ZINB-WaVE: A general and flexible method for signal extraction from single-cell RNA-seq data

Davide Risso[1], Fanny Perraudeau[2], Svetlana Gribkova[3], Sandrine Dudoit*[2,4], and Jean-Philippe Vert*[5,6,7,8]

[1]Division of Biostatistics and Epidemiology, Department of Healthcare Policy and Research, Weill Cornell Medicine, New York, NY, USA.
[2]Division of Biostatistics, School of Public Health, University of California, Berkeley, USA
[3]Laboratoire de Probabilités et Modèles Aléatoires, Université Paris Diderot, Paris, France
[4]Department of Statistics, University of California, Berkeley, USA
[5]MINES ParisTech, PSL Research University, CBIO-Centre for Computational Biology, Paris, France
[6]Institut Curie, Paris, France
[7]INSERM U900, Paris, France
[8]Ecole Normale Supérieure, Department of Mathematics and Applications, Paris, France

April 6, 2017

Abstract

Single-cell RNA sequencing (scRNA-seq) is a powerful technique that enables researchers to measure gene expression at the resolution of single cells. Because of the low amount of RNA present in a single cell, many genes fail to be detected even though they are expressed; these genes are usually referred to as dropouts. Here, we present a general and flexible zero-inflated negative binomial model (ZINB-WaVE), which leads to low-dimensional representations of the data that account for zero inflation (dropouts), over-dispersion, and the count nature of the data. We demonstrate, with simulations and real data, that the model and its associated estimation procedure are able to give a more stable and accurate low-dimensional representation of the data than principal component analysis (PCA) and zero-inflated factor analysis (ZIFA), without the need for a preliminary normalization step.

Introduction

Single-cell RNA sequencing (scRNA-seq) is a powerful and relatively young technique enabling the characterization of the molecular states of individual cells through their transcriptional profiles [1]. It represents a major advance with respect to standard "bulk" RNA sequencing, which is only capable of measuring gene expression levels averaged over millions of cells. Such averaged gene expression profiles may be enough to characterize the global state of a tissue, but completely mask signal coming from individual cells, ignoring tissue heterogeneity. Assessing cell-to-cell variability in expression is crucial for disentangling complex heterogeneous tissues [2–4] and for understanding dynamic biological processes, such as embryo development [5] and cancer [6]. Despite the early successes of scRNA-seq, in order to fully exploit the potential of this new technology, it is essential to develop statistical and computational methods specifically designed for the unique challenges of this type of data [7].

Because of the tiny amount of RNA present in a single cell, the input material needs to go through many rounds of amplification before being sequenced. This results in strong *amplification bias*, as well as *dropouts*, i.e., genes that fail to be detected even though they are expressed in the sample [8]. The inclusion in the library preparation of unique molecular identifiers (UMIs) reduces amplification bias [9], but does not remove dropout events, nor the need for data normalization [10, 11]. In addition to the host of unwanted

*Corresponding authors: `sandrine@stat.berkeley.edu` , `jean-philippe.vert@ens.fr` technical effects that affect bulk RNA-seq, scRNA-seq data exhibit much higher variability between technical replicates, even for genes with medium or high levels of expression [12].

The large majority of published scRNA-seq analyses include a dimensionality reduction step. This achieves a two-fold objective: (i) the data become more tractable, both from a statistical (cf. curse of dimensionality) and computational point of view; (ii) noise can be reduced while preserving the often intrinsically low-dimensional signal of interest. Dimensionality reduction is used in the literature as a preliminary step prior to clustering [3, 13, 14], the inference of developmental trajectories [15–18], spatio-temporal ordering of the cells [5, 19], and, of course, as a visualization tool [20, 21]. Hence, the choice of dimensionality reduction technique is a critical step in the data analysis process.

A natural choice for dimensionality reduction is principal component analysis (PCA), which projects the observations onto the space defined by linear combinations of the original variables with successively maximal variance. However, several authors have reported on shortcomings of PCA for scRNA-seq data. In particular, for real datasets, the first or second principal components often depend more on the proportion of detected genes per cell (i.e., genes with at least one read) than on an actual biological signal [22, 23]. In addition to PCA, dimensionality reduction techniques used in the analysis of scRNA-seq data include independent components analysis (ICA) [15], Laplacian eigenmaps [18, 24], and t-distributed stochastic neighbor embedding (t-SNE) [2, 4, 25]. Note that none of these techniques can account for dropouts, nor for the count nature of the data. Typically, researchers transform the data using the logarithm of the (possibly normalized) read counts, adding an offset to avoid taking the log of zero.

Recently, Pierson & Yau [26] proposed a zero-inflated factor analysis (ZIFA) model to account for the presence of dropouts in the dimensionality reduction step. Although the method accounts for the zero inflation typically observed in scRNA-seq data, the proposed model does not take into account the count nature of the data. In addition, the model makes a strong assumption regarding the dependence of the probability of detection on the mean expression level, modeling it as an exponential decay. The fit on real datasets is not always good and, overall, the model lacks flexibility, with its inability to include covariates and/or normalization factors.

Here, we propose a general and flexible method that uses a zero-inflated negative binomial (ZINB) model to extract low-dimensional signal from the data, accounting for zero inflation (dropouts), over-dispersion, and the count nature of the data. We call this approach Zero-Inflated Negative Binomial-based Wanted Variation Extraction (ZINB-WaVE). The proposed model includes a sample-level intercept, which serves as a global-scaling normalization factor, and gives the user the ability to include both gene-level and sample-level covariates. The inclusion of observed and unobserved sample-level covariates enables normalization for complex, non-linear effects (often referred to as batch effects), while gene-level covariates may be used to adjust for sequence composition effects, such as gene length and GC-content effects. ZINB-WaVE is an extension of the RUV model [27, 28], which accounts for zero inflation and over-dispersion and for which unobserved sample-level factors may either capture variation of interest or unwanted variation. We demonstrate, with simulations and real data, that the model and its associated estimation procedure are able to give a more stable and accurate low-dimensional representation of the data than PCA and ZIFA, without the need for a preliminary normalization step. The model is implemented in the open-source R package `zinbwave`, publicly available at `https://github.com/drisso/zinbwave` and to be released through the Bioconductor Project (`http://www.bioconductor.org`).

Results

ZINB-WaVE: A general and flexible model for scRNA-seq

ZINB-WaVE is a general and flexible model for the analysis of high-dimensional zero-inflated count data, such as those recorded in single-cell RNA-seq assays. Given n samples (typically, n single cells) and J features (typically, J genes) that can be counted for each sample, we denote with Y_{ij} the count of feature j ($j = 1, \dots, J$) for sample i ($i = 1, \dots, n$). To account for various technical and biological effects, typical of single-cell sequencing technologies, we model Y_{ij} as a random variable following a zero-inflated negative binomial distribution (see Methods for details).

Both the mean expression level (μ) and the probability of dropouts (π) are modeled in terms of *observed* sample-level and gene-level covariates (X and V, respectively, Fig. 1). In addition, we include a set of *unobserved* sample-level covariates (W) that need to be inferred from the data. The matrix X can include covariates that induce variation of interest, such as cell types, or covariates that induce unwanted variation, such as batch or quality control (QC) measures. It can also include a constant column of ones for an intercept that accounts for gene-specific differences in mean expression level or dropout rate (cf. scaling in PCA). The matrix V can also accommodate an intercept to account for cell-specific global effects, such as size factors representing differences in library sizes (i.e., total number of reads per sample). In addition, V can include gene-level covariates, such as gene length or GC-content.

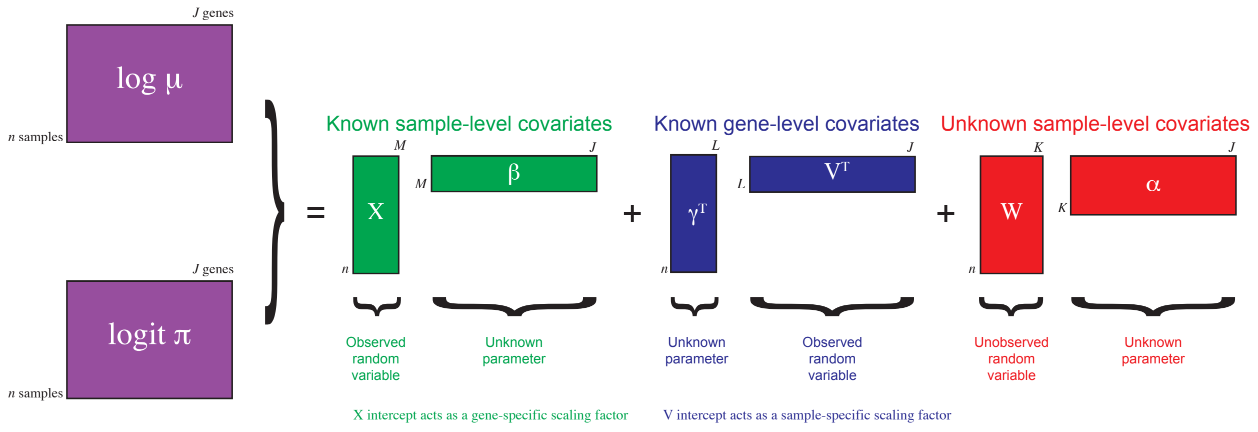

J genes
log μ
n samples
J genes
logit π
n samples
=
Known sample-level covariates
M
J
X
M
β
n
+
Known gene-level covariates
L
J
γT
L
VT
n
+
Unknown sample-level covariates
K
J
W
K
α
n
Observed random variable
Unknown parameter
Unknown parameter
Observed random variable
Unobserved random variable
Unknown parameter
X intercept acts as a gene-specific scaling factor
V intercept acts as a sample-specific scaling factor

Figure 1: *Schematic view of the ZINB-WaVE model.* Given n cells and J genes, let Y_{ij} denote the count of gene j ($j = 1, \dots, J$) for cell i ($i = 1, \dots, n$) and Z_{ij} an unobserved indicator variable, equal to one if gene j is a dropout in cell i and zero otherwise. Then, $\mu_{ij} = E[Y_{ij}|Z_{ij} = 0, X, V, W]$ and $\pi_{ij} = Pr(Z_{ij} = 1|X, V, W)$. We model $\ln(\mu)$ and $\text{logit}(\pi)$ with the regression specified in the figure. Note that the model allows for different covariates to be specified in the two regressions; we have omitted the μ and π indices for clarity (see Methods for details).

