Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data

Jan Lause[1], Philipp Berens[1,2,3,4], and Dmitry Kobak[1]

[1] *Institute for Ophthalmic Research, University of Tübingen, Germany*
[2] *Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany*
[3] *Bernstein Center for Computational Neuroscience, University of Tübingen, Germany*
[4] *Center for Integrative Neuroscience, University of Tübingen, Germany*
`name.surname@uni-tuebingen.de`

November 30, 2020

Abstract

Standard preprocessing of single-cell RNA-seq UMI data includes normalization by sequencing depth to remove this technical variability, and nonlinear transformation to stabilize the variance across genes with different expression levels. Instead, two recent papers propose to use statistical count models for these tasks: Hafemeister and Satija (2019) recommend using Pearson residuals from negative binomial regression, while Townes et al. (2019) recommend fitting a generalized PCA model. Here, we investigate the connection between these approaches theoretically and empirically, and compare their effects on downstream processing. We show that the model of Hafemeister and Satija (2019) produces noisy parameter estimates because it is overspecified (which is why the original paper employs post-hoc regularization). When specified more parsimoniously, it has a simple analytic solution equivalent to the rank-one Poisson GLM-PCA of Townes et al. (2019). Further, our analysis indicates that per-gene overdispersion estimates in Hafemeister and Satija (2019) are biased, and that the data analyzed in that paper are in fact consistent with constant overdispersion parameter across genes. We then use negative control data without biological variability to estimate the technical overdispersion of UMI counts, and find that across several different experimental protocols, the data suggest very moderate overdispersion. Finally, we argue that analytic Pearson residuals (or, equivalently, rank-one GLM-PCA or negative binomial regression after regularization) strongly outperform standard preprocessing for identifying biologically variable genes, and capture more biologically meaningful variation when used for dimensionality reduction, compared to other methods.

Introduction

The standard preprocessing pipeline for single-cell RNA-seq data includes sequencing depth normalization followed by log-transformation (Luecken and Theis, 2019). The normalization aims to remove technical variability associated with cell-to-cell differences in sequencing depth, whereas the log-transformation is supposed to make the variance of gene counts approximately independent of the mean expression. Two recent papers argue that neither step works very well in practice (Hafemeister and Satija, 2019; Townes et al., 2019). Instead, both papers suggest to model UMI (unique molecular identifier) data with count models, explicitly accounting for the cell-to-cell variation in sequencing depth (defined here as the total UMI count per cell). Hafemeister and Satija (2019) use a negative binomial (NB) regression model, while Townes et al. (2019) propose Poisson generalized principal component analysis (GLM-PCA). These two models are seemingly very different.

Here we show that the model used by Hafemeister and Satija (2019) has a too flexible parametrization, resulting in noisy parameter estimates. As a consequence, the original paper employs post-hoc smoothing to correct for that. We show that a more parsimonious model produces stable estimates even without regularization and is equivalent to a special case of GLM-PCA. We then demonstrate that the estimates of gene-specific overdispersion in the original paper are strongly biased, and further argue

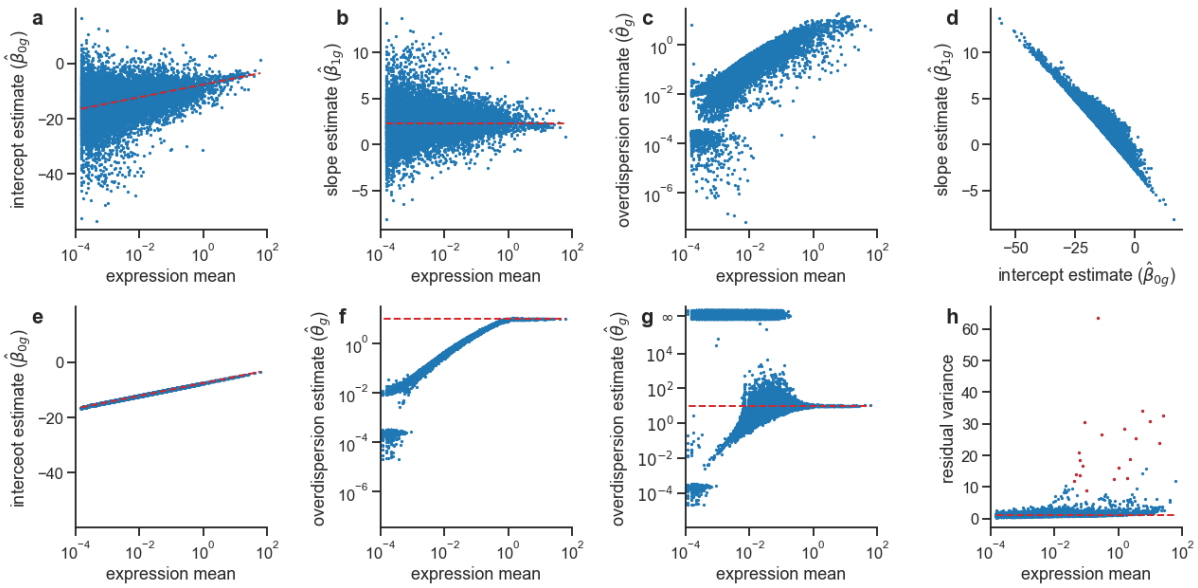

Figure 1: Regression model of Hafemeister and Satija (2019) compared to the offset model. Each single dot corresponds to a model fit to the counts of a single gene in the publicly available 33k PBMC dataset (10x Genomics, $N = 33\,148$ cells). Following Hafemeister and Satija (2019), we included only the 16 809 genes that were detected in at least five cells. Expression mean was computed as $\frac{1}{N}\sum_c X_{cg}$. **a:** Intercept estimates $\hat{\beta}_{0g}$ in the original regression model. Dashed line: Analytical solution for $\hat{\beta}_{0g}$ in the offset model we propose. **b:** Slope estimates $\hat{\beta}_{1g}$. Dashed line: $\beta_{1g} = \ln(10) \approx 2.3$. **c:** Overdispersion estimates $\hat{\theta}_g$. **d:** Relationship between slope and intercept estimates ($\rho = -0.91$). **e:** Intercept estimates in the offset model, where the slope coefficient is fixed to 1. Dashed line shows the analytical solution, which is a linear function of gene mean. **f:** Overdispersion estimates $\hat{\theta}_g$ on simulated data with true $\theta = 10$ (dashed line) for all genes. **g:** Overdispersion estimates $\hat{\theta}_g$ on the same simulated data as in panel (f), but now with 100 instead of 10 iterations in the `theta.ml()` optimizer (R, `MASS` package). Cases for which the optimization diverged to infinity or resulted in spuriously large estimates ($\hat{\theta}_g > 10^6$) are shown at $\hat{\theta}_g = \infty$. Dashed line: true value $\theta = 10$. **h:** Variance of Pearson residuals in the offset model. The residuals were computed analytically, assuming $\theta = 100$ for all genes. Following Hafemeister and Satija (2019), we clipped the residuals to a maximum value of $\sqrt{N}$. Dashed line indicates unit variance. Red dots show the genes identified in the original paper as most variable.

that UMI data do not require gene-specific overdispersion parameters. Rather, the data are consistent with the same overdispersion parameter value shared between all genes. We use available negative control datasets to estimate this technical overdispersion. Furthermore, we compare Pearson residuals, GLM-PCA, and variance-stabilizing transformations (Wagner, 2019, 2020) for feature selection and downstream processing.

