Transformation and Preprocessing of Single-Cell RNA-Seq Data

Constantin Ahlmann-Eltze[⋆,†] and Wolfgang Huber[⋆]

[⋆] *Genome Biology Unit, EMBL, Heidelberg, 69117, Germany.*
[†] *Collaboration for joint PhD degree between EMBL and Heidelberg University, Faculty of Biosciences*

August 25, 2021

Abstract

The count table, a numeric matrix of genes × cells, is the 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 find that although the residuals and latent expression state-based models have appealing theoretical properties, in benchmarks using simulated and real-world data the simple shifted logarithm in combination with principal component analysis performs surprisingly well.
Software: An R package implementing the delta method and residuals-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. 1A, we provide a schematic example of heteroskedastic data: 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 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 single-cell RNA sequencing counts—a theoretically and empirically well-supported and widely used choice is the Gamma-Poisson distribution[1] (Grün et al., 2014; Svensson, 2020; Kharchenko, 2021)—address the problem of heteroskedasticity, but their parameter inference 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 use the many existing statistical methods that implicitly or explicitly assume uniform variance for best performance (Amezquita et al., 2020; Kharchenko, 2021).

[1] also referred to as Negative Binomial distribution

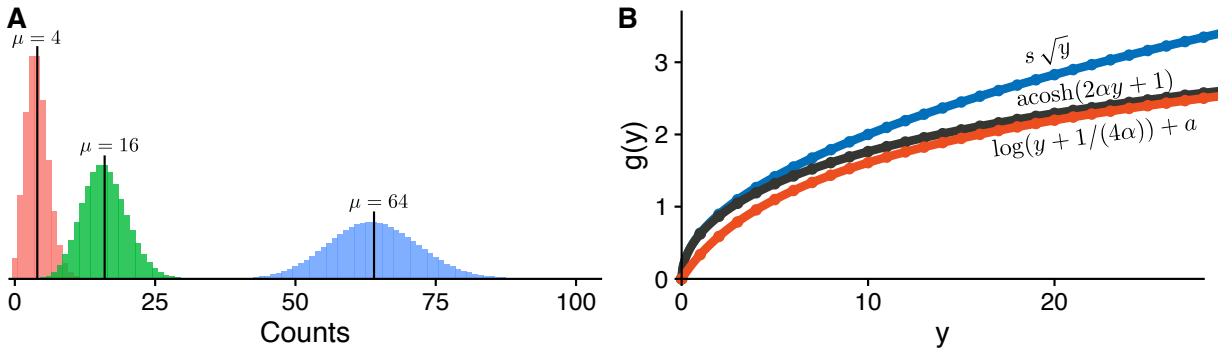


Figure 1: Example of heteroskedastic data. (A) Histograms of three Poisson distributions with increasing means. (B) Graph of the three delta method-based variance-stabilizing transformations that are most relevant for count data. The curves are shown for overdispersion parameter $\alpha = 0.1$. We chose the offset $a = \log(4\alpha)$ in the shifted logarithm and the scaling $s = 2\sqrt{\alpha}$ in the square-root transformation to match the acosh transformation.

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

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

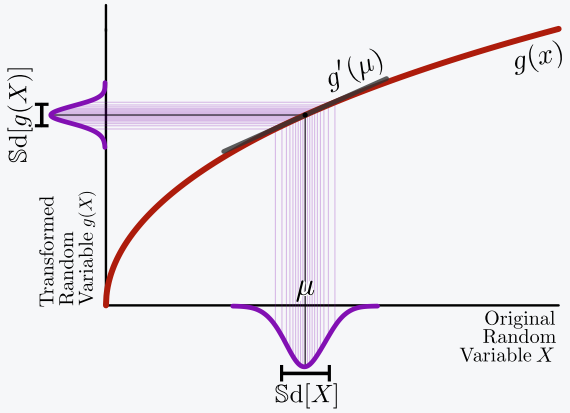
The shifted log transformation

$$g(y) = \log(y + c) \qquad (2)$$

is a good approximation for Eq. (1) if the pseudo-count is $c = \frac{1}{4\alpha}$ (see Fig. 1B 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 $\alpha \ll 0.01$.

What is the delta method?

The delta method is a way to find the standard deviation of a transformed random variable.



If we apply a function g to a random variable X with mean μ, the standard deviation of the transformed random variable $g(X)$ can be approximated by

$$\mathbb{S}\mathrm{d}[g(X)] \approx a\, \mathbb{S}\mathrm{d}[X],$$

where $a = |g'(\mu)|$ is the slope of g at μ.

Now consider a set of random variables $X_1, X_2, \ldots$ whose variances and means are related through some function v, i. e., $\mathbb{V}\mathrm{ar}[X_i] = v(\mu_i)$, or equivalently $\mathbb{S}\mathrm{d}[X_i] = \sqrt{v(\mu_i)}$. Then we can find a variance-stabilizing transformation g by requiring constant standard deviation, $\mathbb{S}\mathrm{d}[g(X_i)] = \text{const.}$, which using the above approximation becomes

$$g'(x) = \frac{\text{const.}}{\sqrt{v(\mu)}},$$

and can be solved by integration.

One problem for variance-stabilizing transformations based on the delta method are the so-called *size factors*. There is one of these per cell, meant to simultaneously adjust for variable cell sizes and variable efficiency with which molecules are sampled from the mRNA pool of a cell during the measurement process (Lun et al., 2016). There is a variety of approaches to estimate size factors from the data, but the common next step is to divide the counts by the estimates before the variance-stabilizing transformation is applied (Love et al., 2014; Amezquita et al., 2020)[2].

However, this operation does not completely remove the confounding effects: 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, but the variability of the resulting values is different. 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) pointed out, that transformations cannot reasonably be expected to stabilize the variance of small counts. Suppl. Fig. S3 shows that the variance per gene after transformation with the shifted log or acosh is practically zero for a mean expression of less than 0.1.

Instead of transforming the raw counts, the shifted logarithm is sometimes applied to the counts per million (CPM)

$$\mathrm{CPM}_i = \frac{y_i}{\sum_i y_i} \times 10^6, \tag{3}$$

where y_i is the count for gene i and $\sum_i y_i$ is the total number of UMIs per the cell. Typical droplet-based single-cell data have an average sequencing depth of $\sum_i y_i \approx 5,000$. Thus, this approach is equivalent to using a pseudo-count of $c = 0.005$ in Eq. (2), which amounts to assuming an overdispersion of $\alpha = 250$, three orders of magnitude larger than the overdispersions of real-world single-cell datasets. This is problematic because it means that observing a zero or an one has an outsized influence on the result (see column 2 vs. 3 in Fig. 2).

[2]Conventionally, the size factors are scaled to be close to 1, e. g., by dividing them by their geometric mean. Thus, the size factor adjusted counts have about the same range as the raw counts.