The unobserved matrix W contains unknown sample-level covariates, which could correspond to unwanted variation as in RUV [27, 28] or could be of interest as in cluster analysis (e.g., cell type). The model extends the RUV framework to the ZINB distribution (thus far, RUV had only been implemented for linear [27] and log-linear regression [28]). It differs in interpretation from RUV in the $W\alpha$ term, which is not necessarily considered unwanted; this term generally refers to unknown low-dimensional variation, that could be due to unwanted technical effects (as in RUV), such as batch effects, or to biological effects of interest, such as cell cycle or cell differentiation.

It is important to note that although W is the same, the matrices X and V could differ in the modeling of μ and π, if we assume that some known factors do not affect both. When $X = \mathbf{1}_n$ and $V = \mathbf{1}_J$, the model is a factor model akin to principal component analysis, where W is a factor matrix and α_μ and α_π are loading matrices. However, the model is more general, allowing the inclusion of additional sample and gene-level covariates that might help to infer the unknown factors.

Real datasets: ZINB-WaVE leads to biologically meaningful clusters

We applied the ZINB-WaVE procedure to four publicly-available real datasets (see Methods). First, we compared the goodness-of-fit of our ZINB-WaVE model to that of a negative binomial (NB) model (as implemented in edgeR). As previously noted in the literature, NB models, which are quite successful for bulk RNA-seq, are not appropriate for single-cell RNA-seq as they cannot accommodate zero inflation. In particular, NB models lead to biased estimation of the overall mean count and zero probability (Supplementary Fig. S1 and S2) and appear to handle the excess of zeros by over-estimating the dispersion parameter (Supplementary Fig. S3). By contrast, ZINB models, that account specifically for the extra zeros, lead to better fit for the overall mean count and zero probability and more stable estimators of the dispersion parameter.

As previously shown [22, 23], the first few principal components of scRNA-seq data, even after normalization, can be influenced by technical rather than biological features, such as the proportion of genes with at least one read (*detection rate*) or the total number of reads (*sequencing depth*) per sample. Figure 2 illustrates this using the Glioblastoma dataset: although the first two principal components somewhat segregated the data by patient (Fig. 2a), the clustering was far from perfect and individuals MGH28 and MGH31 were especially difficult to tell apart. This is at least partly due to unwanted technical effects, such as sequencing depth and amount of starting material. To quantify such technical effects, we computed a set of QC measures, such as dropout rate and total number of reads (see Methods). The first principal component turned out to be correlated with such measures, in particular dropout rate, total number of reads, and total number of detected genes (Fig. 2b).

ZIFA suffered from the same problem: the clustering of the samples in two dimensions was not qualitatively different from PCA (Fig. 2c) and the second component was highly correlated with dropout and detection rates (Fig. 2d). Conversely, ZINB-WaVE led to tighter clusters, grouping the cells by patient (Fig. 2e). Perhaps more importantly, the two components inferred by ZINB-WaVE showed lower correlation with the QC features (Fig. 2f), highlighting that the clusters shown in Figure 2e are not driven by technical effects.

These observations are not limited to the Glioblastoma dataset, as the same trend was observed for the V1 dataset (Supplementary Fig. S4), S1/CA1 dataset (Supplementary Fig. S5), and mESC dataset (Supplementary Fig. S6). See also Hicks *et al.* [22] for additional datasets in which principal components are strongly correlated with detection rate.

As a measure of the goodness of the clustering results, we used the average silhouette width (see Methods), computed using the labels available from the original study: these were either known *a priori* (e.g., the patient ID in the Glioblastoma dataset) or inferred from the data (and consequently validated) by the authors of the original publication (e.g., the cell types in the S1/CA1 dataset). For all four datasets, ZINB-WaVE led to generally tighter clusters, as shown by an increased average silhouette width per cluster (Supplementary Fig. S7).

Real datasets: Impact of normalization

As for any high-throughput genomic technology, an important aspect of scRNA-seq data analysis is normalization. Indeed, there are a variety of steps in scRNA-seq experiments that can introduce bias in the data and that need to be corrected for [11]. Some of these effects, e.g., sequencing depth, can be captured by a global-scaling factor (usually referred to as *size factor*). Other more complex, non-linear effects, such as those collectively known as *batch effects*, require more sophisticated normalization [28]. Accordingly, a typical scRNA-seq pipeline will include a normalization step, i.e., a linear or non-linear transformation of read counts, to make the distributions of expression measures comparable between samples. The normalization step is usually carried out prior to any inferential procedure (e.g., clustering, differential expression analysis, or pseudotime ordering). In this work, we considered three popular normalization methods: total-count (TC), trimmed-mean of M-values (TMM), and full-quantile (FQ) normalization (see Methods); we also compared the results to unnormalized data (RAW). TC, along with the related methods Transcripts Per Million (TPM) and Fragments Per Kilobase Million (FPKM), is by far the most widely used normalization in the scRNA-seq literature; hence, TC-normalized data were used for the results shown in Figure 2 and in Supplementary Figures S4, S5, and S6.

Normalization highly affected the projection of the data in lower dimensions (Supplementary Fig. S8) and, consequently, the clustering results varied greatly between normalization methods in all four datasets (Fig. 3). Strikingly, the choice of normalization method was more critical than the choice between PCA and ZIFA. For instance, in the V1 dataset (Fig. 3a), FQ normalization followed by either PCA or ZIFA outperformed all the other approaches. Critically, the ranking of normalization methods was not consistent across datasets (Fig. 3), highlighting that the identification of the best normalization method for a given dataset is a difficult problem for scRNA-seq [11, 29].

One important feature of the ZINB-WaVE model is that the sample-level intercept γ (corresponding to a column of ones in the gene-level covariate matrix V, see Methods) acts as a global-scaling normalization factor, making a preliminary global-scaling step unnecessary. As a consequence, ZINB-WaVE can be directly applied to unnormalized read counts, hence preserving the distributional properties of count data. ZINB-WaVE applied to unnormalized counts led to results that were comparable, in terms of average silhouette width, to PCA and ZIFA applied using the best performing normalization (Fig. 3). In particular, ZINB-