The code in Python is available at `http://github.com/berenslab/umi-normalization`.

Results

The regression model in Hafemeister and Satija (2019) is overspecified

Hafemeister and Satija (2019) use the 33k PBMC (peripheral blood mononuclear cells, an immune cell class that features several distinct subpopulations) dataset for their analysis. For each gene g in this dataset, the authors fit an independent NB regression that models the UMI counts X_{cg} of each cell c as

$$X_{cg} \sim \mathrm{NB}(\mu_{cg}, \theta_g) \tag{1}$$

$$\ln(\mu_{cg}) = \beta_{0g} + \beta_{1g} \log_{10}(n_c). \tag{2}$$

Here θ_g is the gene-specific overdispersion parameter, and the NB variance is given by $\mu_{cg} + \mu_{cg}^2/\theta_g$, with $\theta_g \to \infty$ corresponding to the Poisson distribution). Further, $n_c = \sum_g X_{cg}$ is the sequencing depth of cell c, and β_{0g} and β_{1g} are the gene-specific intercept and slope. The natural logarithm follows from the logarithmic link function that is used in NB regression by default. The original paper estimates β_{0g} and β_{1g} using Poisson regression, and then uses the obtained estimates to find the maximum likelihood

estimate of θ_g. The resulting estimates for each gene in the 33k PBMC dataset are shown in Figure 1a–c, reproducing Figure 2A from the original paper.

The authors observed that the estimates $\hat{\beta}_{0g}$ and $\hat{\beta}_{1g}$ were unstable and showed high variance for genes with low average expression (Figure 1a–b). They addressed this by a 'regularization' procedure that re-set all estimates to the local kernel average estimate for a given expression level. The kernel smoothing they employed resulted in an approximately linear increase of the intercept with the average gene expression and an approximately constant slope value of $\hat{\beta}_{1g} \approx 2.3$ (Figure 2A in the original paper). The nature of these dependencies was left unexplained. Moreover, we found that $\hat{\beta}_{0g}$ and $\hat{\beta}_{1g}$ were strongly correlated ($\rho = -0.91$), especially for weakly expressed genes (Figure 1d). Together, these clear symptoms of overfitting suggest that the original model was overspecified.

Indeed, the theory calls for a less flexible model as well. A common modeling assumption for UMI data without biological variability is that each gene takes up a certain fraction p_g of the total amount of UMI counts (Lopez et al., 2018; Svensson et al., 2020; Townes et al., 2019), i.e. for a given gene g, the predicted value μ_{cg} should be modeled as $\mu_{cg} = p_g n_c$, or equivalently, as

$$\ln(\mu_{cg}) = \ln(p_g) + \ln(n_c) = \beta_{0g} + \ln(n_c). \tag{3}$$

Under this assumption, the slope β_{1g} does not need to be fit at all and should be fixed to 1, if $\ln(n_c)$ is used as predictor. Not only does this suggest an alternative, simpler parametrization of the model, but it also explains why Hafemeister and Satija (2019) found that $\hat{\beta}_{1g} \approx 2.3$: they used $\log_{10}(n_c) = \ln(n_c)/\ln(10)$ instead of $\ln(n_c)$ as predictor, and so obtained $\ln(10) \approx 2.3$ as the average slope.

Under the assumption of Eq. (3), a Poisson or NB regression model should be specified using $\ln(n_c)$ as predictor with a fixed slope of 1, a so-called *offset* (Eqs. (1) and (3)). This way, the resulting model has only one free parameter and is not overspecified. In fact, fitting a model with the offset term on the PBMC data resulted in intercept estimates that showed zero variance and were a linear function of the log-expression (Figure 1e). This is because the offset model has an analytical maximum likelihood solution

$$\hat{\beta}_{0g} = \ln\left(\textstyle\sum_c X_{cg} / \sum_c n_c\right) = \ln(\tfrac{1}{N}\textstyle\sum_c X_{cg}) - \ln(\tfrac{1}{N}\textstyle\sum_c n_c), \tag{4}$$

which forms a straight line when plotted against the log-scaled average gene count $\frac{1}{N}\sum_c X_{cg}$ (Figure 1e). This provides an explanation for the linear trend in $\hat{\beta}_{0g}$ in the original two-parameter model of Hafemeister and Satija (2019) (Figure 1a).

This solution can be equivalently rewritten in a simple symmetric form as

$$\hat{\mu}_{cg} = \hat{p}_g n_c = \left(\frac{\sum_i X_{ig}}{\sum_{i,j} X_{ij}}\right) \sum_j X_{cj} = \frac{\sum_i X_{ig} \cdot \sum_j X_{cj}}{\sum_{i,j} X_{ij}}. \tag{5}$$

Note that it holds for both Poisson and NB regressions with offset terms. In practice, our one-parameter offset model and the original two-parameter model after regularization arrive at qualitatively similar predictions (see Figure 1h below). However, we argue that the one-parameter model is more appealing from a theoretical perspective, has an analytic solution, and does not require post-hoc averaging of the coefficients across genes.

The offset model is equivalent to the rank-one GLM-PCA

The offset model turns out to be a special case of the GLM-PCA method developed by Townes et al. (2019). There, the UMI counts are modeled as

$$X_{cg} \sim \text{Poisson}(\mu_{cg}) \text{ or } \text{NB}(\mu_{cg}, \theta_g) \tag{6}$$

$$\mu_{cg} = n_c \exp\left(\sum_{l=0}^{k} U_{cl} V_{lg}\right) = n_c \exp\left(V_{0g} + \sum_{l=1}^{k} U_{cl} V_{lg}\right), \tag{7}$$

assuming $k+1$ latent factors, with U and V playing the role of principal components and corresponding eigenvectors in standard PCA. Importantly, the first latent factor is constrained to $U_{c0} = 1$ for all cells c, such that V_{0g} can be interpreted as gene-specific intercepts. If the data are modeled without any further latent factors, Eq. (7) can be simplified as

$$\ln(\mu_{cg}) = V_{0g} + \ln(n_c), \tag{8}$$

which is identical to Eq. (3) with $V_{0g} = \beta_{0g}$. This shows that the proposed one-parameter offset model becomes exactly equivalent to the intercept-only rank-one GLM-PCA.

Overdispersion estimates in Hafemeister and Satija (2019) are biased

After discussing the overparametrization of the systematic component of the model, we now turn to the NB noise model employed by Hafemeister and Satija (2019). The $\hat{\theta}_g$ estimates in the original paper are monotonically increasing with the average gene expression, both before and after smoothing (Figure 1c). This suggests that there is a biologically meaningful relationship between the expression strength and the overdispersion parameter θ_g. However, this conclusion is in fact unsupported by the data.

To demonstrate this, we simulated a dataset with NB-distributed counts $\widetilde{X}_{cg} \sim \mathrm{NB}(\mu_{cg}, \theta)$ with μ_{cg} given by Eq. (5), $\hat{p}_g$ and n_c taken from the PBMC data, and θ set to 10 for all genes. Applying the original estimation procedure to this simulated dataset showed the same positive correlation of $\hat{\theta}_g$ with the average expression as in real data (Figure 1f), strongly suggesting that it does not represent an underlying technical or biological cause but only the estimation bias. Low-expressed genes had a larger bias and only for genes with the highest average expression we could estimate θ correctly.