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

$$r = \frac{y - \mu}{\sqrt{\mu + \alpha\mu^2}}, \tag{4}$$

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

$$\begin{aligned} Y_{ij} &\sim \mathrm{GammaPoisson}\left(\mu_{ij}, \alpha_i\right) \\ \log(\mu_{ij}) &= \beta_{i0} + \beta_{i\mathrm{s}} \log(s_j), \end{aligned} \tag{5}$$

where $j = 1, \ldots, J$ is the cell index, s_j is the size factor (one for each cell), and β_{i0} and $\beta_{i\mathrm{s}}$ are intercept and slope parameters (one for each gene). Note that the denominator in Eq. (4) 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). By using the gene-wise mean and standard deviation estimate, the transformation ensures that also the variances of lowly expressed genes are stabilized (Suppl. Fig. S3). Note, however, that for lowly expressed genes the denominator in Eq. (4) can be very small, so that due to the discreteness of the counts, individual r values can get very large. Hafemeister and Satija (2019) address this by clipping r to the range $[-\sqrt{J}, +\sqrt{J}]$.

sctransform, the implementation of the Pearson residuals method provided by Hafemeister and Satija (2019), performed well in a recent benchmark (Germain et al., 2020), however, there has been a debate around its statistical model. Lause et al. (2021) argued that the model was overparameterized, and that neither the estimation of $\beta_{i\mathrm{s}}$ nor the estimation of a different overdispersion parameter for each gene were necessary. Instead, Lause et al. (2021) suggested fixing $\beta_{i\mathrm{s}} = 1$ and the overdispersion to $\alpha = 0.01$; the latter being roughly the overdispersion they observed in experiments where an RNA solution is homogeneously encapsulated in droplets. In the latest version of sctransform and after setting `vst.flavor="v2"`, fixing $\beta_{i\mathrm{s}} = 1$ has become the default (Choudhary and Satija, 2021). However,

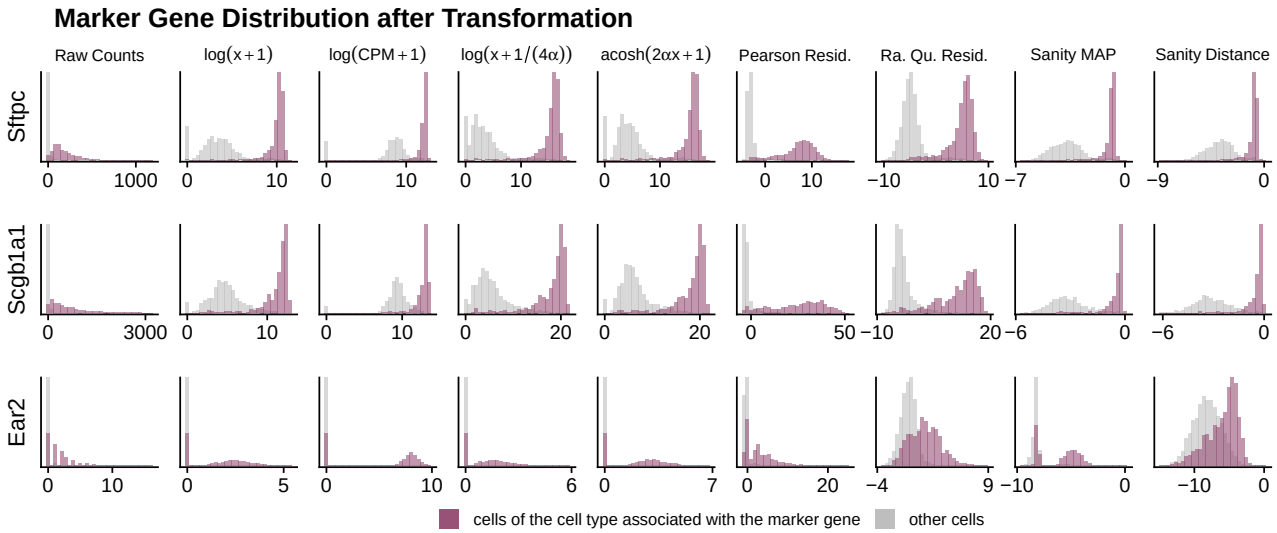


Figure 2: Distribution of the transformed values for three genes with strong expression differences across subgroups of cells. We show histograms of the raw gene counts and the transformed values for three cell-type marker genes in a mouse lung dataset (Angelidis et al., 2019). The color of the bars 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). We down-sampled the other cells (grey) to match the number of those from the marked cell type for ease of visualisation. The x in the transformations refers to the size-factor s normalized counts $x = y/s$.

on the question of fixing the overdispersion to a small value, Hafemeister and Satija (2020) responded that this over-emphasizes the variation of highly abundant housekeeping genes.

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 confirmed 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, 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. This finding agrees with the results from a recent follow-up study by the authors of sctransform, who analyzed the overdispersion across a large number of datasets (Choudhary and Satija, 2021).

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. 2. The figure shows the expression pattern of three

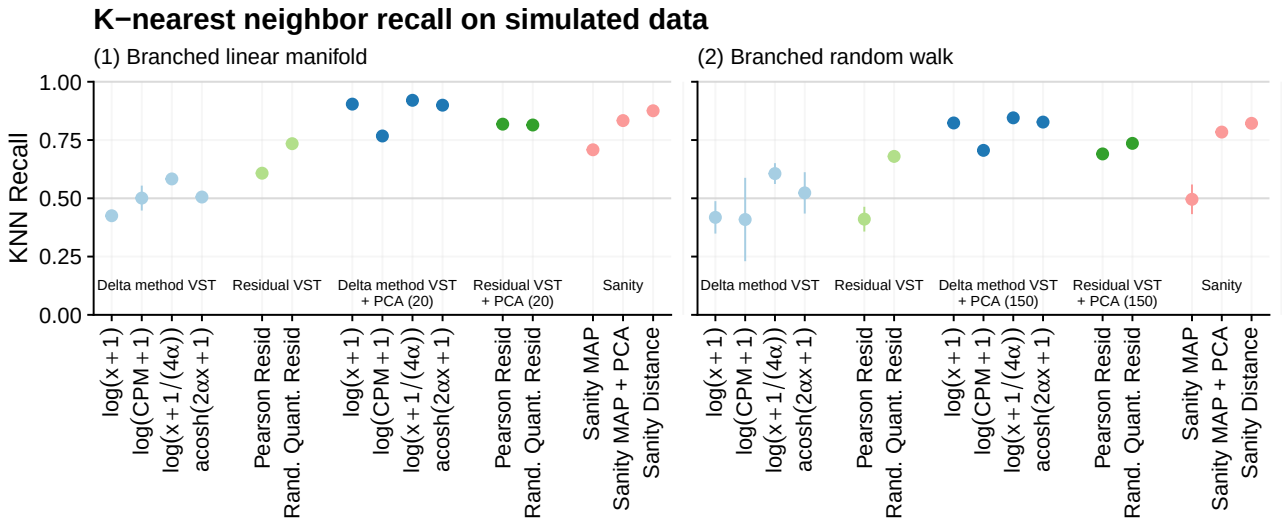


Figure 3: Benchmark of the k nearest neighbor recall on two simulated branching datasets: (1) each branch is a linear interpolation between two points, (2) each branch is a random walk (using the same simulation approach as Breda et al. (2021)). On the linear manifold dataset, we used the first 20 principal components; on the random walk dataset, we used 150. For both, we use $k = 100$. 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. The point ranges show the mean and standard deviation across 10 replicates.