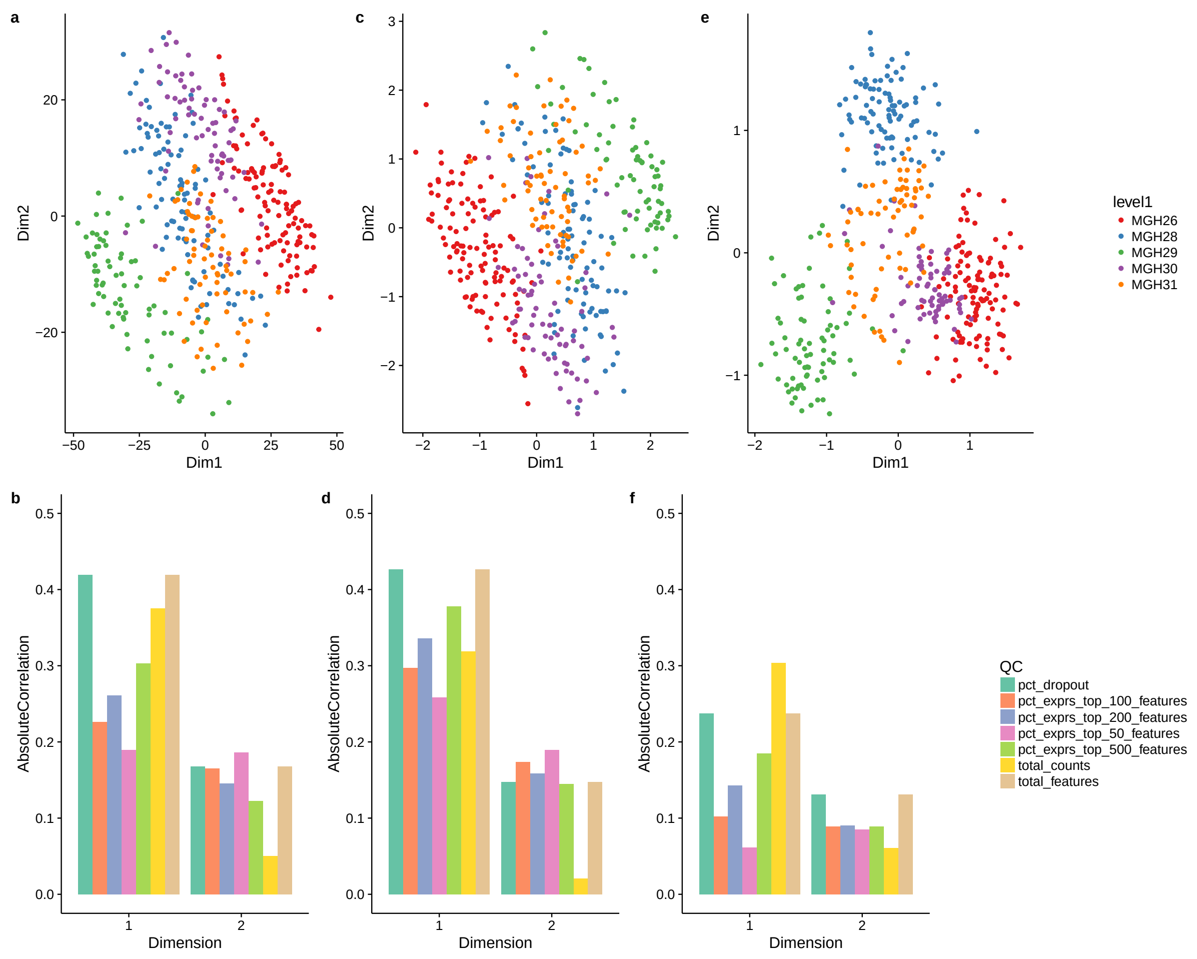


Figure 2: *Low-dimensional representation of the Glioblastoma dataset.* Upper panels provide two-dimensional representations of the data, after selecting the 1,000 most variable genes. Lower panels provide barplots of the absolute correlation between the first two components and a set of QC measures (see Methods). **(a, b)** PCA (on TC-normalized counts); **(c, d)** ZIFA (on TC-normalized counts); **(e, f)** ZINB-WaVE (no normalization needed). ZINB-WaVE leads to a low-dimensional representation that is less influenced by technical variation and to tighter, biologically meaningful clusters.

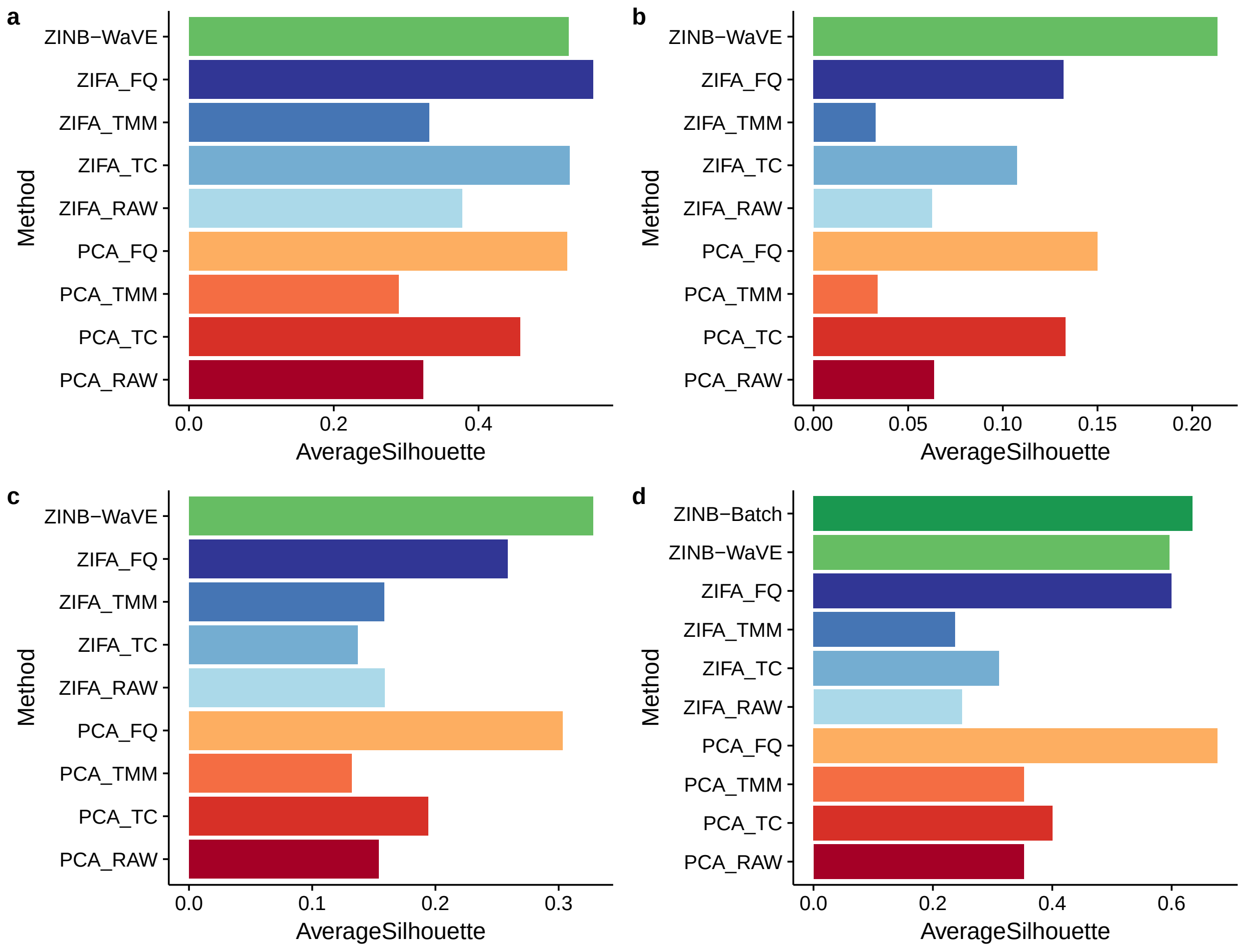


Figure 3: *Average silhouette width, real datasets.* **(a)** V1 dataset; **(b)** S1/CA1 dataset; **(c)** Glioblastoma dataset; **(d)** mESC dataset. Silhouette widths were computed in the low-dimensional space, using the groupings provided by the authors of the original publications: unsupervised clustering procedure **(a, b)**, observed characteristics of the samples, such as patient **(c)** and culture condition **(d)**. PCA and ZIFA were applied after normalization: unnormalized (RAW), total-count (TC), trimmed-mean of M values (TMM), and full-quantile (FQ). For the mESC dataset **(d)**, we fitted the ZINB-WaVE model with batch as a sample-level covariate (ZINB-Batch) in addition to the default model with only a sample-level intercept (see Fig. 4).

WaVE outperformed PCA and ZIFA on the S1/CA1 (Fig. 3b) and Glioblastoma (Fig. 3c) datasets, while it was slightly worse than PCA (after FQ normalization) on the mESC dataset (Fig. 3d) and than PCA and ZIFA (after FQ normalization) on the V1 dataset (Fig. 3a). Interestingly, the overall average silhouette width was lower for the S1/CA1 and Glioblastoma datasets than it was for the V1 and mESC datasets, suggesting that ZINB-WaVE leads to better clustering in more complex situations (e.g., lower sequencing depth).

Although the overall average silhouette width is a useful metric to rank the different methods in terms of how well they represent known biological structure, looking only at the average across many different cell types may be misleading. For the V1 dataset, for instance, ZINB-WaVE led to a slightly lower overall average silhouette width than ZIFA (Fig. 3a). However, ZINB-WaVE yielded higher cluster-level average silhouette widths for the L6a, L5b, and L5 samples, while ZIFA produced higher average silhouette widths for the L4 and L5a samples (Supplementary Fig. S7a). In fact, certain cell types may be easier to cluster than others, leading to silhouette widths that differ greatly between different clusters, as is the case for the S1/CA1 dataset: oligodendrocytes are much easier to cluster than the other cell types and all methods were able to identify them (Supplementary Fig. S7b); on the other hand, only ZINB-WaVE was able to achieve a positive silhouette width for the pyramidal SS and CA1 neurons and it performed much better than PCA and ZIFA in the interneuron cluster; finally, certain groups, such as the endothelial-mural and astrocytes, were simply too hard to distinguish in three dimensions (Supplementary Fig. S7b).

Real datasets: Accounting for batch effects

Although the sample-level intercept (corresponding to a column of ones in the gene-level covariate matrix V) can act as a global-scaling normalization factor, this may not be sufficient to accurately account for complex, non-linear technical effects that may influence the data (e.g., batch effects). Hence, we explored the possibility of including additional sample-level covariates in X to account for such effects (see Methods). We illustrate this using the mESC dataset, which is a subset of the data described in Kolodziejczyk *et al.* [30] and which comprises two batches of mESCs grown in three different media (see Methods). ZINB-WaVE applied with no additional covariates led to three clusters, corresponding to the three media (Fig. 4a and Supplementary Fig. S6). However, the clusters corresponding to media 2i and a2i are further segregated by batch (Fig. 4a). Including indicator variables for batch in ZINB-WaVE (as columns of X) removed such batch effects, consolidating the clustering by medium (Fig. 4b). This led to slightly larger average silhouette widths, both overall (Fig. 3c) and at the cluster level (Fig. 4c). Conversely, the average silhouette width for batch, a measure of how well the cells cluster by batch, was much larger in the model that did not include the batch covariate (Fig. 4d), indicating that explicitly accounting for batch in the ZINB-WaVE model effectively removed the dependence of the inferred low-dimensional space on batch effects.