Moreover, the $\hat{\theta}_g$ estimates strongly depended on the exact details of the estimation procedure. Using the `theta.ml()` R function with its default 10 iterations (as the original paper did) led to multiple convergence warnings for the simulated data in Figure 1f. Increasing this maximum number of iterations to 100 eliminated most convergence warnings, but caused 49.9% of the estimates to diverge to infinity or above 10^{10} (Figure 1g). These instabilities are likely due to shallow maxima in the NB likelihood w.r.t. θ (Willson et al., 1986).

The above arguments show that the overdispersion parameter estimates in Hafemeister and Satija (2019) for genes with low expression were strongly biased. In practice, however, the predicted variance is only weakly affected by the exact value of θ for low expression means, and so the bias reported here does not substantially affect the Pearson residuals (see below). We note that large errors in NB overdispersion estimates have been extensively described in other fields, with simulation studies showing that estimation bias occurs especially for low NB means, small sample sizes and large true values of θ (Clark and Perry, 1989; Lord, 2006; Lord and Miranda-Moreno, 2008) (i.e. for samples that are close to the Poisson distribution).

Negative control datasets suggest low overdispersion

To avoid noisy and biased estimates, we suggest to use one common θ value shared between all genes. In fact, rather than estimating this value from a biologically heterogeneous dataset such as PMBC, it is more appropriate to estimate the technical overdispersion using negative control datasets, collected without any biological variability (Svensson, 2020). We analyzed several such datasets spanning different droplet- and plate-based sequencing protocols (10x Genomics, inDrop, MicrowellSeq) and compared the $\hat{\theta}_g$ estimates to the estimates obtained using simulated NB data with various known values of $\theta \in \{10, 100, 1000, \infty\}$. For the simulations, we used the empirically observed sample sizes and sequencing depths. We found that across different protocols, negative control data were consistent with overdispersion $\theta \approx 100$ or larger (Figure S1). The plateau at $\theta \approx 10$ in the PBMC data visible in Figure 1c could reflect biological and not technical variability. At the same time, negative control data were not consistent with the Poisson model ($\theta = \infty$), but likely overdispersion parameter values ($\theta \approx 100$) are large enough to make the Poisson model acceptable in practice.

Analytic Pearson residuals work well for identifying variable genes

Both Hafemeister and Satija (2019) and Townes et al. (2019) suggest to use Pearson (or deviance) residuals based on models that only account for technical variability, in order to identify biologically variable genes. Indeed, genes showing biological variability should have higher variance than predicted by such a model. Pearson residuals in the offset model (or, equivalently, rank-one GLM-PCA) can be conveniently written in closed form:

$$Z_{cg} = \frac{X_{cg} - \hat{\mu}_{cg}}{\sqrt{\hat{\mu}_{cg} + \hat{\mu}_{cg}^2/\theta}}, \quad \hat{\mu}_{cg} = \frac{\sum_j X_{cj} \cdot \sum_i X_{ig}}{\sum_{i,j} X_{ij}}, \quad \theta = 100. \tag{9}$$

Computing the variance of Pearson residuals for the PBMC data using Eq. (9) is, up to a constant, equivalent to computing the Pearson χ^2 goodness-of-fit statistic (Agresti, 2015). Using it for gene selection yielded a very similar result to the one presented by Hafemeister and Satija (2019), with almost the same set of genes identified as biologically variable (Figure 1h, cf. Figure 4C in the original paper).

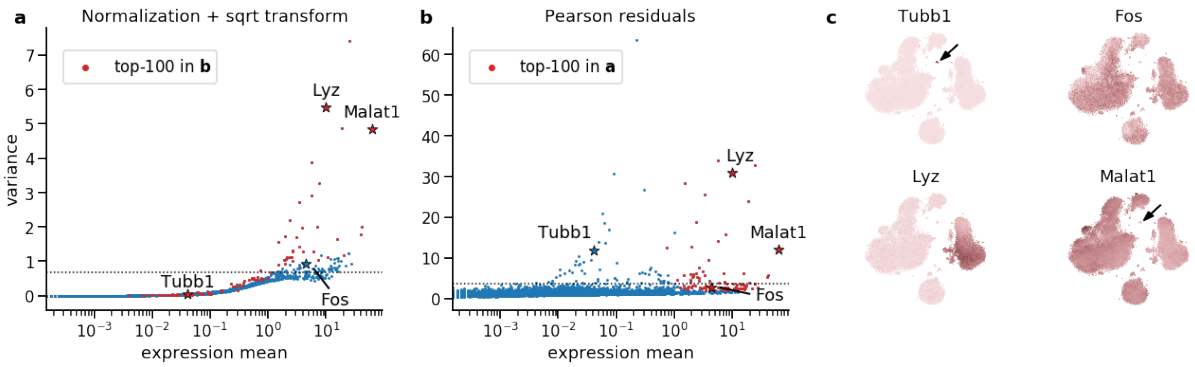

Figure 2: Selection of variable genes. In the first two panels, each dot shows the variance of a single gene in the PBMC dataset after applying a normalization method. The dotted horizontal line shows a threshold adjusted to select 100 most variable genes. Red dots mark 100 genes that are selected by the other method, i.e. that are above the threshold in the other panel. Stars indicate genes shown in the last panel. **a:** Gene variance after sequencing depth normalization, median-scaling, and the square-root transformation. **b:** Variance of Pearson residuals (assuming $\theta = 100$). **c:** t-SNE of the entire PBMC dataset (see Figure S4), colored by expression of four example genes (after sequencing depth normalization and square-root transform). Platelet marker *Tubb1* with low average expression is only selected by Pearson residuals. Arrows indicate the platelet cluster. *Fos* is only selected by the square root-based method, and does not show a clear type-specific expression pattern. *Malat1* (expressed everywhere apart from platelets) and monocyte marker *Lyz* with higher average expression are selected by both methods.

For most genes, the variance of the Pearson residuals was close to 1, indicating that this model predicted the variance of the data correctly and suggesting that most genes did not show biological variability. Using $\theta = 100$ led to several high-expression genes selected as biologically variable that would not be selected with a lower θ (e.g. *Malat1*), but overall, using $\theta = 10$, $\theta = 100$, or even the Poisson model with $\theta = \infty$ led to only minor differences. This demonstrates that in practice, although it is more parsimonious and unregularized, the offset model yields similar results to the regularized NB regression of Hafemeister and Satija (2019).

It is instructive to compare the variance of Pearson residuals to the variance that one gets after explicit sequencing depth normalization followed by a variance-stabilizing transformation. For Poisson data, the square root transformation $\sqrt{x}$ is approximately variance-stabilizing, and several modifications exist in the literature (Bar-Lev and Enis, 1988), such as the Anscombe transformation $2\sqrt{x + 3/8}$ (Anscombe, 1948) and the Freeman-Tukey transformation $\sqrt{x} + \sqrt{x+1}$ (Freeman and Tukey, 1950). Normalizing UMI counts X_{cg} by sequencing depths n_c (and multiplying the result by the median sequencing depth $\langle n_c \rangle$ across all cells) followed by one of the square-root transformations has been advocated for UMI data processing (e.g. Wagner, 2019, 2020).