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. (4)). 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 modeling, 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. S4 shows how they are constructed.) Same as Pearson residuals, randomized quantile residuals stabilize the variance for small counts (Suppl. Fig. S3) but stabilize the variance also if a gene's expression strongly differs across cells (Fig. 2).

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 the logarithm of the gene activity as a data transformation.

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 for the logarithm of the gene activity, 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.

There are two ways to incorporate the posterior from Sanity into a standard single-cell analysis workflow: 1) Sanity provides its own method to calculate a cell-by-cell distance matrix, which can then, for instance, be used to identify the k nearest neighbors of each cell. 2) Ignore the inferred uncertainties, use the maximum of the posterior as the transformed value, and apply the usual Euclidean geometry. To distinguish the two approaches, we call the first *Sanity Distance* and the second *Sanity MAP*.

Breda et al. (2021) presented a benchmark that showed that Sanity was the best method for identifying the k nearest neighbors of a cell. However, we find that the delta method-based and residuals-based variance-stabilizing transformations perform similarly well if we project the cells to a lower-dimensional representation using principal component analysis (PCA) before computing Euclidean distances and looking for nearest neighbors (Fig. 3). 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, this PCA-based approach requires the choice of the number of dimensions, which can greatly affect the performance (Suppl. Fig. S5).

A limitation of Sanity is that it is slow compared to the other transformations (Suppl. Fig. S6). In our experiments, Sanity MAP needed 500-1,800× more CPU time than the shifted logarithm. Sanity Distance was even slower (3,000-10,000× more CPU than the shifted logarithm) because its distance calculation, which takes into account the uncertainty for the nearest neighbor search, scales quadratically with the number of cells. In contrast, the other transformations (including Sanity MAP) 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.

Benchmark on Subsampled Counts

There is no straightforward measure of success for a preprocessing method, as it is contingent on the subsequent analysis and its objectives. For instance, if interest lies on identification of cell type-specific marker genes, one could assess the shape of distributions such as in Figure 2, or the performance of a supervised classification method. Here, we will consider the use case that arguably has been the main driver of single-cell RNA-Seq development and applications so far: understanding the variety of cell types and states in terms of a lower-dimensional mathematical structure, such as a planar embedding, a clustering, trajectories, branches, or combinations thereof. For all of these, it is common to approach them via the k nearest neighbor graph. Even given the objective, a second difficulty is the definition of "ground truth". Thus, previous benchmarks employed synthetic data (Breda et al., 2021; Lause et al., 2021, Fig. 3 of this paper), qualitative inspection of non-linear dimension reduction plots (Hafemeister and Satija, 2019; Lause et al., 2021), or focused on clustering, one of the simplest type of structure (Germain et al., 2020).

Here, we take another approach, which tries to infer a ground truth nearest neighbor graph from recent deeply sequenced single-cell datasets based on molecular crowding (mcSCRB-seq, Smart-seq3) to benchmark the performance of different transformations when infering the k nearest neighbor graph from the same data subsampled to sequencing depths that are typical for the currently prevalent droplet based techniques. For each cell, we identify a set of common nearest neighbors as the intersection of the k nearest neighbors across all candidate transformations applied to the deeply sequenced dataset. In the next step, repeatedly using subsampled datasets, we compare for each transformation how many of the common nearest neighbors are recalled (Fig. 4A). While this benchmark is not perfect, we suggest that it is still informative: first, the impact of the different transformations is smaller on the full dataset (where the counts are higher) than on the subsampled datasets (e.g., a median of 11 common neighbors per cell on the full Fibroblast (2) dataset vs. 1 on the subsampled dataset), and thus the analysis of the full datasets provides a reasonably neutral reference. Second, the common nearest neighbors provide a noisy ground truth, and this will lead to bias in the calculation of performance measures such as receiver operating characteristics (ROC) or recall at fixed call size, but importantly, still provides valid ranking of methods if the noise is sufficiently random (Bourgon, 2006).

Following this approach, we employed four deeply sequenced datasets (Bagnoli et al., 2018; Hagemann-Jensen et al., 2020; Larsson et al.,

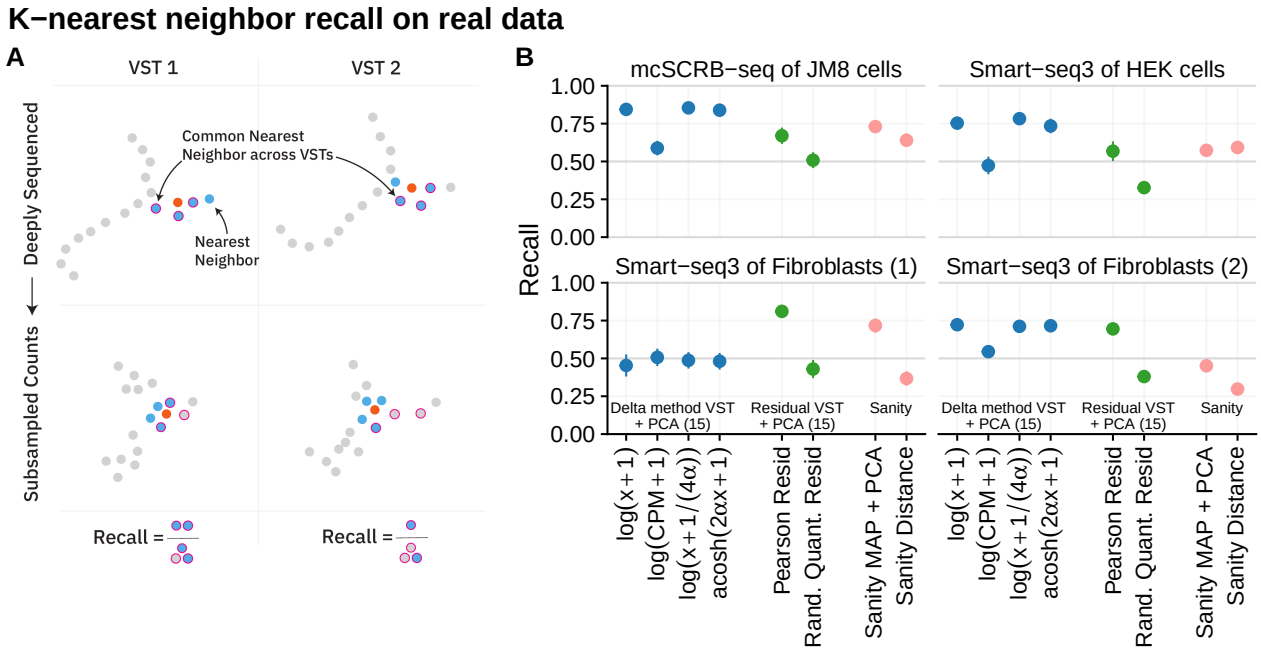


Figure 4: Benchmark of k nearest neighbor inference on real-world data. We subsample deeply sequenced single-cell data and measure the recall of common nearest neighbors identified across all candidate variance-stabilizing transformations (VSTs). A) Schematic of how the benchmark works; in blue are the four nearest neighbors of the orange cell, the pink border indicates common neighbors across transformations on the deeply sequenced data. B) Plots of the performance ($k = 30$) on four deeply sequenced single-cell datasets subsampled to a median of 5,000 UMIs per cell. The point ranges show the mean and standard deviation across 10 replicates.