Simulated datasets: ZINB-WaVE estimators are asymptotically unbiased and robust

We next turned to simulations to explore in greater detail the performance of ZINB-WaVE. First, we evaluated the ZINB-WaVE estimation procedure on simulated data from a zero-inflated negative binomial distribution, to assess both accuracy under a correctly specified model and robustness to model misspecification. The approach involves computing maximum likelihood estimators (MLE) for the parameters of the model of Figure 1, namely, α, β, γ, and W, and hence μ and π. MLE are well-behaved, in that they are asymptotically unbiased and efficient. However, because the likelihood function of our ZINB-WaVE model is not convex, our numerical optimization procedure may converge to a local maximum, rather than to the true MLE (see Methods). We observed that our estimators are asymptotically unbiased and have decreasing variance as the number of cells n increases (Supplementary Fig. S9). This suggests that our estimates are not far from the true MLE.

To assess the robustness of the estimators, we looked at bias and mean squared error (MSE) for μ and π, for misspecified number of unobserved covariates K (i.e., number of columns of W), gene-level covariate matrix V, and dispersion parameter ϕ (Fig. 5, Supplementary Fig. S10 and S11). In Figure 5, the data were simulated with $K = 2$ unknown factors, $X = \mathbf{1}_n$, $V = \mathbf{1}_J$, and genewise dispersion. The model parameters were then estimated with $K = 1, 2, 3, 4$, both for a model that included a cell-level intercept ($V = \mathbf{1}_J$) and

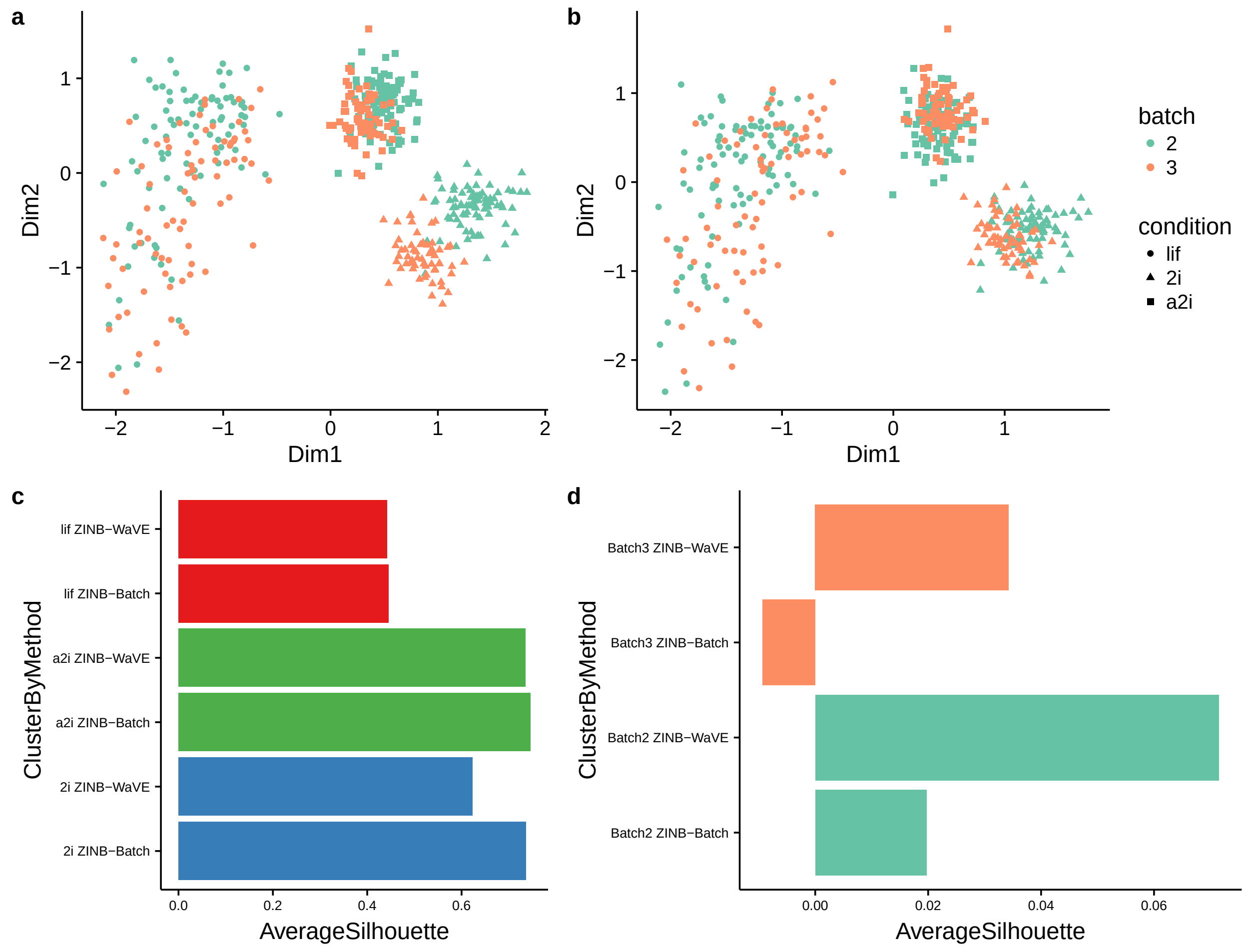


Figure 4: *ZINB-WaVE applied to the mESC dataset.* Upper panels provide two-dimensional representations of the data, with cells color-coded by batch and shape reflecting culture conditions: **(a)** Default ZINB-WaVE with only sample-level intercept; **(b)** ZINB-WaVE with batch as known sample-level covariate. **(c)** Average silhouette widths by biological condition for ZINB-WaVE with and without batch covariate; **(d)** Average silhouette widths by batch for ZINB-WaVE with and without batch covariate. Although the data cluster primarily based on culture condition, batch effects are evident in **(a)**. Accounting for batch effects in the ZINB-WaVE model **(b)** leads to slightly better clustering by biological condition **(c)** and removes the clustering by batch **(d)**. Note the difference in scale between the barplots of **(c)** and **(d)**.

one that did not include such intercept ($V = \mathbf{0}_J$). When the intercept was correctly included in the model and $K \geq 2$, we observed no or very small bias (Fig. 5a, b) and small MSE (Fig. 5c, d). Interestingly, the misspecification of K led to only a slight increase in bias and MSE. When no intercept was included in the model, the sensitivity to K became more important. Although the data were simulated with $K = 2$, specifying $K \geq 3$ led to small bias and low MSE (Fig. 5). This is likely because one column of W acted as a *de facto* intercept, overcoming the absence of V. In addition, we observed robustness of the results to the choice of dispersion parameter (genewise or common), even though the data were simulated with genewise dispersion (Fig. 5). The estimators for $\ln(\mu)$ and π were unbiased over the whole range of mean expression and zero inflation probability (Supplementary Fig. S12).

Simulated datasets: ZINB-WaVE is more accurate than state-of-the-art methods

We next compared ZINB-WaVE to PCA and ZIFA, in terms of their ability to recover the true underlying low-dimensional signal and clustering structure. For datasets simulated from a ZINB model, our estimation procedure with correctly specified K led to almost perfect correlation between distances in the true and estimated low-dimensional space (Fig. 6a, b). The correlation remained high even for misspecified K, in most cases higher than for PCA and ZIFA. ZINB-WaVE performed consistently well across a range of simulation scenarios, including different numbers of cells n, different zero fractions, and varying cluster tightness (Supplementary Fig. S14). We observed a consistent ranking, although noisier, when the methods were compared in terms of silhouette widths (Fig. 6c, d).

As with the real datasets, the choice of normalization influenced the simulation results. Critically, there was not an overall best performing normalization method; rather, the performance of the normalization methods depended on the dimensionality reduction method and on intrinsic characteristics of the data, such as the fraction of zero counts and the number and tightness of the clusters (Fig. 6 and Supplementary Fig. S14). For instance, TC normalization worked better for PCA on data simulated from the V1 dataset (Fig. 6a, c), while FQ and TMM normalization worked better for PCA and ZIFA, respectively, on data simulated from the S1/CA1 dataset (Fig. 6b, d).

It is important to note that the previous results were obtained for data simulated from the ZINB-WaVE model underlying our estimation procedure. It is hence not surprising that ZINB-WaVE outperformed its competitors. To provide a fairer comparison, we also assessed the methods on data simulated from the model proposed by Lun & Marioni [31]. Although this model is also based on a zero-inflated negative binomial distribution, the distribution is parameterized differently and, in particular, does not involve regression on gene-level and sample-level covariates (see Methods).

When the data were simulated to have a moderate fraction of zeros (namely, 40%), all methods performed well in terms of average silhouette width (Fig. 6e-g). However, the performance of PCA decreased dramatically with 60% of zeros, independently of the number of cells n. Although ZIFA worked well with 60% or fewer zeros, its performance too decreased at 80% of zeros, especially when only 100 cells were simulated (Fig. 6e). Conversely, the performance of ZINB-WaVE remained good even when simulating data with 80% of zeros, independently of the sample size.

Discussion

Recent advances in single-cell technologies, such as droplet-based methods [2], make it easy and inexpensive to collect hundreds of thousands of scRNA-seq profiles, allowing researchers to study very complex biological systems at high resolution. The resulting libraries are often sequenced at extremely low depth (tens of thousands of reads per cell, only), making the corresponding read count data truly sparse. Hence, there is a growing need for developing reliable statistical methods that are scalable and that can account for zero inflation.