Comparing the gene variances after the square-root transformation (Figure 2a) with the variances of Pearson residuals (Figure 2b) in the PBMC dataset showed that the square-root transformation is not sufficient. In particular, low-expression genes have variance close to zero after the square-root transform. For example, platelet gene markers such as *Tubb1* have low average expression (because platelets are a rare population in the PBMC dataset) and do not show high variance after any kind of square-root transform (another example was given by the B-cell marker *Cd79a*). At the same time, Pearson residuals correctly indicate that these genes have high variance and are biologically meaningful (Figure 2c). For the genes with higher average expression, some differentially expressed genes like the monocyte marker *Lyz* or the above-mentioned *Malat1* showed high variance in both approaches. However, the selection based on the square-root transform also included genes like *Fos*, which showed noisy and biologically unspecific expression patterns (Figure 2c). Similar patterns were observed in the retinal dataset of Macosko et al. (2015) (Figure S2). Applying the widely-used $\log(1 + x)$-transform instead of the square-root transform led to similar results with many biologically meaningful genes not being selected (Figure S3).

The gene with the highest average expression in the PBMC dataset, *Malat1*, showed clear signs of biologically meaningful variability (e.g. it is not expressed in platelets) (Figure 2c). While this gene is selected as biologically variable based on Pearson residuals with $\theta \approx 100$ as we propose (Figure 2b), it was not selected by Hafemeister and Satija (2019) who used $\theta \approx 10$ (Figure 1c,h). This again suggests that $\theta \approx 100$ is more appropriate than $\theta \approx 10$ to model technical variability of UMI counts.

Pearson residuals are a good basis for dimensionality reduction

Finally, we studied the effect of the three normalization approaches on t-SNE embeddings. The first approach is sequencing depth normalization and median scaling, followed by the variance-stabilizing square-root transform (Wagner, 2019, 2020). We used 50 principal components of the resulting data matrix to construct a t-SNE embedding. The second approach is computing Pearson residuals according to Eq. (9) with $\theta = 100$, followed by PCA reduction to 50 components. The third approach is computing 50 components of negative binomial GLM-PCA with $\theta = 100$ (Townes et al., 2019). We used the same initialization to construct t-SNE embeddings to ease the visual comparison (Kobak and Berens, 2019).

We applied these methods to the full PBMC dataset (Figure S4) and to three retinal datasets (Macosko et al., 2015; Shekhar et al., 2016; Tran et al., 2019) (Figure S5). The resulting embeddings were mostly similar, suggesting comparable performance between methods. Hafemeister and Satija (2019) argued that using Pearson residuals reduces the amount of variance in the embedding explained by the sequencing depth variation, compared to sequencing depth normalization and log-transformation. We argue that this effect was mostly due to the large factor (10^4) that the authors used for re-scaling the counts after normalization (Figure S6): large scaling factors or small pseudocounts are known to introduce spurious variation into the distribution of normalized counts (Lun, 2018; Townes et al., 2019). For the PMBC dataset, all three of our t-SNE embeddings showed similar amount of sequencing depth variation across the embedding space (Figure S7).

However, on closer inspection embeddings based on Pearson residuals consistently outperformed the other two. For example, while the Pearson residual embeddings clearly separated fine cell types in the Macosko et al. (2015) dataset, the square-root embedding mixed some of them (we observed the same when using the log-transform). For the same dataset, GLM-PCA embedding did not fully separate some of the biologically distinct cell types. Furthermore, GLM-PCA embeddings often featured Gaussian-shaped blobs with no internal structure (Figure S5), suggesting that some fine manifold structure was lost, possibly due to convergence difficulties.

These methods differ in both space and time computational complexity. UMI count data are typically very sparse (e.g. in the PBMC dataset, 95% of entries are zero) and can be efficiently stored as a sparse matrix object. Sequencing depth normalization and square-root transformation do not affect the zeros, preserving the sparsity of the matrix, and PCA can be run directly on a sparse matrix. In contrast, Pearson residuals form a dense matrix without any zeros, and so can take a large amount of memory to store (4.5 Gb for the PBMC dataset). For larger datasets with several hundreds of thousands of cells this can become prohibitive. In contrast, GLM-PCA can run directly on a sparse matrix but takes a long time to converge (Table S1), possibly becoming prohibitively slow for bigger datasets.

Computational complexity can be greatly reduced if gene selection is performed in advance. We selected 1000 genes with the largest Pearson residual variance in the PMBC dataset (note that this variance can be computed per gene, without storing the entire residual matrix in memory) and found that gene selection did not noticeably affect the embedding quality (Figure S4). After gene selection, Pearson residuals do not require a lot of memory (0.3 Gb) and so can be conveniently used. GLM-PCA, however, remained slow even after gene selection (Table S1, 4 h vs. 5 s for Pearson residuals).

Conclusions

We have reviewed and contrasted three different methods for normalization of UMI count data. We showed that without post-hoc regularization, the negative binomial regression model of Hafemeister and Satija (2019) exhibits high variance in its parameter estimates because it is overspecified, which is why it had to be regularized. We argued that instead of regularizing an overspecified model, one should resort to a more parsimonious and theoretically motivated model specification involving an offset term. This made the model equivalent to the rank-one GLM-PCA of Townes et al. (2019) and yielded a simple analytic solution. Further, we showed that the estimates of per-gene overdispersion parameter θ_g in the original paper exhibit substantial bias and that the data are in fact consistent with a single θ value for all genes. We used negative control datasets from different experimental protocols to show that UMI counts have low overdispersion and are well described by $\theta \approx 100$ shared across genes.

We found that the approach developed by Hafemeister and Satija (2019) and implemented in the R package `sctransform` in practice often yields Pearson residuals that are very similar to our analytic residuals with fixed overdispersion parameter (cf. Figure 1h and Figure 4C in the original paper). Here we argued that the offset model with its analytic solution is attractive for reasons of parsimony, theoretical simplicity, and computational speed. Moreover, it provided an explanation for the linear trends in the

smoothed estimates in the original paper. Analytic residuals from the offset model, as well as fixed overdispersion parameter, can be easily integrated as options into `sctransform` as well as into other bioinformatics libraries.

We argued that using Pearson residuals strongly outperforms naïve application of the variance-stabilizing transformation for the purpose of selecting biologically variable genes. Also, we demonstrated that Pearson residuals produce better-quality 2D embeddings than both GLM-PCA and the square-root transform. Applying gene selection prior to dimensionality reduction reduces the computational cost of using Pearson residuals down to negligible. In contrast, GLM-PCA model can be over 1000× longer to fit. Therefore, we conclude that analytic Pearson residuals is a theory-based, fast, and convenient method for normalization of UMI datasets.

Acknowledgements We thank William Townes, Christoph Hafemeister, and Rahul Satija for discussions and helpful comments.

This work was supported by the Deutsche Forschungsgemeinschaft (BE5601/4-1, BE5601/6-1, and EXC 2064 ML, project number 390727645), by the German Ministry of Education and Research (01GQ1601 and 01IS18039A), and by the National Institute Of Mental Health of the National Institutes of Health under Award Number U19MH114830. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Competing interests The authors declare that they have no competing interests.

Availability of data and materials All used datasets are publicly available. The Python code to reproduce the analysis of this paper will be available at http://github.com/berenslab/umi-normalization, including download instructions for the used datasets.

References

Alan Agresti. *Foundations of linear and generalized linear models*. John Wiley & Sons, 2015.

Francis J Anscombe. The transformation of Poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.