2021) with a median of 55,000 to 198,000 UMIs per cell and subsampled them to a median of 5,000 UMIs per cell. We used $k = 30$, and the top 15 principal components in PCA. These numbers are smaller than for the synthetic data benchmark because the datasets have only 249-741 cells. We also only consider the delta method, residuals-based and Sanity MAP transformations in combination with PCA because of clear performance advantage we found in the benchmark on synthetic data.

For three of the four datasets, the results shown in Fig. 4B are broadly in agreement with the synthetic data benchmark and indicate good performance of the delta method-based variance-stabilizing transformations. However, for the first fibroblast dataset from Hagemann-Jensen et al. (2020), we find the opposite result, a clear advantage for Pearson residuals and Sanity MAP + PCA (but not Sanity Distance). Closer inspection of this dataset however shows an atypically large overdispersion of $\alpha \approx 0.8$, which is about 2.5 times higher than $\alpha \approx 0.3$ observed in a second dataset by the same authors on the same cell type using the same protocol (Larsson et al., 2021). Consistent with a potential data quality issue, the common neighbor sets in this dataset also were smaller than in the other three (Suppl. Fig. S7).

Discussion

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

We recommend against using counts per million as input for the shifted logarithm. We have shown that for typical datasets this amounts to assuming an unrealistically large overdispersion. Accordingly, our benchmarks show poor perfor-

mance by this approach.

The Pearson residuals-based variance-stabilizing transformation approach first developed by Hafemeister and Satija (2019) has attractive theoretical properties. It stabilizes the variance across all genes and is less sensitive to variations of the size factor. However, as exemplified in Figure 2, the fact that it is a linear function of the data for a gene across cells reduces its suitability for comparisons along this direction, such as differential expression analysis between cell subpopulations or visualization. There is no variance stabilization if a gene's expression strongly differs across cells. To fix this, we considered using non-linear residuals, such as randomized quantile residuals. However, in our benchmarks, 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 of its biological interpretability and mathematical common sense, and in particular, because the Sanity Distance method does not have any tunable parameters. Interestingly, the related Sanity MAP method, which ignores the uncertainty in the posterior, in combination with PCA performed nearly as well as Sanity Distance on our simulated data and outperformed Sanity Distance on three of the four benchmarks on real data. A further advantage of Sanity MAP is that it is directly compatible with existing downstream statistical methods that expect as their input a single value per gene and cell. This means that Sanity MAP is compatible with methods that scale linearly with the number of cells for k nearest neighbors inference unlike Sanity Distance which scales quadratically. Yet, Sanity MAP was still 500-1,800× slower than alternative transformations and did not show consistently better performance than delta method-based transformations in combination with PCA.

The results of our analysis agree with results from a recent preprint by Lause et al. (2021). They benchmarked combinations of gene selection and transformation approaches (e.g., Pearson residuals for gene selection and then working on the shifted log-transformed counts for k nearest neighbors identification). They concluded that Pearson residuals are the best approach for selecting biologically meaningful genes, but at the same time show that subsetting the genes is usually detrimental for the k nearest neighbors identification performance or neutral at best. Furthermore, for the k nearest neighbors identification, the shifted log transformation with a pseudo-count of $c = 1$ performs as well as Pearson residuals on their synthetic benchmark dataset.

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. At first, it might be surprising that reducing the dimensionality of data using PCA *improves* the result, as it is a lossy compression. However, it appears to be helpful that it smoothes out uncorrelated noise in the original feature space. Thus, in the future, we expect attention to shift away from simple variance stabilizing transformations to transformations that work well in combination with dimensionality reduction.

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

Availability

An R package that provides convenient methods for the delta method and residuals-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. All datasets used in this manuscript are listed in Appendix C.

Acknowledgments

We thank Dr. Simon Anders for extensive discussions about variance-stabilizing transformations and how to benchmark preprocessing methods. Furthermore, we thank Prof. Dr. Erik van Nimwegen and Dr. Dmitry Kobak, whose feedback on an earlier version helped to improve the manuscript.

Funding

This work has been supported by the EMBL International Ph.D. Programme, by the German Federal Ministry of Education and Research (CompLS project SIMONA under grant agreement no. 031L0263A), and the European Research Council (Synergy Grant DECODE under grant agreement no. 810296).

References

10X Genomics (2018). 5k 1:1 mixture of fresh frozen human (HEK293T) and mouse (NIH3T3) cells (v3 chemistry). 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.

Bagnoli, J. W., Ziegenhain, C., Janjic, A., Wange, L. E., Vieth, B., Parekh, S., Geuder, J., Hellmann, I., and Enard, W. (2018). Sensitive and powerful single-cell RNA sequencing using mcSCRB-seq. *Nature Communications*, 9(1):1–8.

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.

Bourgon, R. W. (2006). Chromatin-immunoprecipitation and high-density tiling microarrays: a generative model, methods for analysis, and methodology assessment in the absence of a 'gold standard'. https://www.huber.embl.org/pub/pdf/Richard-Bourgon-PhD-thesis-UCB-2006.pdf.

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.

Choudhary, S. and Satija, R. (2021). Comparison and evaluation of statistical error models for scrna-seq. *bioRxiv*.

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.

Hagemann-Jensen, M., Ziegenhain, C., Chen, P., Ramsköld, D., Hendriks, G.-J., Larsson, A. J., Faridani, O. R., and Sandberg, R. (2020). Single-cell RNA counting at allele and isoform resolution using Smart-seq3. *Nature Biotechnology*, 38(6):708–714.

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.

Larsson, A. J., Ziegenhain, C., Hagemann-Jensen, M., Reinius, B., Jacob, T., Dalessandri, T., Hendriks, G.-J., Kasper, M., and Sandberg, R. (2021). Transcriptional bursts explain autosomal random monoallelic expression and affect allelic imbalance. *PLoS computational biology*, 17(3):e1008772.

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.

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.

