance of Sanity was included as a reference; it does not depend on the number of PCA dimensions because it is always fitted on the full data.



Suppl. Figure S7: Bee swarm plot of the CPU time of the transformation and k nearest neighbor (KNN) search for the benchmarks in Fig. 4. The secondary y-axis shows the performance relative to the median time observed when we transform the data with the shifted logarithm and reduce the dimensions using PCA (i.e., 20 and 143 seconds on the two datasets, respectively).

B Appendix

B.1 Variation of log fold changes and the coefficient of variation

The coefficient of variation for a random variable X_i is defined as

$$c_{\mathrm{v}} = \frac{\sqrt{\mathbb{V}\mathrm{ar}\left[X_i\right]}}{\mathbb{E}[X_i]}. \tag{6}$$

The variance of a log fold change for two independent random variables is

$$\mathbb{V}\mathrm{ar}\left[\log \frac{X_1}{X_2}\right] = \mathbb{V}\mathrm{ar}\left[\log X_1\right] + \mathbb{V}\mathrm{ar}\left[\log X_2\right]. \tag{7}$$

We can use the delta method to approximate

$$\mathbb{V}\mathrm{ar}\left[g(X_i)\right] \approx (g'(\mathbb{E}[X_i]))^2 \mathbb{V}\mathrm{ar}\left[X\right]. \tag{8}$$

The derivative of $\log(x)$ is

$$\frac{d}{dx}\log(x) = \frac{1}{x}. \tag{9}$$

We can now plug Eq. (8) and (9) into Eq. (7) and find that

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\log \frac{X_1}{X_2}\right] &\approx \frac{1}{\mathbb{E}[X_1]^2}\mathbb{V}\mathrm{ar}[X_1] + \frac{1}{\mathbb{E}[X_2]^2}\mathbb{V}\mathrm{ar}[X_2] \\ &= c_{\mathrm{v}1}^2 + c_{\mathrm{v}2}^2.\end{aligned} \tag{10}$$

This expression shows that the log fold changes decrease with the mean, as long as the coefficient of variation c_{v} decreases with the mean.

B.2 Approximating the acosh transformation with the shifted logarithm

The inverse hyperbolic cosine transformation from Eq. (1) is defined as

$$\begin{aligned}g(y) &= \frac{1}{\sqrt{\alpha}}\,\mathrm{acosh}\left(2\alpha y + 1\right) \\ &= \frac{1}{\sqrt{\alpha}}\log\left(2\alpha y + \sqrt{(2\alpha y+1)^2 - 1} + 1\right).\end{aligned} \tag{11}$$

We want to approximate this transformation using the shifted logarithm and thus find a, b, and c in

$$h(y) = a + b\log(y + c), \tag{12}$$

so that $h(y) \approx g(y)$.

To find a, b, and c, so that for large y, $h(y)$ converges as quickly as possible to $g(y)$, we notice that

$$\lim_{y\to\infty} \frac{\sqrt{(2\alpha y+1)^2 - 1}}{2\alpha y} = 1 \tag{13}$$

and thus for large y

$$\begin{aligned}g(y) &\approx \frac{1}{\sqrt{\alpha}}\log\left(4\alpha y + 1\right) \\ &= \frac{1}{\sqrt{\alpha}}\log\left(y + \frac{1}{4\alpha}\right) + \frac{\log\left(4\alpha\right)}{\sqrt{\alpha}}\end{aligned} \tag{14}$$

The linear scaling b and the offset a do not influence the variance stabilization; the important insight is that the pseudo-count $c = \frac{1}{4\alpha}$ ensures that the shifted logarithm is most similar to the variance-stabilizing transformation derived using the delta method.

B.3 Delta method based variance-stabilizing transformation and size factors

Suppl. Fig. S2 shows that delta method-based variance-stabilizing transformations struggle to incorporate varying size factors.

To incorporate cell-specific size factors in the delta method-based variance stabilizing transformation approach, the counts K_{ij} are divided by the size factor s_j before applying the transformation: $g(K_{ij}/s_j)$ (Love et al., 2014). To see the implications of this, it is helpful to look at a decomposition of the variance of a Gamma-Poisson random variable K:

$$\begin{aligned}K|Q &\sim \mathrm{Poisson}(Q) \\ Q &\sim \mathrm{Gamma}(\mu, \alpha) \\ K &\sim \mathrm{GammaPoisson}(\mu, \alpha).\end{aligned} \tag{15}$$

In the context of RNA-seq count data, the Poisson level of this hierarchical model represents the technical sampling noise and Q models additional variation. According to the law of total variation

$$\begin{aligned}\mathbb{V}\mathrm{ar}[K] &= \mathbb{E}[\mathbb{V}\mathrm{ar}(K|Q)] + \mathbb{V}\mathrm{ar}[\mathbb{E}(K|Q)] \\ &= \mu + \alpha\mu^2,\end{aligned} \tag{16}$$

where $\mathbb{V}\mathrm{ar}[K|Q] = \mu$ and $\mathbb{V}\mathrm{ar}[Q] = \alpha\mu^2$.

If we apply the same approach to a model with size factors

$$K'|Q, s \sim \mathrm{Poisson}(sQ), \tag{17}$$

we find that

$$\begin{aligned}\mathbb{V}\mathrm{ar}[K'] &= \mathbb{E}[\mathbb{V}\mathrm{ar}(K'|Q)] + \mathbb{V}\mathrm{ar}[\mathbb{E}(K'|Q)] \\ &= s\mu' + \alpha s^2 \mu'^2 \\ &= \mu + \alpha\mu^2\end{aligned} \tag{18}$$

where $\mu = s\mu'$.

If, however, we want to apply the delta method-based variance-stabilizing transformation to a size factor standardized count

$$Y = K'/s, \tag{19}$$

we find that

$$\begin{aligned} \mathbb{V}\mathrm{ar}[Y] &= \frac{1}{s^2}\mathbb{V}\mathrm{ar}[K'] \\ &= \frac{1}{s^2}(s\mu' + \alpha s^2 \mu'^2) \\ &= \frac{1}{s}\mu' + \alpha\mu'^2 \end{aligned} \tag{20}$$

The difference between the final line of Eq. (18) and Eq. (20) explains the problem observed when applying the delta method-based variance-stabilizing transformation to correct data where the size factors vary a lot between cells.

C Data Availability

In this manuscript, we used several different single-cell datasets, all of which have been previously published.

Effect of transformation on three marker genes	Mouse lung	Angelidis et al. (2019)	GEO GSE124872
Basis for simulating the branched linear and random walk datasets	Human pancreas	Baron et al. (2016)	scRNAseq Bioconductor package
Effect of fixing β_{is} and/or α in *sctransform*	HEK 293T	10X Genomics (2018)	`https://data.caltech.edu/records/1264`
	Mesmer	Messmer et al. (2019)	scRNAseq Bioconductor package
	NIH/3T3	10X Genomics (2018)	`https://data.caltech.edu/records/1264`
	PBMC4k	10X Genomics (2017)	TENxPBMCData Bioconductor package
	Zeisel	Zeisel et al. (2015)	scRNAseq Bioconductor package
	Zheng	Zheng et al. (2017)	DuoClustering2018 Bioconductor package
Mean-variance relation	Klein	Klein et al. (2015)	`https://data.caltech.edu/records/1264`
	Svenson 1,2	Svensson et al. (2017)	`https://data.caltech.edu/records/1264`
	NCI-H1975	Tian et al. (2019)	`https://github.com/LuyiTian/sc_mixology/blob/master/data/csv/sc_10x.count.csv.gz`
	GM18502	Osorio et al. (2019)	GEO GSE126321