Shaul K Bar-Lev and Peter Enis. On the classical choice of variance stabilizing transformations and an application for a Poisson variate. *Biometrika*, 75(4):803–804, 1988.

Suzanne J Clark and Joe N Perry. Estimation of the negative binomial parameter κ by maximum quasi-likelihood. *Biometrics*, pages 309–316, 1989.

Murray F Freeman and John W Tukey. Transformations related to the angular and the square root. *The Annals of Mathematical Statistics*, pages 607–611, 1950.

Christoph Hafemeister and Rahul Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20:296, 2019.

Xiaoping Han, Renying Wang, Yincong Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Fang Ye, et al. Mapping the mouse cell atlas by microwell-seq. *Cell*, 172 (5):1091–1107, 2018.

Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015.

Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature Communications*, 10(1):1–14, 2019.

George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature methods*, 16(3): 243–245, 2019.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018.

Dominique Lord. Modeling motor vehicle crashes using Poisson-gamma models: Examining the effects of low sample mean values and small sample size on the estimation of the fixed dispersion parameter. *Accident Analysis & Prevention*, 38(4):751–766, 2006.

Dominique Lord and Luis F Miranda-Moreno. Effects of low sample mean values and small sample size on the estimation of the fixed dispersion parameter of Poisson-gamma models for modeling motor vehicle crashes: A bayesian perspective. *Safety Science*, 46(5):751–770, 2008.

Malte D Luecken and Fabian J Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6), 2019.

Aaron Lun. Overcoming systematic errors caused by log-transformation of normalized single-cell rna sequencing data. *BioRxiv*, page 404962, 2018.

Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Karthik Shekhar, Sylvain W Lapan, Irene E Whitney, Nicholas M Tran, Evan Z Macosko, Monika Kowalczyk, Xian Adiconis, Joshua Z Levin, James Nemesh, Melissa Goldman, et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166(5):1308–1323, 2016.

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

Valentine Svensson, Kedar Nath Natarajan, Lam-Ha Ly, Ricardo J Miragaia, Charlotte Labalette, Iain C Macaulay, Ana Cvejic, and Sarah A Teichmann. Power analysis of single-cell rna-sequencing experiments. *Nature Methods*, 14(4):381–387, 2017.

Valentine Svensson, Adam Gayoso, Nir Yosef, and Lior Pachter. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics*, 2020.

F William Townes, Stephanie C Hicks, Martin J Aryee, and Rafael A Irizarry. Feature selection and dimension reduction for single-cell RNA-seq based on a multinomial model. *Genome Biology*, 20:295, 2019.

Nicholas M Tran, Karthik Shekhar, Irene E Whitney, Anne Jacobi, Inbal Benhar, Guosong Hong, Wenjun Yan, Xian Adiconis, McKinzie E Arnold, Jung Min Lee, et al. Single-cell profiles of retinal ganglion cells differing in resilience to injury reveal neuroprotective genes. *Neuron*, 104(6):1039–1055, 2019.

Florian Wagner. Straightforward clustering of single-cell RNA-Seq data with t-SNE and DBSCAN. *BioRxiv*, page 770388, 2019.

Florian Wagner. Monet: An open-source Python package for analyzing and integrating scRNA-Seq data using PCA-based latent spaces. *bioRxiv*, 2020.

LJ Willson, JL Folks, and JH Young. Complete sufficiency and maximum likelihood estimation for the two-parameter negative binomial distribution. *Metrika*, 33(1):349–362, 1986.

Supplementary Figures

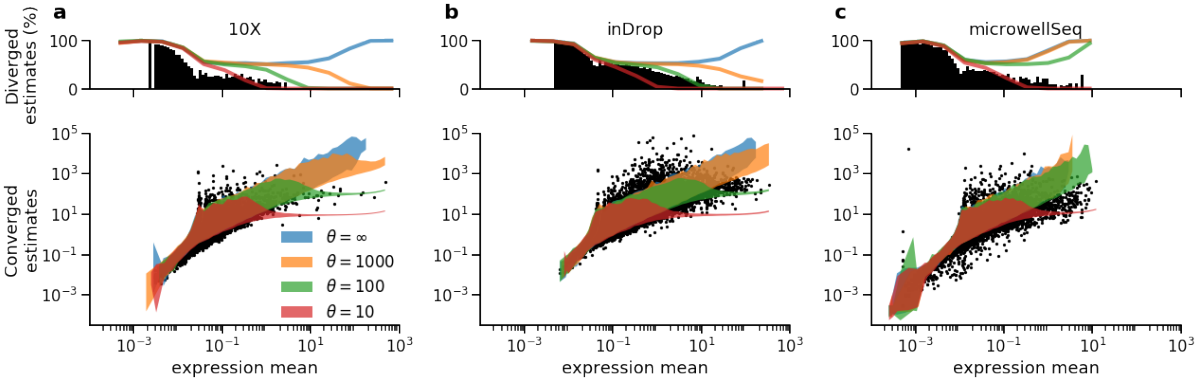

Figure S1: Overdispersion in negative control datasets. In the lower panels, each dot corresponds to one gene g. The overdispersion parameter estimates were obtained with `theta.ml()` using 100 iterations, using the expression means predicted by the offset model. Diverged estimates were clipped as in Figure 1g. Colors show the 5th to 95th percentile regions of the overdispersion estimates for simulated datasets with a known θ shared across genes. For each value of $\theta \in \{10, 100, 1000, \infty\}$, we simulated 1000 datasets according to Eq. (1), using means $\mu_{cg} = p_g n_c$ with empirical n_c and a series of p_g values chosen to cover the entire data range. The upper panels show the fraction of genes for which the estimate diverged (black: original data, colors: simulated data). **a:** 10x Genomics Chromium technical negative control dataset (sample 1), consisting of RNA solution split into droplets (Svensson et al., 2017). 2 000 cells, 13 025 genes. **b:** inDrop technical negative control dataset prepared as in (a) (Klein et al., 2015). 953 cells, 25 025 genes. **c:** microwellSeq biological negative control dataset from non-differentiating embryonic stem cell culture (E14 line from strain 129P2/Ola) (Han et al., 2018). 9 994 cells, 15 069 genes.

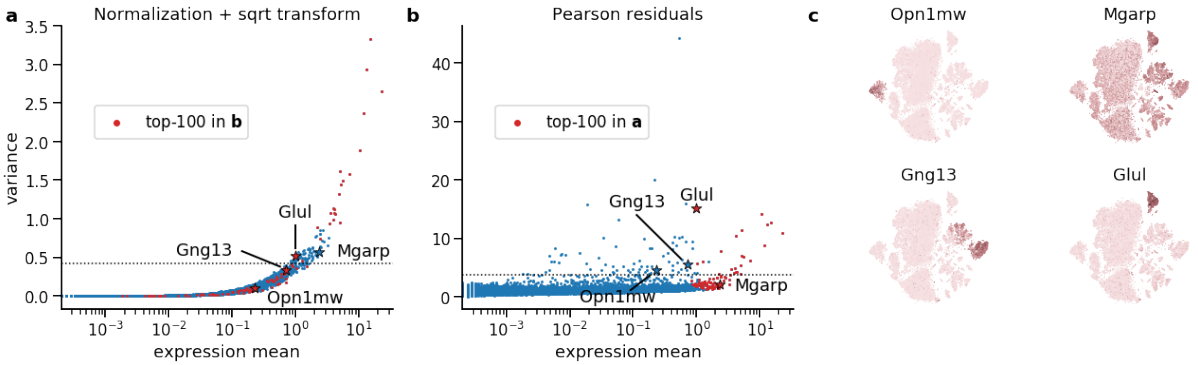

Figure S2: Selection of variable genes for a retinal dataset. All panels as in Figure 2, here showing gene selection for the largest batch of the retina dataset from Macosko et al. (2015). It consists of replicates `p1` and `r4`–`r6`, with a total of $N = 24\,769$ cells, and 17 973 genes detected in five or more cells. The cone photoreceptor marker *Opn1mv* and the bipolar cell marker *Gng13* were only selected when the variance of Pearson residuals was used as criterion. The diffusely expressed gene *Mgarp* is present in many cell types, and was only selected based on the variance of the square-root transformed data. The Müller glia marker *Glul* was selected by both methods.