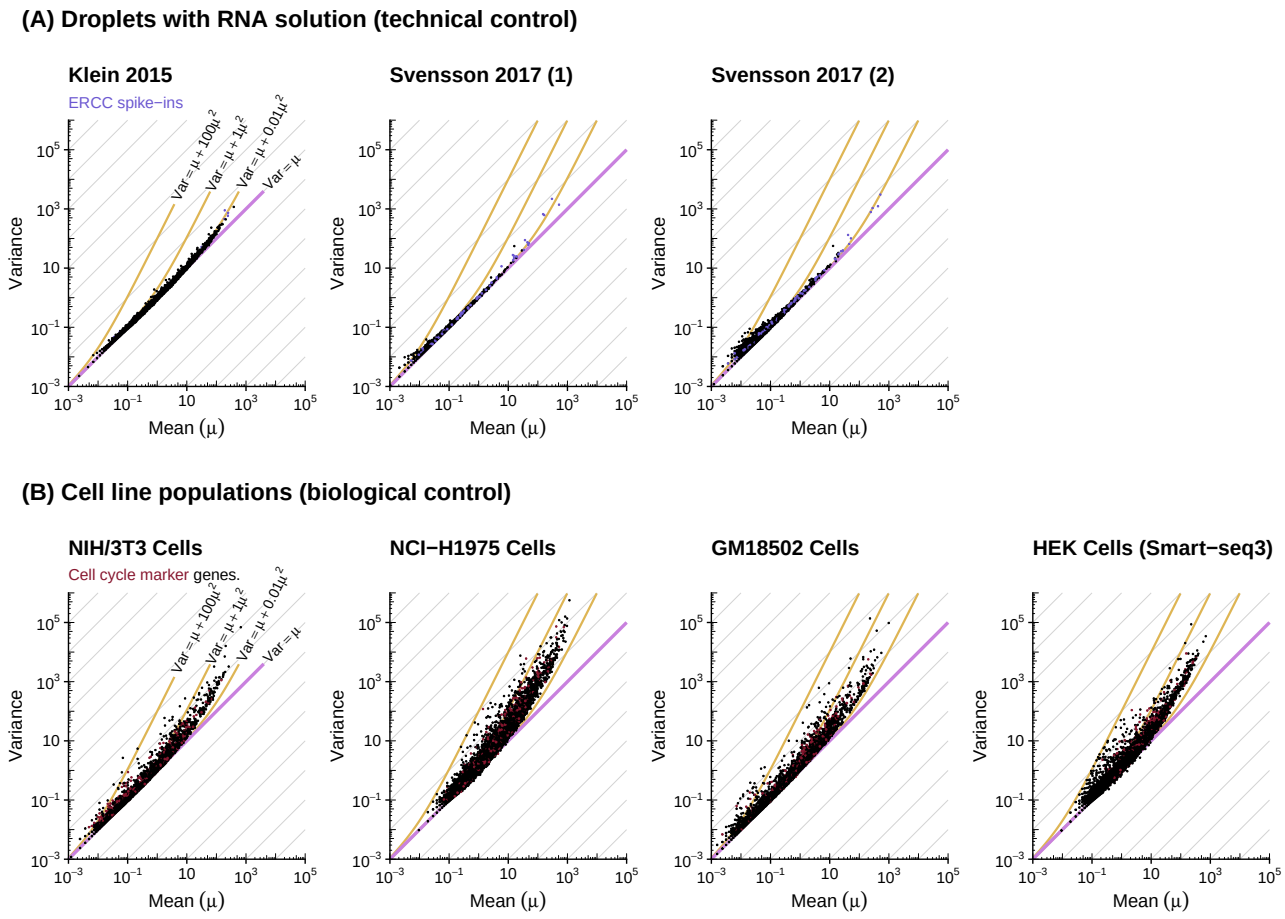
Tian, L., Dong, X., Freytag, S., Lê Cao, K.-A., Su, S., JalalAbadi, A., Amann-Zalcenstein, D., Weber, T. S., Seidi, A., Jabbari, J. S., et al. (2019). Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nature Methods*, 16(6):479–487.

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

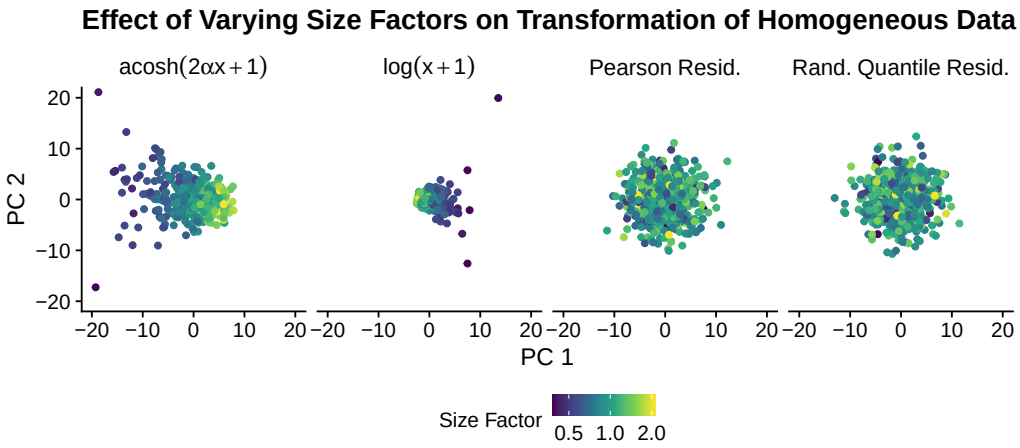
Warton, D. I. (2018). Why you cannot transform your way out of trouble for small counts. *Biometrics*, 74(1):362–368.

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

A Supplementary Figures



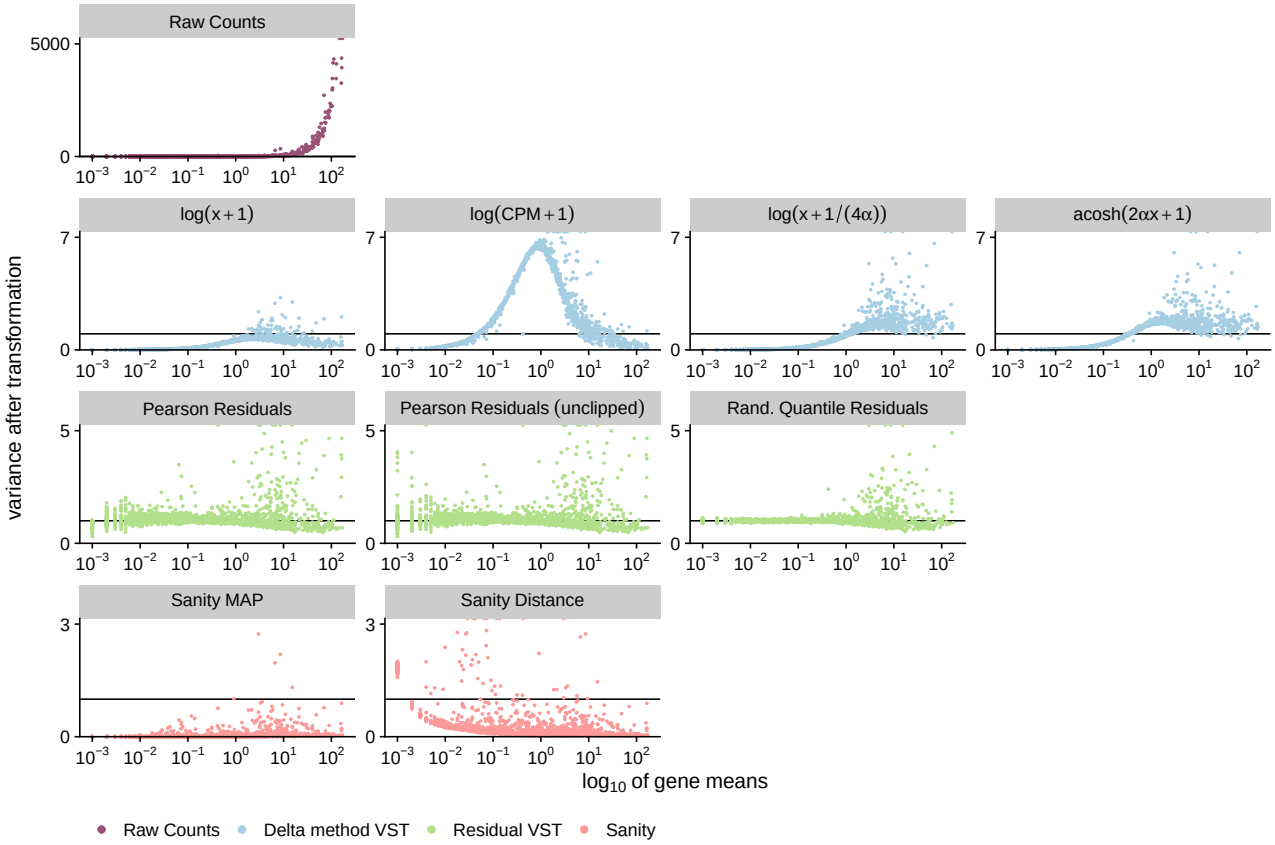
Suppl. Figure S1: Scatter plots on the log-log scale of the mean and variance per gene for technical and biological control experiments. (A) shows three datasets where endogenous RNA plus a known concentration of the External RNA Control Consortium (ERCC) spike-in standard has been captured in droplets so that the variations of a gene's counts per droplet are purely statistical. The best overdispersion fit for genes with a mean of more than 1 for the three tested datasets 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 $\alpha = 0.12$, 0.16, 0.17, and 0.13, 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. A table with the references for all seven datasets is in Appendix C.