ZINB-WaVE is a general and flexible approach to extract low-dimensional signal from noisy, zero-inflated data, such as those from scRNA-seq experiments. We have shown with simulations and real data analyses that ZINB-WaVE leads to robust and unbiased estimators of the underlying biological signals. The better performance of ZINB-WaVE with respect to PCA comes at a computational cost, since we need to numerically optimize a non-convex likelihood function. However, we empirically found that the computing time was

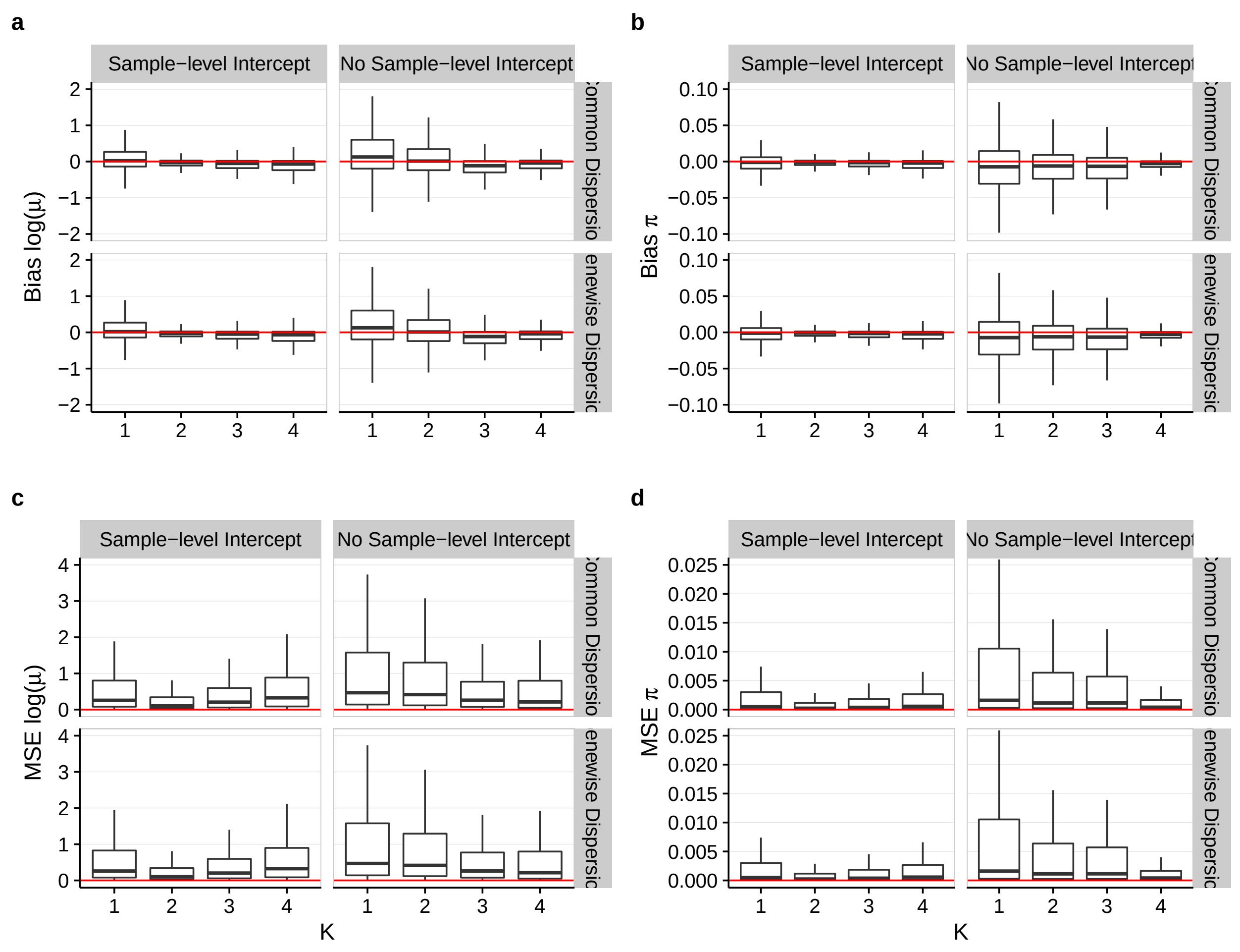


Figure 5: *Bias and MSE for ZINB-WaVE estimation procedure, ZINB-WaVE simulation model.* Panels show boxplots of **(a)** bias $\ln(\mu)$, **(b)** bias π, **(c)** MSE $\ln(\mu)$, and **(d)** MSE π for ZINB-WaVE estimation procedure, as a function of the number of unknown covariates K, for different fitted dispersion models (common and genewise) and with/without sample-level intercept (i.e., column of ones in gene-level covariate matrix V); $X = \mathbf{1}_n$ in all fits. For each gene and cell, bias and MSE were averaged over $B = 10$ datasets simulated from our ZINB-WaVE model, based on the S1/CA1 dataset and with $n = 1,000$ cells, $J = 1,000$ genes, scaling of one for the ratio of within to between-cluster sums of squares ($b^2 = 1$), $K = 2$ unknown factors, zero fraction of about 80%, $X = \mathbf{1}_n$, cell-level intercept ($V = \mathbf{1}_J$), and genewise dispersion. The lower and upper hinges of the boxplots correspond to the first and third quartiles (the 25th and 75th percentiles), respectively. The upper whisker extends from the hinge to the largest value no further than $1.5 \times IQR$ from the hinge (where IQR is the inter-quartile range or difference between the third and first quartiles). The lower whisker extends from the hinge to the smallest value no further than $1.5 \times IQR$ from the hinge. Data beyond the end of the whiskers are called outliers and are not plotted here. Fig. S10 provides the same boxplots with individually plotted outliers.