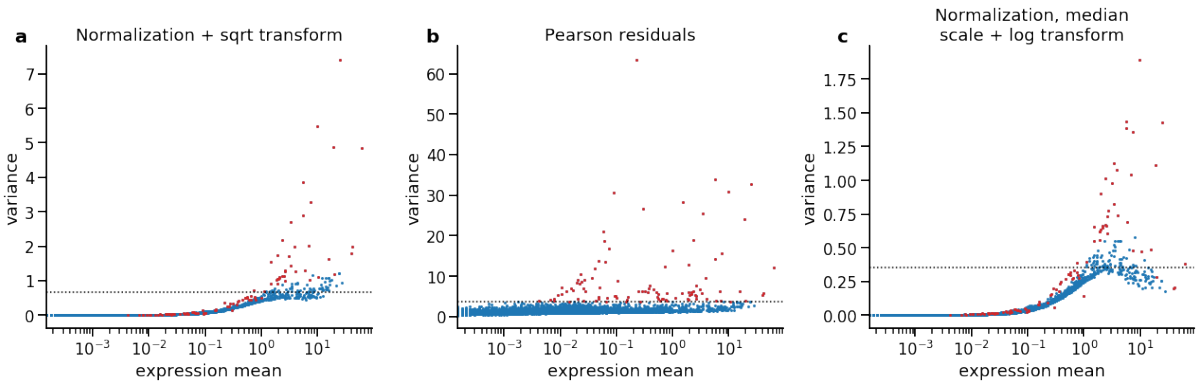

Figure S3: Comparing gene selection criteria. Each dot shows the variance of a single gene in the PMBC dataset after applying a normalization method. The 100 genes with the largest variances lie above the dotted horizontal line. Red dots show the top 100 genes selected by variance of Pearson residuals. **a–b:** Reproduced from Figure 2. **c:** Variance after sequencing depth normalization, median-scaling and $\log(1+x)$-transform. Note that here the variance is a non-monotonic function of the average expression. Note also that some biologically variable genes (red dots) lie below the main cloud, i.e. have *lower* variance than the genes with the same average expression but without biological variability. This happens for genes that are strongly expressed but only in a small subset of cells. Log-transformation acts 'too strong' on these large counts, yielding lower overall variance.

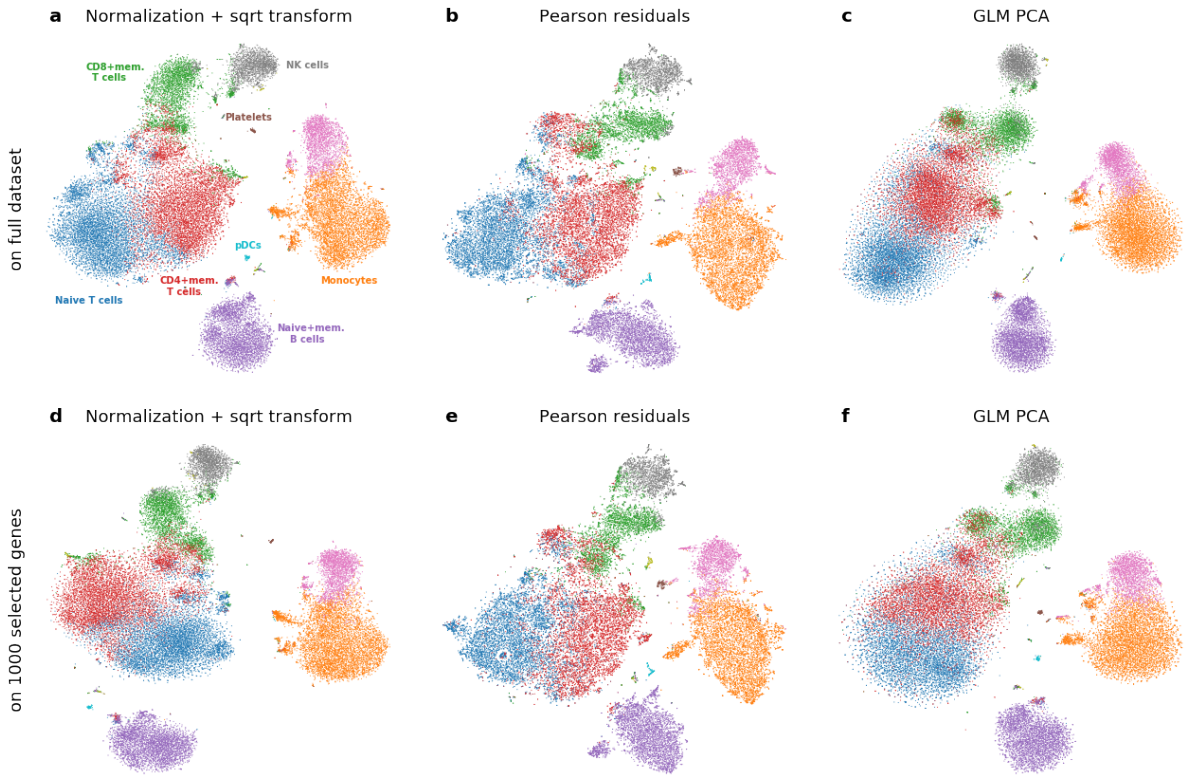

Figure S4: t-SNE embeddings based on different normalization approaches. Each panel shows a t-SNE of the PBMC dataset, based on a different normalization method. All embeddings were computed using `FIt-SNE` (Linderman et al., 2019). All PCAs were computed using `sklearn` (Pedregosa et al., 2011) with default settings. Embeddings in panels a–c were initialized with the first two principal components of the normalized and square-root transformed data to ease the comparison. We used perplexity combination of 30 and $N/100 = 331$ and learning rate $n/12$, following Kobak and Berens (2019). Colors correspond to 10 k-means clusters provided by the 10x Genomics together with the PBMC dataset. We annotated eight clusters based on known marker genes (taken from Wagner (2019)). **a:** The UMI counts after sequencing depth normalization and square-root transform, reduced to 50 principal components. **b:** Pearson residuals assuming $\theta = 100$, reduced to 50 principal components. **c:** 50 components of negative binomial GLM-PCA with $\theta = 100$ (Townes et al., 2019). We used the `glmpca-py` Python implementation (`https://github.com/willtownes/glmpca-py`), using default settings and the option to fix $\theta = 100$ during optimization. The effect of alternative settings on the embedding is shown in Figure S8. **d–f:** Same as a–c, but now we reduced the dimensionality of the dataset by keeping only the 1000 genes with the highest Pearson residual variance. For shared initialization, we used the first two principal components of the square-root transformed data matrix after gene selection. GLM-PCA on the reduced dataset did not converge with default penalty 1, so we increased it to 5, following the automatic tuning procedure used in the authors' `R` implementation.