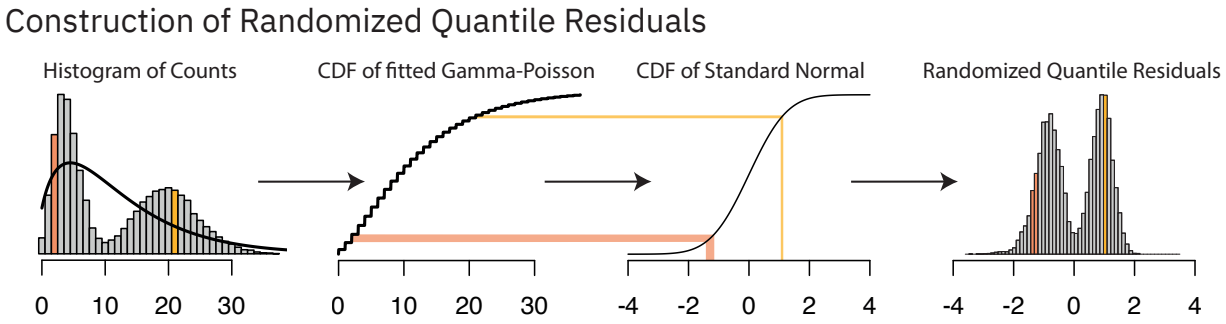
Suppl. Figure S2: Plots of the first two principal components of homogeneous data with size factors that vary across cells. We simulated 500 cells and 4000 genes according to the following model

$$\begin{aligned} Y_{ij} &\sim \text{GammaPoisson}(\mu_i s_j, \alpha_i) \\ \log(\mu_i) &\sim \text{Normal}(4, 2.6) \\ 100\alpha &\sim \chi^2(5) \\ \log(s_j^*) &\sim \text{Normal}(4, 0.3) \\ s_j &= s_j^* / \text{mean}(s_j^*). \end{aligned} \tag{6}$$

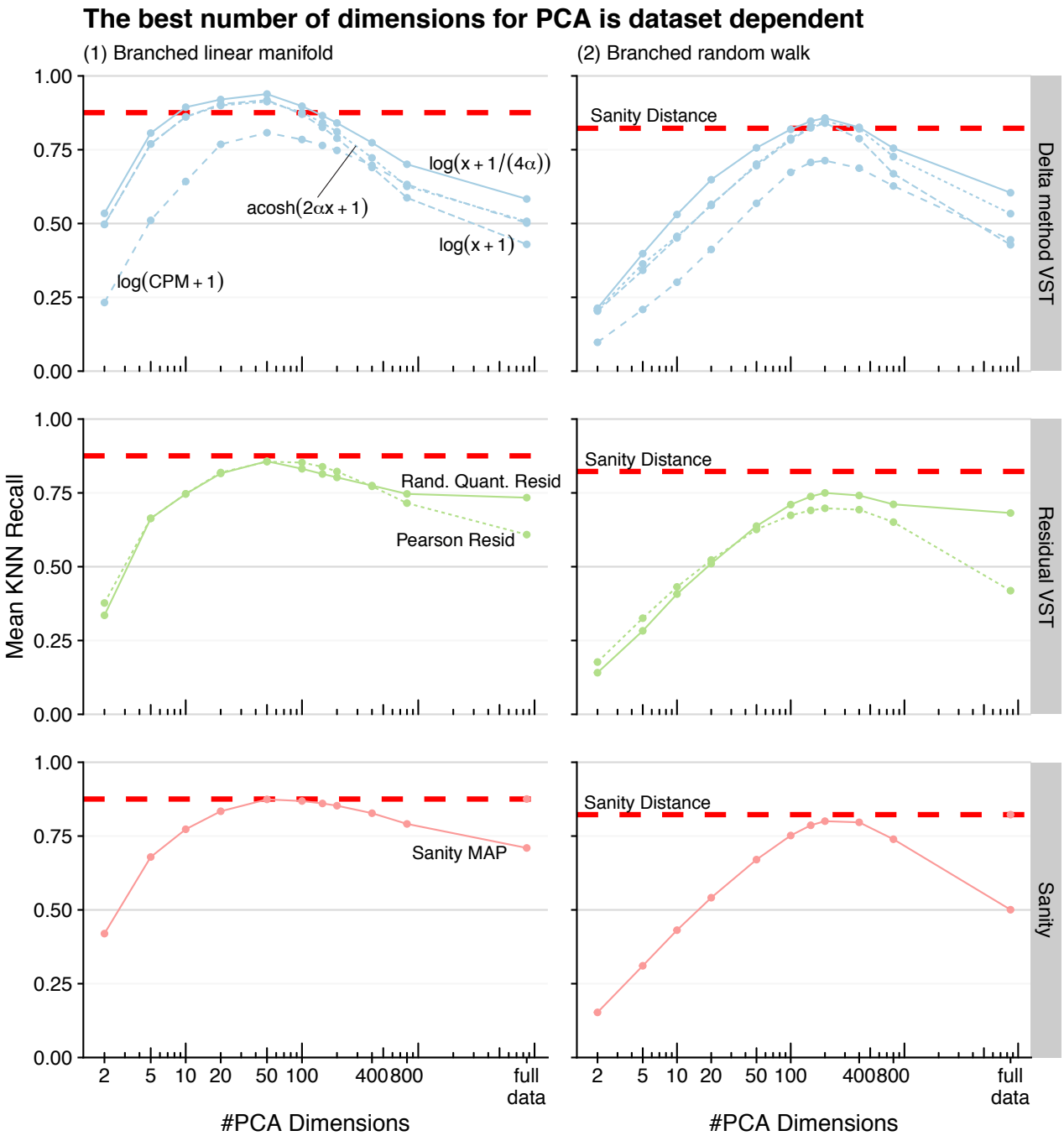
This figure was inspired by Lun (2020). The x in the transformations refers to the size-factor s normalized counts $x = y/s$.