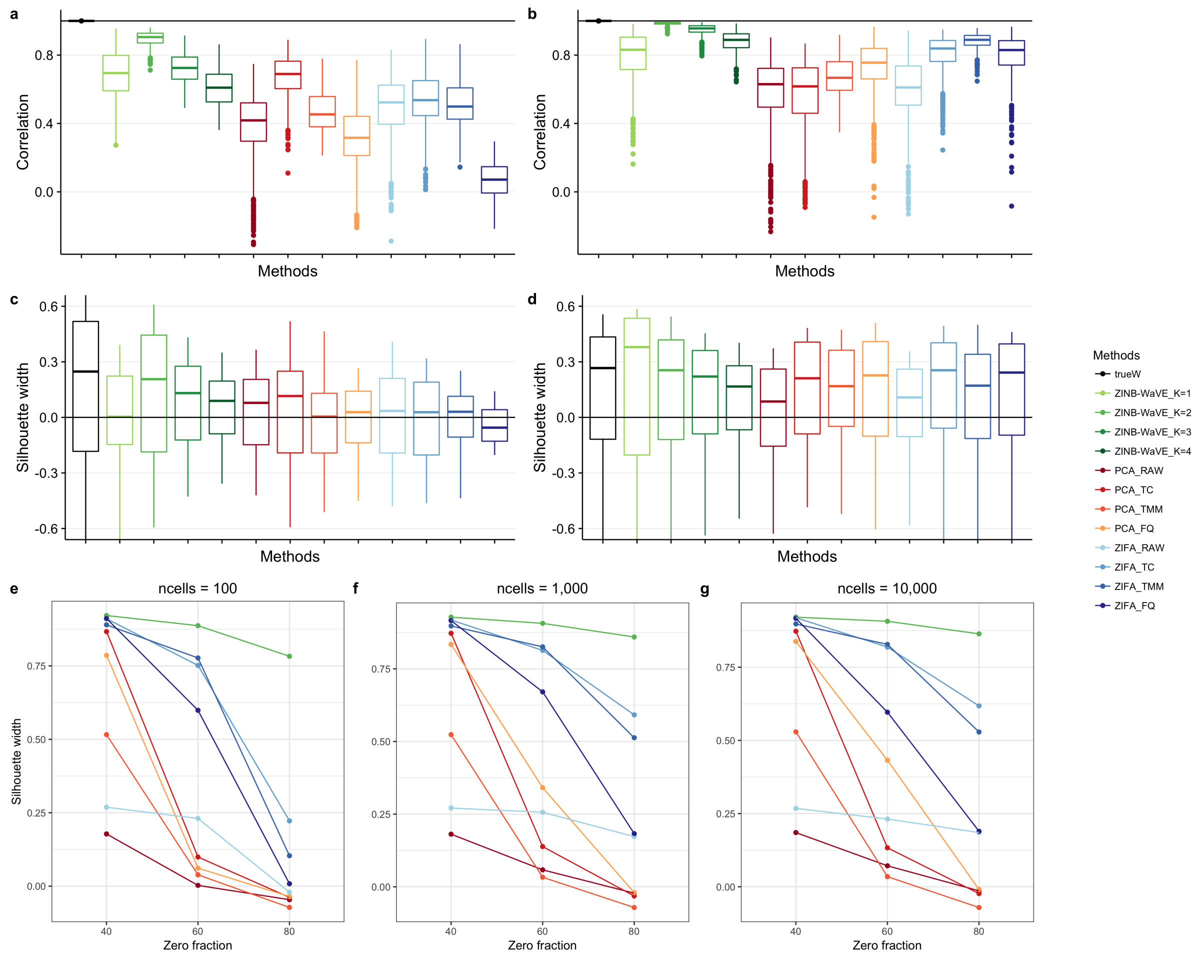


Figure 6: *Between-sample distances and silhouette widths for ZINB-WaVE, PCA, and ZIFA, simulation models.* **(a)** Boxplots of correlations between between-sample distances based on true and estimated low-dimensional representations of the data for datasets simulated from the V1 dataset. **(b)** Same as **(a)** for simulations based on the S1/CA1 dataset. **(c)** Boxplots of silhouette widths for true clusters for datasets simulated from the V1 dataset. **(d)** Same as **(c)** for simulations based on the S1/CA1 dataset. For **(a-d)**, all datasets were simulated from our ZINB-WaVE model with $n = 1,000$ cells, $J = 1,000$ genes, "harder" clustering ($b^2 = 5$), $K = 2$ unknown factors, zero fraction of about 80%, $X = \mathbf{1}_n$, cell-level intercept ($V = \mathbf{1}_J$), and genewise dispersion. Each boxplot is based on n values corresponding to each of the n samples and defined as averages of correlations **(a, b)** or silhouette widths **(c, d)** over $B = 10$ simulations. See Supplementary Figure S13 for the same scenario but with $n = 10,000$ cells and Supplementary Figure S14 for additional scenarios. **(e-g)** Average silhouette widths (over n samples and $B = 10$ simulations) for true clusters vs. zero fraction, for $n \in \{100, 1,000, 10,000\}$ cells, for datasets simulated from the Lun & Marioni [31] model, with $C = 3$ clusters and equal number of cells per cluster. While ZINB-WaVE was relatively robust to the sample size n and zero fraction, the performance of PCA and ZIFA decreased with larger zero fraction. Between-sample distance matrices and silhouette widths were based on W for ZINB-WaVE, the first two principal components for PCA, and the first two latent variables for ZIFA. ZINB-WaVE was applied with $X = \mathbf{1}_n$, $V = \mathbf{1}_J$, genewise dispersion, and $K \in \{1, 2, 3, 4\}$ (only $K = 2$ is shown in **(e-g)**). For PCA and ZIFA, different normalization methods were used. Colors correspond to the different methods.

approximately linear in the number of cells (Supplementary Fig. S15). The algorithm benefits from parallelization on multicore machines, and takes a few minutes on a modern laptop to converge with thousands of cells.

One major difference between ZINB-WaVE and previously proposed factor analysis models (such as PCA and ZIFA) is the ability to include sample-level and gene-level covariates. In particular, by including a column of ones in the gene-level covariate matrix, the corresponding cell-level intercept acts as a global-scaling normalization factor, allowing the modeling of raw count data, with no need for prior normalization.

Several authors have recognized that high-dimensional genomic data are affected by confounding factors, such as batch effects, and have proposed methods to account for such effects in either a supervised [32] or unsupervised way [27, 33]. Recently, Lin *et al.* [34] proposed a model that can extend PCA to adjust for confounding factors. This model, however, does not seem to be ideal for zero-inflated count data. In the scRNA-seq literature, MAST [23] uses the inferred *cellular detection rate* to adjust for the main source of confounding, in a differential expression setting, but is not designed to infer low-dimensional signal. The removal of batch effects is an important example of how including additional covariates in the ZINB-WaVE model may lead to better low-dimensional representations of the data. However, ZINB-WaVE is not limited to including batch effects, as other sample-level (e.g., sample QC metrics) and/or gene-level (e.g., GC-content) covariates may be included in the model.

In this article, we have focused on an unsupervised setting, where the goal is to extract a low-dimensional signal from noisy zero-inflated data. However, our proposed ZINB model is more general and can be used, in principle, for supervised differential expression analysis, where the parameters of interest are regression coefficients β corresponding to known sample-level covariates in the matrix X (e.g., cell type, treatment/control status). Differentially expressed genes may be identified via likelihood ratio tests or Wald tests, with standard errors of estimators of β obtained from the Hessian matrix of the likelihood function. We envision a future version of the zinbwave package with this added capability.

Although the low-dimensional signal inferred by ZINB-WaVE can be used to visually inspect hidden structure in the data, visualization is not the main point of our proposed method. Low-dimensional factors are intended as the closest possible approximation to the true signal, which is assumed to be intrinsically low-dimensional. Such low-dimensional representation can be used in downstream analyses, such as clustering or pseudotime ordering of the cells [15].

Visualization of high-dimensional datasets is an equally important area of research and many proposed algorithms exist, among which t-SNE [25] has become the most popular for scRNA-seq data. Recently, Wang *et al.* [35] have proposed a novel visualization algorithm that can account for zero inflation and showed improvement over t-SNE. As t-SNE takes as input a matrix of pairwise distances between cells, a promising line of future work may be to derive such a matrix from ZINB-WaVE and use t-SNE for visualization purpose.

Methods

ZINB-WaVE model

For any $\mu \geq 0$ and $\theta > 0$, let $f_{NB}(\cdot\,; \mu, \theta)$ denote the probability mass function (PMF) of the negative binomial (NB) distribution with mean μ and inverse dispersion parameter θ, namely:

$$f_{NB}(y; \mu, \theta) = \frac{\Gamma(y+\theta)}{\Gamma(y+1)\Gamma(\theta)} \left(\frac{\theta}{\theta+\mu}\right)^{\theta} \left(\frac{\mu}{\mu+\theta}\right)^{y}, \quad \forall y \in \mathbb{N}. \tag{1}$$

Note that another parametrization of the NB PMF is in terms of the dispersion parameter $\phi = \theta^{-1}$ (although θ is also sometimes called dispersion parameter in the literature). In both cases, the mean of the NB distribution is μ and its variance is:

$$\sigma^2 = \mu + \frac{\mu^2}{\theta} = \mu + \phi\mu^2 \,. \tag{2}$$

In particular, the NB distribution boils down to a Poisson distribution when $\phi = 0 \Leftrightarrow \theta = +\infty$.

For any $\pi \in [0, 1]$, let $f_{ZINB}(\cdot\,; \mu, \theta, \pi)$ be the PMF of the zero-inflated negative binomial (ZINB) distribution given by:

$$f_{ZINB}(y; \mu, \theta, \pi) = \pi\delta_0(y) + (1-\pi)f_{NB}(y; \mu, \theta), \quad \forall y \in \mathbb{N}, \tag{3}$$

where $\delta_0(\cdot)$ is the Dirac function. Here, π can be interpreted as the probability that a 0 is observed instead of the actual count, resulting in an inflation of zeros compared to the NB distribution, hence the name ZINB.

Given n samples (typically, n single cells) and J features (typically, J genes) that can be counted for each sample, let Y_{ij} denote the count of feature j (for $j = 1, \ldots, J$) for sample i ($i = 1, \ldots, n$). To account for various technical and biological effects frequent, in particular, in single-cell sequencing technologies, we model Y_{ij} as a random variable following a ZINB distribution with parameters μ_{ij}, θ_{ij}, and π_{ij}, and consider the following regression models for the parameters:

$$\ln(\mu_{ij}) = \left(X\beta_\mu + (V\gamma_\mu)^\top + W\alpha_\mu + O_\mu\right)_{ij} , \tag{4}$$

$$\text{logit}(\pi_{ij}) = \left(X\beta_\pi + (V\gamma_\pi)^\top + W\alpha_\pi + O_\pi\right)_{ij} , \tag{5}$$

$$\ln(\theta_{ij}) = \zeta_j , \tag{6}$$

where

$$\text{logit}(\pi) = \ln\left(\frac{\pi}{1-\pi}\right)$$

and elements of the regression models are as follows.

- X is a known $n \times M$ matrix corresponding to M cell-level covariates and $\beta = (\beta_\mu, \beta_\pi)$ its associated $M \times J$ matrices of regression parameters. X can typically include covariates that induce variation of interest, such as cell types, or covariates that induce unwanted variation, such as batch or quality control measures. It can also include a constant column of ones, $\mathbf{1}_n$, to account for gene-specific intercepts.
- V is a known $J \times L$ matrix corresponding to J gene-level covariates, such as gene length or GC-content, and $\gamma = (\gamma_\mu, \gamma_\pi)$ its associated $L \times n$ matrices of regression parameters. V can also include a constant column of ones, $\mathbf{1}_J$, to account for cell-specific intercepts, such as size factors representing differences in library sizes.
- W is an unobserved $n \times K$ matrix corresponding to K unknown cell-level covariates, which could be of "unwanted variation" as in RUV [27, 28] or of interest (such as cell type), and $\alpha = (\alpha_\mu, \alpha_\pi)$ its associated $K \times J$ matrices of regression parameters.
- O_μ and O_π are known $n \times J$ matrices of offsets.
- $\zeta \in \mathbb{R}^J$ is a vector of gene-specific dispersion parameters on the log scale.

This model deserves a few comments.