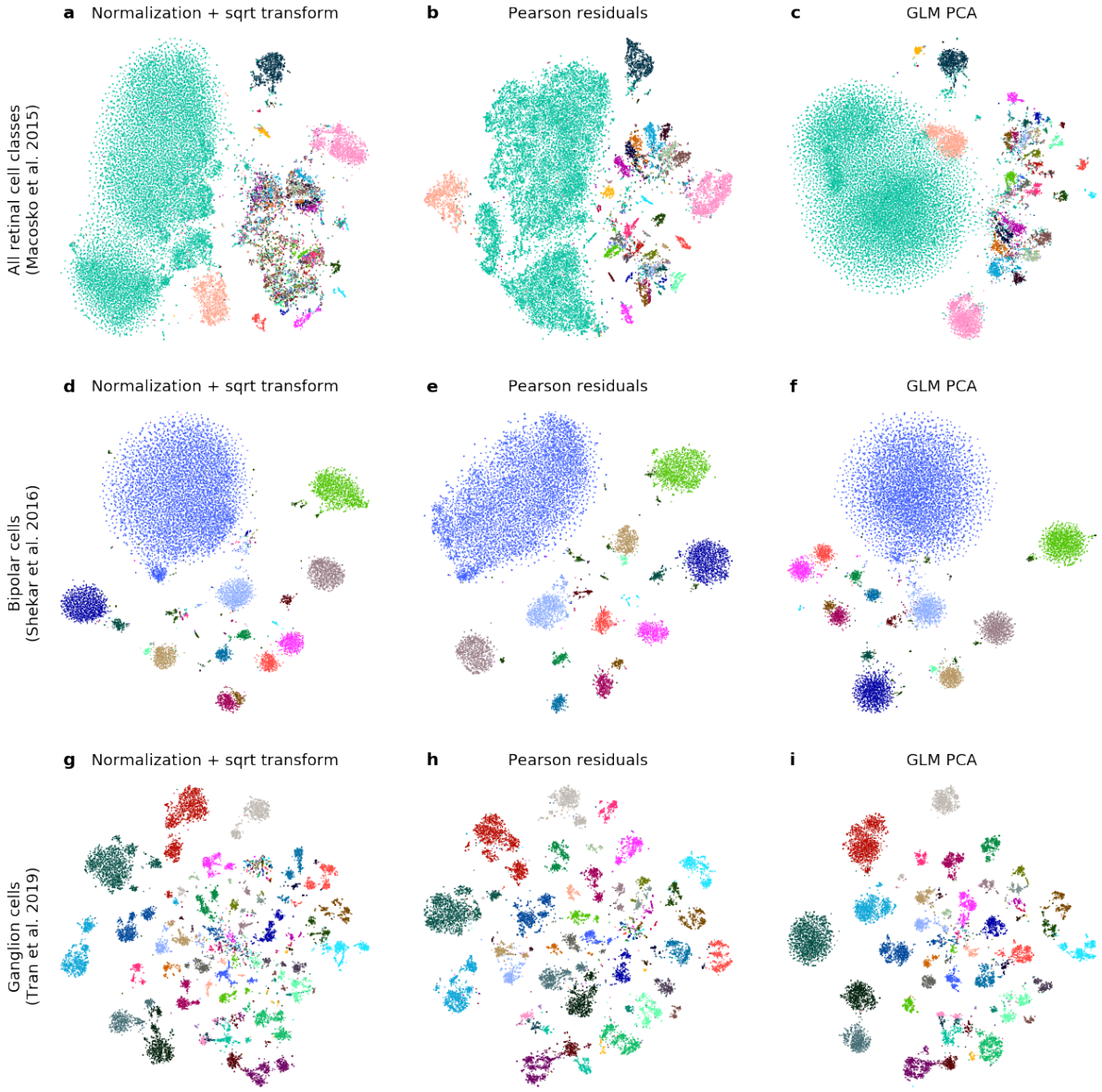

Figure S5: t-SNE embeddings of three retinal datasets after different normalizations. Panels in each of the three rows as in Figure S4, and each row shows a different retinal dataset. For each of them, we only used the largest batch and kept only genes detected in five or more cells. Colors correspond to cell type clusters provided by the original papers. Note that t-SNE embeddings based on depth-normalized counts without any further processing resulted in poor embedding quality across all datasets (not shown). **a–c:** Full-retina dataset (DropSeq) from Macosko et al. (2015), containing all retinal cell types (including glia and vascular cells). 24 769 cells, 17 973 genes. **d–f:** Bipolar cell dataset (DropSeq) from Shekhar et al. (2016). 13 987 cells, 16 520 genes. **g–i:** Retinal ganglion cell dataset (10X v2) from Tran et al. (2019). 15 750 cells, 17 685 genes.