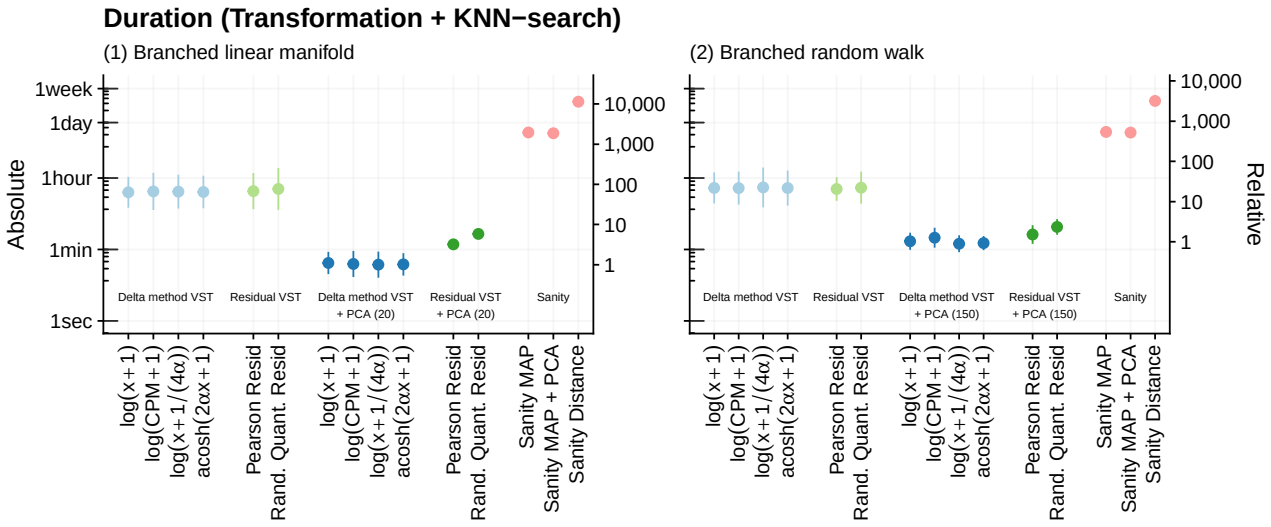
Suppl. Figure S3: The variance per gene after applying different transformations on a cell line dataset. Each point is one gene. The x-axis shows the logarithmized mean of the raw counts across cells; the y-axis shows the variance across cells after transforming the raw counts. For Sanity Distance, we drew one random value from a normal distribution with the logarithm of the gene activities as the mean and the standard deviation representing the width of the posterior to account for the probabilistic nature of Sanity's output. We sampled 1,000 cells and 1,852 genes from the NIH/3T3 mouse cell line dataset (10X Genomics, 2018). We chose the genes so that they uniformly cover the log expression space. We set $\alpha = 0.12$ using the estimate from Suppl. Fig. S1. The horizontal line highlights the target variance of 1. Points outside the range of the respective y-axes are plotted at the top edge of the plot.



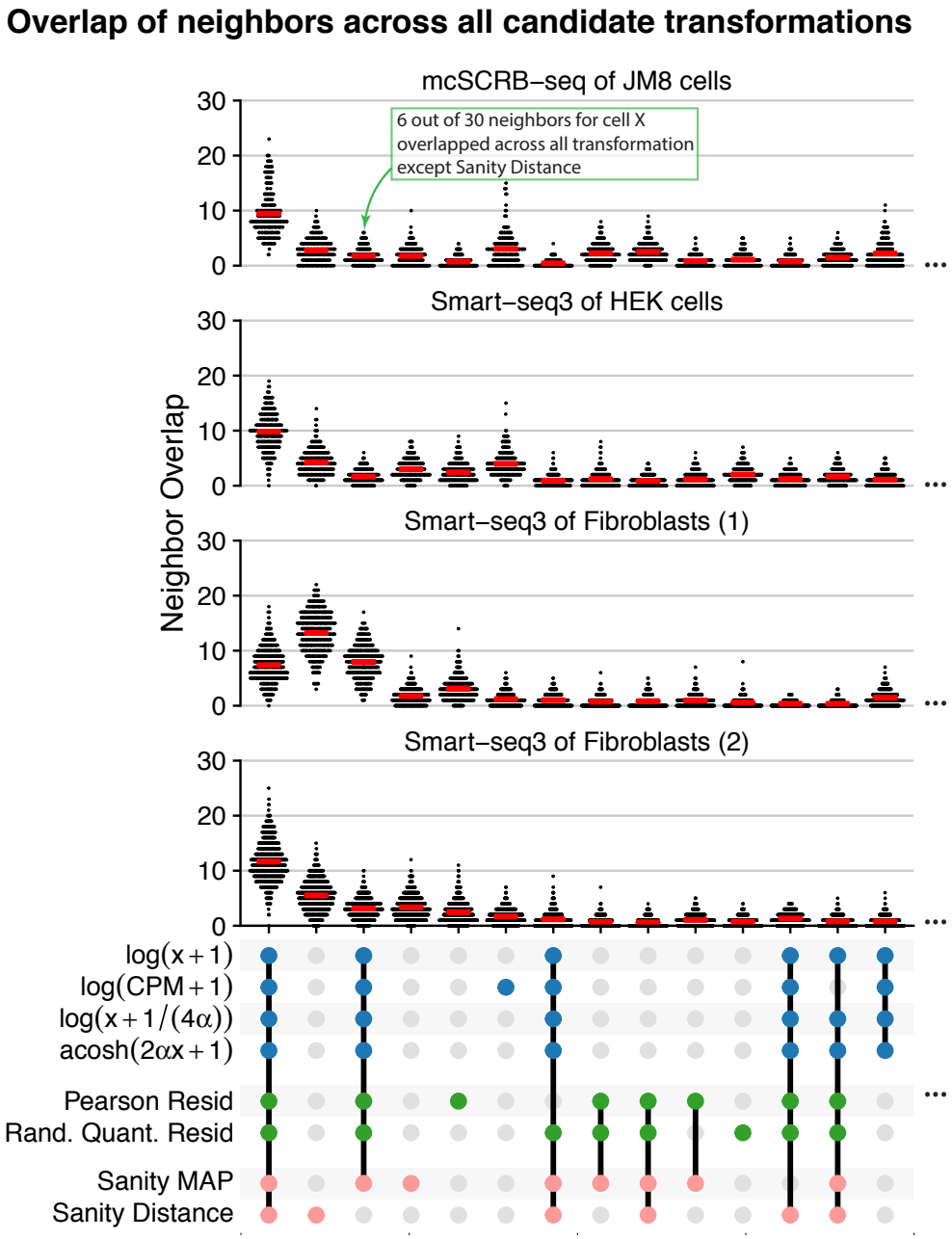
Suppl. Figure S4: Schematic representation of how randomized quantile residuals are constructed. In the first step, a Gamma-Poisson distribution (black line) is fitted to the observed counts. Then, the quantiles of the Gamma-Poisson distribution are matched with the quantiles of a standard normal distribution 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 S5: Dependence of the k nearest neighbor (KNN) recall on the number of dimensions used for the principal component analysis (PCA) for the different transformations on two simulated branching datasets: (1) each branch is a linear interpolation between two points, (2) each branch is a random walk. Each point is the median performance (mean recall for $k = 100$) across 10 replicates. The performance of the Sanity Distance method was included as a reference (dashed horizontal line); it does not depend on the number of PCA dimensions because it is always fitted on the full data.