- By default, X and V contain a constant column of ones, to account, respectively, for gene-specific (e.g., baseline expression level) and cell-specific (e.g., library size) variation. In that case, X and V are of the form $X = [\mathbf{1}_n, X^0]$ and $V = [\mathbf{1}_J, V^0]$ and we can similarly decompose the corresponding parameters as $\beta = [\beta^1, \beta^0]$ and $\gamma = [\gamma^1, \gamma^0]$, where $\beta^1 \in \mathbb{R}^{1\times J}$ is a vector of gene-specific intercepts and $\gamma^1 \in \mathbb{R}^{1\times n}$ a vector of cell-specific intercepts. The representation $\mathbf{1}_n\beta^1 + (\mathbf{1}_J\gamma^1)^\top$ is then not unique, but could be made unique by adding a constant and constraining β^1 and γ^1 to each have elements summing to zero.
- Although W is the same, the matrices X and V could differ in the modeling of μ and π, if we assume that some known factors do not affect both μ and π. To keep notation simple and consistent, we use the same matrices, but will implicitly assume that some parameters may be constrained to be 0 if needed.
- By allowing the models to differ for μ and π, we can model and test for differential expression in terms of either the NB mean or the ZI probability.
- We limit ourselves to a gene-dependent dispersion parameter. More complicated models for θ_{ij} could be investigated, such as a model similar to μ_{ij} or a functional of the form $\theta_{ij} = f(\mu_{ij})$, but we restrict ourselves to a simpler model that has been shown to be largely sufficient in bulk RNA-seq analysis.

ZINB-WaVE estimation procedure

The input to the model are the matrices X, V, O_μ, and O_π and the integer K; the parameters to be inferred are $\beta = (\beta_\mu, \beta_\pi)$, $\gamma = (\gamma_\mu, \gamma_\pi)$, W, $\alpha = (\alpha_\mu, \alpha_\pi)$, and ζ. Given an $n \times J$ matrix of counts Y, the log-likelihood function is

$$\ell(\beta, \gamma, W, \alpha, \zeta) = \sum_{i=1}^{n} \sum_{j=1}^{J} \ln f_{ZINB}(Y_{ij}; \mu_{ij}, \theta_{ij}, \pi_{ij}) ,$$

where μ_{ij}, θ_{ij}, and π_{ij} depend on $(\beta, \gamma, W, \alpha, \zeta)$ through Equations (4)–(6).

To infer the parameters, we follow a penalized maximum likelihood approach, by trying to solve

$$\max_{\beta,\gamma,W,\alpha,\zeta} \left\{ \ell(\beta, \gamma, W, \alpha, \zeta) - \text{Pen}(\beta, \gamma, W, \alpha, \zeta) \right\} ,$$

where $\text{Pen}(\cdot)$ is a regularization term to reduce overfitting and improve the numerical stability of the optimization problem in the setting of many parameters. For nonnegative regularization parameters $(\epsilon_\beta, \epsilon_\gamma, \epsilon_W, \epsilon_\alpha, \epsilon_\zeta)$, we set

$$\text{Pen}(\beta, \gamma, W, \alpha, \zeta) = \frac{\epsilon_\beta}{2} \|\beta^0\|^2 + \frac{\epsilon_\gamma}{2} \|\gamma^0\|^2 + \frac{\epsilon_W}{2} \|W\|^2 + \frac{\epsilon_\alpha}{2} \|\alpha\|^2 + \frac{\epsilon_\zeta}{2} \text{Var}(\zeta) ,$$

where β^0 and γ^0 denote the matrices β and γ without the rows corresponding to the intercepts if an unpenalized intercept is included in the model, $\| \cdot \|$ is the Frobenius matrix norm ($\|A\| = \sqrt{\text{tr}(A^*A)}$, where A^* denotes the conjugate transpose of A), and $\text{Var}(\zeta) = 1/(J-1) \sum_{i=1}^{J} \left(\zeta_i - (\sum_{j=1}^{J} \zeta_j)/J \right)^2$ is the variance of the elements of ζ (using the unbiased sample variance statistic). The penalty tends to shrink the estimated parameters to 0, except for the cell and gene-specific intercepts which are not penalized and the dispersion parameters which are not shrunk towards 0 but instead towards a constant value across genes. Note also that the likelihood only depends on W and α through their product $R = W\alpha$ and that the penalty ensures that at the optimum W and α have the structure described in the following result which generalizes standard results such as [36] (Lemma 1) and [37] (Lemma 6).

Lemma 1. *For any matrix R and positive scalars s and t, the following holds:*

$$\min_{S,T \,:\, R=ST} \frac{1}{2} \left(s\|S\|^2 + t\|T\|^2 \right) = \sqrt{st} \|R\|_* ,$$

where $\|A\|_* = tr\left(\sqrt{A^*A}\right)$. If $R = R_L R_\Sigma R_R$ is a singular value decomposition (SVD) of R, then a solution to this optimization problem is:

$$S = \left(\frac{t}{s} \right)^{\frac{1}{4}} R_L R_\Sigma^{\frac{1}{2}} , \quad T = \left(\frac{s}{t} \right)^{\frac{1}{4}} R_\Sigma^{\frac{1}{2}} R_R .$$

Proof. Let $\tilde{S} = \sqrt{s}S$, $\tilde{T} = \sqrt{t}T$, and $\tilde{R} = \sqrt{st}R$. Then, $\|\tilde{S}\|^2 = s\|S\|^2$, $\|\tilde{T}\|^2 = t\|T\|^2$, and $\tilde{S}\tilde{T} = \sqrt{st}ST$, so that the optimization problem is equivalent to:

$$\min_{\tilde{S},\tilde{T} \,:\, \tilde{S}\tilde{T}=\tilde{R}} \frac{1}{2} \left(\|\tilde{S}\|^2 + \|\tilde{T}\|^2 \right) ,$$

which by [37] (Lemma 6) has optimum value $\|\tilde{R}\|_* = \sqrt{st}\|R\|_*$ reached at $\tilde{S} = \tilde{R}_L \tilde{R}_\Sigma^{\frac{1}{2}}$ and $\tilde{T} = \tilde{R}_\Sigma^{\frac{1}{2}} \tilde{R}_R$, where $\tilde{R}_L \tilde{R}_\Sigma \tilde{R}_R$ is a SVD of $\tilde{R}$. Observing that $\tilde{R}_L = R_L$, $\tilde{R}_R = R_R$, and $\tilde{R}_\Sigma = \sqrt{st} R_\Sigma$, gives that a solution of the optimization problem is $S = s^{-1/2}\tilde{S} = s^{-1/2} R_L (st)^{1/4} R_\Sigma^{1/2} = (t/s)^{1/4} R_L R_\Sigma^{1/2}$. A similar argument for T concludes the proof. □

This lemma implies in particular that at any local maximum of the penalized log-likelihood, W and $\alpha^\top$ have orthogonal columns, which is useful for visualization or interpretation of latent factors.

To balance the penalties applied to the different matrices in spite of their different sizes, a natural choice is to fix $\epsilon > 0$ and set

$$\epsilon_\beta = \frac{\epsilon}{J} , \quad \epsilon_\gamma = \frac{\epsilon}{n} , \quad \epsilon_W = \frac{\epsilon}{n} , \quad \epsilon_\alpha = \frac{\epsilon}{J} , \quad \epsilon_\zeta = \epsilon .$$

In particular, from Lemma 1, we easily deduce the following characterization of the penalty on W and α, which shows that the entries in the matrices W and α have similar standard deviation after optimization:

Corollary 1. *For any $n \times J$ matrix R and positive scalars ϵ, the following holds.*

$$\min_{W,\alpha\,:\,R=W\alpha} \frac{\epsilon}{2}\left(\frac{1}{n}\|W\|^2 + \frac{1}{J}\|\alpha\|^2\right) = \frac{\epsilon}{\sqrt{nJ}}\|R\|_* \,.$$

If $R = R_L R_\Sigma R_R$ is a SVD decomposition of R, then a solution to this optimization problem is:

$$W = \left(\frac{n}{J}\right)^{\frac{1}{4}} R_L R_\Sigma^{\frac{1}{2}}\,, \quad T = \left(\frac{J}{n}\right)^{\frac{1}{4}} R_\Sigma^{\frac{1}{2}} R_R \,.$$

In particular, for any $i = 1, \dots, \min(n, J)$,

$$\frac{1}{n}\sum_{j=1}^{n} W_{j,i}^2 = \frac{1}{J}\sum_{j=1}^{J} \alpha_{i,j}^2 = \frac{[R_\Sigma]_{i,i}}{\sqrt{nJ}}.$$

The penalized likelihood is however not concave, making its maximization computationally challenging. We instead find a local maximum, starting from a smart initialization and iterating a numerical optimization scheme until local convergence, as described below.