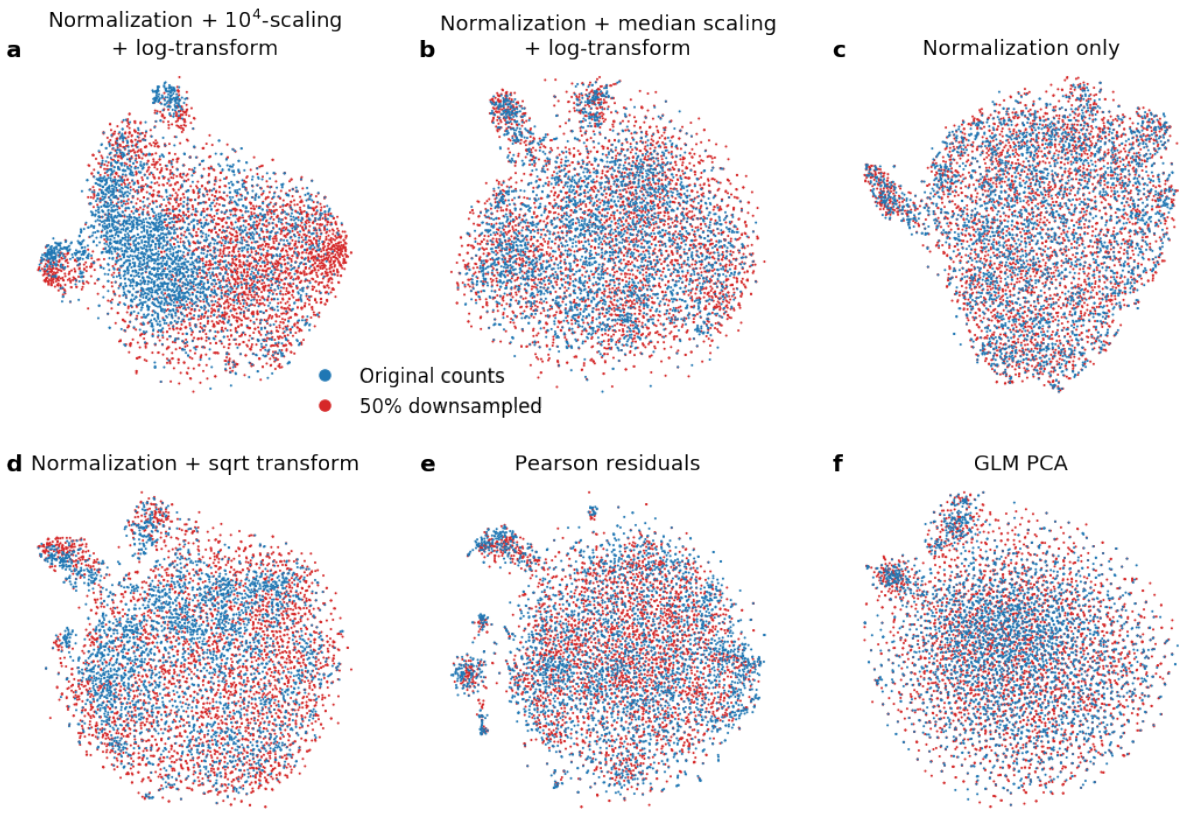

Figure S6: t-SNE embeddings of the monocyte cluster with downsampling. We analyzed the monocyte cluster from the PBMC dataset (marked orange in Figure S4, $n = 6\,307$ cells). Half of the cells (randomly selected) were downsampled to 50% of their original sequencing depth, by simulating new counts $\tilde{X}_{cg} = \text{Binomial}(n = X_{cg}, p = 0.5)$, following the analysis in Hafemeister and Satija (2019) (Figure 6C). We then compared how well different normalization strategies remove the simulated batch-effect. PCA and t-SNE were applied after normalization as in Figure S4. **a:** Depth-normalized counts, scaled with 10^4 and then log-transformed, as in Hafemeister and Satija (2019). Note the strong batch effect due to the large scale. **b:** Depth-normalized counts, scaled with median depth (≈ 1500) and then log-transformed. Note the smaller scale led to a much weaker batch effect compared to (a). **c:** Depth-normalized counts. At the cost of not applying a variance-stabilizing transform, no batch effect remains. **d:** Depth-normalized counts, square-root transformed. A weak batch effect remains. **e:** Pearson residuals, computed with $\theta = 100$. **f:** Negative binomial GLM PCA, computed with fixed $\theta = 100$.

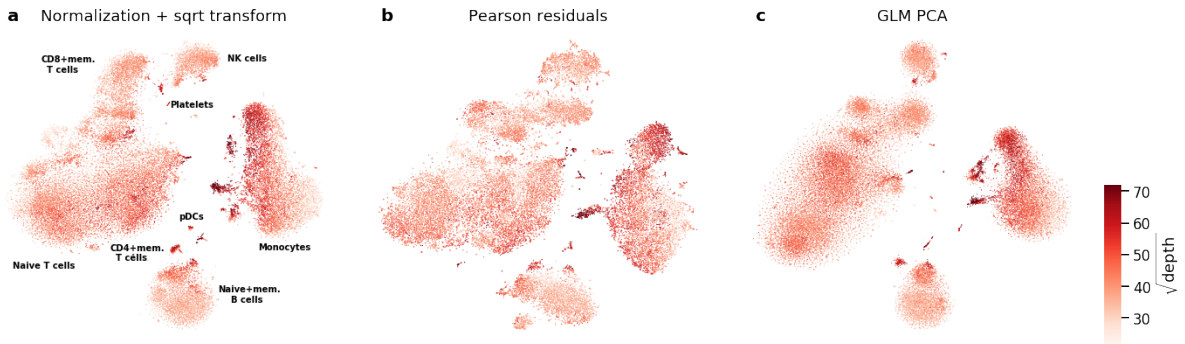

Figure S7: Sequencing depth in t-SNE embeddings based on different normalization approaches. All panels are as in Figure S4 a–c, but overlayed with square-root-transformed sequencing depth. Values above 70 were clipped to 70.

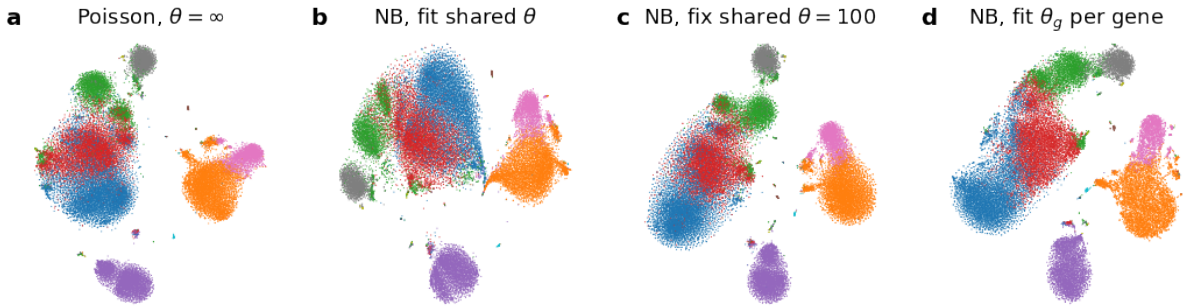

Figure S8: t-SNE embeddings based on GLM PCA with different noise models. All panels show embeddings of the PBMC dataset, colored by clusters as in Figure S4. The GLM PCA was fit using the R implementation by Townes et al. (2019) with default settings unless otherwise stated (`glmpca`, version 0.2.0) **a:** Poisson model, fit with convergence threshold 10^{-8} to avoid early stopping. **b:** NB model with shared overdispersion θ as a free parameter, that was estimated to be $\hat{\theta} = 11.29$. **c:** NB model with fixed overdispersion $\theta = 100$, a setting only available in the Python implementation `glmpca-py`. Same as Figure S4c. **d:** NB model with per-gene overdispersions θ_g as free parameters.

	Normalization, $\sqrt{x}$-transform, PCA	Pearson residuals ($\theta = 100$), PCA	GLM PCA ($\theta = 100$)
Full dataset	30 s	34 s	15 h
After gene selection	4 s	5 s	4 h

Table S1: Runtimes for different normalization pipelines on the PMBC dataset. We ran the three pipelines discussed once on the full PMBC dataset, once after selecting the 1 000 genes with maximum Pearson residual variance. The gene selection itself took 15 seconds. All times given are walltimes from running the code in a docker container on an Ubuntu 18 system with 256 GB RAM and 2×24 CPUs (Xeon Silver 4116 (Dodecacore)). The docker container was restricted to use 30 CPUs at most.

dataset	protocol	cells	genes	species	description
33k PBMC	10X v1	33 148	16 809	human	peripheral blood mononuclear cells
Svensson et al. (2017)	10X v2	2 000	13 025	human	droplets of bulk RNA solution
Klein et al. (2015)	inDrop	953	25 025	human	droplets of bulk RNA solution
Han et al. (2018)	MicrowellSeq	9 994	15 069	mouse	non-differentiating stem cells
Macosko et al. (2015)	DropSeq	24 769	17 973	mouse	retinal cells
Shekhar et al. (2016)	DropSeq	13 987	16 520	mouse	retinal bipolar cells
Tran et al. (2019)	10X v2	15 750	17 685	mouse	retinal ganglion cells

Table S2: Overview of UMI datasets used for analysis. In the Svensson et al. (2017) dataset, we used only sample 1. In the Han et al. (2018) dataset, we used the E14 dataset. In the three retinal datasets, we only used cells from the largest batch. Numbers of genes and cells are after batch selection (where applicable) and exclusion of genes that were detected in fewer than 5 cells.