Suppl. Figure S6: CPU time of the transformation and k nearest neighbor (KNN) search for the benchmarks in Fig. 3. 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., 23 and 81 seconds on the two datasets, respectively). The point ranges show the mean and standard deviation across 10 replicates.



Suppl. Figure S7: Overlap of neighbors across all candidate transformations applied to the deeply sequenced datasets. Each point in the beeswarm plot represents the intersection pattern indicated on the x-axis for one cell: i.e. how many neighbors of that cell were shared across all eight transformations or some combination of them. The red horizontal line shows the mean overlap per intersection pattern. We limited the x-axis to the 14 most common combinations for ease of visualisation. The first column of each facet are what we call the *common* nearest neighbors.

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

$$c_{\mathrm{v}} = \frac{\sqrt{\mathbb{V}\mathrm{ar}\left[X_i\right]}}{\mathbb{E}[X_i]}. \tag{7}$$

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

$$\mathbb{V}\mathrm{ar}\left[\log\frac{X_1}{X_2}\right] = \mathbb{V}\mathrm{ar}\left[\log X_1\right] + \mathbb{V}\mathrm{ar}\left[\log X_2\right]. \tag{8}$$

We can use the delta method to approximate

$$\mathbb{V}\mathrm{ar}\left[g(X_i)\right] \approx \left(g'(\mathbb{E}[X_i])\right)^2 \mathbb{V}\mathrm{ar}\left[X\right]. \tag{9}$$

The derivative of $\log(x)$ is

$$\frac{d}{dx}\log(x) = \frac{1}{x}. \tag{10}$$

We can now plug Eq. (9) and (10) into Eq. (8) and find that

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\log\frac{X_1}{X_2}\right] &\approx \frac{1}{\mathbb{E}[X_1]^2}\mathbb{V}\mathrm{ar}[X_1] + \frac{1}{\mathbb{E}[X_2]^2}\mathbb{V}\mathrm{ar}[X_2] \\ &= c_{\mathrm{v}1}^2 + c_{\mathrm{v}2}^2.\end{aligned} \tag{11}$$

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 (acosh) transformation from Eq. (1) is defined as

$$\begin{aligned}g(y) &= \frac{1}{\sqrt{\alpha}}\operatorname{acosh}\left(2\alpha y + 1\right) \\ &= \frac{1}{\sqrt{\alpha}}\log\left(2\alpha y + \sqrt{(2\alpha y+1)^2 - 1} + 1\right).\end{aligned} \tag{12}$$

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

$$h(y) = a + b\log(y + c), \tag{13}$$

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

$$\lim_{y\to\infty}\frac{\sqrt{(2\alpha y+1)^2 - 1}}{2\alpha y} = 1 \tag{14}$$

and thus for large y

$$\begin{aligned}g(y) &\approx \frac{1}{\sqrt{\alpha}}\log\left(4\alpha y + 1\right) \\ &= \frac{1}{\sqrt{\alpha}}\log\left(y + \frac{1}{4\alpha}\right) + \frac{\log\left(4\alpha\right)}{\sqrt{\alpha}}\end{aligned} \tag{15}$$

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 demonstrates 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 Y_{ij} are divided by the size factor s_j before applying the transformation: $g(Y_{ij}/s_j)$ (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 Y:

$$\begin{aligned}Y|Q &\sim \mathrm{Poisson}(Q) \\ Q &\sim \mathrm{Gamma}(\mu, \alpha) \\ Y &\sim \mathrm{GammaPoisson}(\mu, \alpha).\end{aligned} \tag{16}$$

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

$$\begin{aligned}\mathbb{V}\mathrm{ar}[Y] &= \mathbb{E}[\mathbb{V}\mathrm{ar}(Y|Q)] + \mathbb{V}\mathrm{ar}[\mathbb{E}(Y|Q)] \\ &= \mu + \alpha\mu^2,\end{aligned} \tag{17}$$

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

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

$$Y'|Q, s \sim \mathrm{Poisson}(sQ), \tag{18}$$

we find that

$$\begin{aligned}\mathbb{V}\mathrm{ar}[Y'] &= \mathbb{E}[\mathbb{V}\mathrm{ar}(Y'|Q)] + \mathbb{V}\mathrm{ar}[\mathbb{E}(Y'|Q)] \\ &= s\mu' + \alpha s^2\mu'^2 \\ &= \mu + \alpha\mu^2\end{aligned} \tag{19}$$

where $\mu = s\mu'$.

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

$$X = Y'/s, \tag{20}$$

we find that

$$\begin{aligned} \mathbb{V}\mathrm{ar}[X] &= \frac{1}{s^2}\mathbb{V}\mathrm{ar}[Y'] \\ &= \frac{1}{s^2}(s\mu' + \alpha s^2 \mu'^2) \\ &= \frac{1}{s}\mu' + \alpha\mu'^2 \end{aligned} \tag{21}$$

The difference between the final line of Eq. (19) and Eq. (21) 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 marker genes	Mouse lung	DropSeq	Angelidis et al. (2019), GEO GSE124872
Basis for simulating the branched linear and random walk datasets	Human pancreas	InDrops	Baron et al. (2016), scRNAseq BioC package
Deeply sequenced datasets for subsampling benchmark	JM8 cells	mcSCRB-seq	Bagnoli et al. (2018), GEO GSE103568
	HEK cells	Smart-seq3	Hagemann-Jensen et al. (2020), ArrayExpress E-MTAB-8735
	Fibroblasts (1)	Smart-seq3	Hagemann-Jensen et al. (2020), ArrayExpress E-MTAB-8735
	Fibroblasts (2)	Smart-seq3	Larsson et al. (2021), ArrayExpress E-MTAB-10148
Mean-variance relation	Klein	InDrops	Klein et al. (2015), CalTech Data Repo entry 1264
	Svensson 1,2	Chromium v1	Svensson et al. (2017), CalTech Data Repo entry 1264
	NIH/3T3	Chromium v3	10X Genomics (2018), CalTech Data Repo entry 1264
	NCI-H1975	Chromium	Tian et al. (2019), sc_mixology7 Github repo
	GM18502	Chromium v2	Osorio et al. (2019), GEO GSE126321
	HEK cells	Smart-seq3	Hagemann-Jensen et al. (2020), ArrayExpress E-MTAB-8735