Initialization

To initialize the set of parameters we approximate the count distribution by a log-normal distribution and explicitly separate zero and non-zero values, as follows:

1. Set $\mathcal{P} = \{(i,j) \,:\, Y_{ij} > 0\}$.
2. Set $L_{ij} = \ln(Y_{ij}) - (O_\mu)_{ij}$ for all $(i,j) \in \mathcal{P}$.
3. Set $\hat{Z}_{ij} = 1$ if $(i,j) \in \mathcal{P}$, $\hat{Z}_{ij} = 0$ otherwise.
4. Estimate β_μ and γ_μ by solving the convex ridge regression problem:

$$\min_{\beta_\mu,\gamma_\mu} \sum_{(i,j)\in\mathcal{P}} \left(L_{ij} - (X\beta_\mu)_{ij} - (V\gamma_\mu)_{ji}\right)^2 + \frac{\epsilon_\beta}{2}\|\beta_\mu^0\|^2 + \frac{\epsilon_\gamma}{2}\|\gamma_\mu^0\|^2 \,.$$

This is a standard ridge regression problem, but with a potentially huge design matrix, with up to nJ rows and $MJ + nL$ columns. To solve it efficiently, we alternate the estimation of β_μ and γ_μ. Specifically, we initialize parameter values as:

$$\hat{\beta}_\mu \leftarrow 0\,, \quad \hat{\gamma}_\mu \leftarrow 0$$

and repeat the following two steps a few times (or until convergence):

(a) Optimization in γ_μ, which can be performed independently and in parallel for each cell:

$$\hat{\gamma}_\mu \in \arg\min_{\gamma_\mu} \sum_{(i,j)\in\mathcal{P}} \left(L_{ij} - (X\hat{\beta}_\mu)_{ij} - (V\gamma_\mu)_{ji}\right)^2 + \frac{\epsilon_\gamma}{2}\|\gamma_\mu^0\|^2 \,.$$

(b) Optimization in β_μ, which can be performed independently and in parallel for each gene:

$$\hat{\beta}_\mu \in \arg\min_{\beta_\mu} \sum_{(i,j)\in\mathcal{P}} \left(L_{ij} - (V\hat{\gamma}_\mu)_{ji} - (X\beta_\mu)_{ij}\right)^2 + \frac{\epsilon_\beta}{2}\|\beta_\mu^0\|^2 \,.$$

5. Estimate W and α_μ by solving

$$\left(\hat{W}, \hat{\alpha}_\mu\right) \in \arg\min_{W,\alpha_\mu} \sum_{(i,j)\in\mathcal{P}} \left(L_{ij} - (X\hat{\beta}_\mu)_{ij} - (V\hat{\gamma}_\mu)_{ji} - (W\alpha_\mu)_{ij}\right)^2 + \frac{\epsilon_W}{2}\|W\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha_\mu\|^2 \,.$$

Denoting by $D = L - X\hat{\beta} - (V\hat{\gamma})^\top$, this problem can be rewritten as:

$$\min_{W,\alpha} \|D - W\alpha\|^2_{\mathcal{P}} + \frac{1}{2}\left(\epsilon_W \|W\|^2 + \epsilon_\alpha \|\alpha\|^2\right) ,$$

where $\|A\|^2_{\mathcal{P}} = \sum_{(i,j)\in\mathcal{P}} A^2_{ij}$. By Lemma 1, if K is large enough, one can first solve the convex optimization problem:

$$\hat{R} \in \arg \min_{R\,:\,\mathrm{rank}(R)\leq K} \|D - R\|^2_{\mathcal{P}} + \sqrt{\epsilon_W \epsilon_\alpha}\|R\|_* \tag{7}$$

and set

$$W = \left(\frac{\epsilon_\alpha}{\epsilon_W}\right)^{\frac{1}{4}} R_L R_\Sigma^{\frac{1}{2}} , \quad \alpha = \left(\frac{\epsilon_W}{\epsilon_\alpha}\right)^{\frac{1}{4}} R_\Sigma^{\frac{1}{2}} R_R ,$$

where $\hat{R} = R_L R_\Sigma R_R$ is the SVD of $\hat{R}$. This solution is exact when K is at least equal to the rank of the solution of the unconstrained problem (7), which we solve with the `softImpute::softImpute()` function [37]. If K is smaller, then (7) becomes a non-convex optimization problem whose global optimum may be challenging to find. In that case we also use the rank-constrained version of `softImpute::softImpute()` to obtain a good local optimum.

6. Estimate β_π, γ_π, and α_π by solving the regularized logistic regression problem:

$$\min_{(\beta_\pi,\gamma_\pi,\alpha_\pi)} \sum_{(i,j)} \Big[-\hat{Z}_{ij}(X\beta_\pi + (V\gamma_\pi)^\top + \hat{W}\alpha_\pi)_{ij} + \ln\left(1 + e^{(X\beta_\pi + (V\gamma_\pi)^\top + \hat{W}\alpha_\pi)_{ij}}\right)\Big] + \frac{\epsilon_\beta}{2}\|\beta_\pi\|^2 + \frac{\epsilon_\gamma}{2}\|\gamma_\pi\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha_\pi\|^2 . \tag{8}$$

This is a standard ridge logistic regression problem, but with a potentially huge design matrix, with up to nJ rows and $MJ + nL$ columns. To solve it efficiently, we alternate the estimation of β_π, γ_π, and α_π. Specifically, we initialize parameter values as:

$$\hat{\beta}_\pi \leftarrow 0 , \quad \hat{\gamma}_\pi \leftarrow 0 , \quad \hat{\alpha}_\pi \leftarrow 0$$

and repeat the following two steps a few times (or until convergence):

(a) Optimization in γ_π:

$$\hat{\gamma}_\pi \in \arg\min_{\gamma_\pi} \sum_{(i,j)} \Big[-\hat{Z}_{ij}(X\hat{\beta}_\pi + (V\gamma_\pi)^\top + \hat{W}\hat{\alpha}_\pi)_{ij} + \ln\left(1 + e^{(X\hat{\beta}_\pi + (V\gamma_\pi)^\top + \hat{W}\hat{\alpha}_\pi)_{ij}}\right)\Big] + \frac{\epsilon_\gamma}{2}\|\gamma_\pi\|^2 . \tag{9}$$

Note that this problem can be solved for each cell (i) independently and in parallel. When there is no gene covariate besides the constant intercept, the problem is easily solved by setting $(\hat{\gamma}_\pi)_i$ to the logit of the proportion of zeros in each cell.

(b) Optimization in β_π and α_π:

$$\left(\hat{\beta}_\pi, \hat{\alpha}_\pi\right) \in \arg \min_{(\beta_\pi,\alpha_\pi)} \sum_{(i,j)} \Big[-\hat{Z}_{ij}(X\beta_\pi + (V\hat{\gamma}_\pi)^\top + \hat{W}\alpha_\pi)_{ij} + \ln\left(1 + e^{(X\beta_\pi + (V\hat{\gamma}_\pi)^\top + \hat{W}\alpha_\pi)_{ij}}\right)\Big] + \frac{\epsilon_\beta}{2}\|\beta_\pi\|^2 + \frac{\epsilon_\alpha}{2}\|\alpha_\pi\|^2 . \tag{10}$$

7. Initialize $\hat{\zeta} = 0$.

Optimization

After initialization, we maximize locally the penalized log-likelihood by alternating optimization over the dispersion parameters and left and right-factors, iterating the following steps until convergence:

1. Dispersion optimization:
$$\hat{\zeta} \leftarrow \arg\max_{\zeta} \left\{ \ell(\hat{\beta}, \hat{\gamma}, \hat{W}, \hat{\alpha}, \zeta) - \frac{\epsilon_\zeta}{2} \mathrm{Var}(\zeta) \right\} .$$
To solve this problem, we start by estimating a common dispersion parameter for all the genes, by maximizing the objective function under the constraint that $\mathrm{Var}(\zeta) = 0$; in practice, we use a derivative-free one-dimensional optimization vector over the range $[-50, 50]$. We then optimize the objective function by a quasi-Newton optimization scheme starting from the constant solution found by the first step. To derive the gradient of the objective function used by the optimization procedure, note that the derivative of the NB log-density is:
$$\frac{\partial}{\partial\theta} \ln f_{NB}(y; \mu, \theta) = \Psi(y+\theta) - \Psi(\theta) + \ln\theta + 1 - \ln(\mu+\theta) - \frac{y+\theta}{\mu+\theta} ,$$
where $\Psi(z) = \Gamma'(z)/\Gamma(z)$ is the digamma function. We therefore get the derivative of the ZINB density as follows, for any $\pi \in [0, 1]$:
    - If $y > 0$, $f_{ZINB}(y; \mu, \theta, \pi) = (1-\pi) f_{NB}(y; \mu, \theta)$ therefore
    $$\frac{\partial}{\partial\theta} \ln f_{ZINB}(y; \mu, \theta) = \Psi(y+\theta) - \Psi(\theta) + \ln\theta + 1 - \ln(\mu+\theta) - \frac{y+\theta}{\mu+\theta} .$$
    - For $y = 0$, $\frac{\partial}{\partial\theta} \ln f_{NB}(0; \mu, \theta) = \ln\theta + 1 - \ln(\mu+\theta) - \frac{\theta}{\mu+\theta}$, therefore
    $$\frac{\partial}{\partial\theta} \ln f_{ZINB}(y; \mu, \theta) = \frac{\ln\theta + 1 - \ln(\mu+\theta) - \frac{\theta}{\mu+\theta}}{1 + \frac{\pi(\mu+\theta)^\theta}{(1-\pi)\theta^\theta}} .$$

   The derivative of the objective function w.r.t. ζ_j, for $j = 1, \ldots, J$, is then easily obtained by
$$\sum_{i=1}^{n} \theta_j \frac{\partial}{\partial\theta} \ln f_{ZINB}(y_{ij}; \mu_{ij}, \theta_j) - \frac{\epsilon_\zeta}{J-1} \left(\zeta_j - \frac{1}{J} \sum_{k=1}^{J} \zeta_k \right) .$$
(Note that the $J-1$ term in the denominator comes from the use of the unbiased sample variance statistic in the penalty for ζ.)

2. Left-factor (cell-specific) optimization:
$$\left(\hat{\gamma}, \hat{W}\right) \leftarrow \arg\max_{(\gamma, W)} \left\{ \ell(\hat{\beta}, \gamma, W, \hat{\alpha}, \hat{\zeta}) - \frac{\epsilon_\gamma}{2} \|\gamma^0\|^2 - \frac{\epsilon_W}{2} \|W\|^2 \right\} . \tag{11}$$
Note that this optimization can be performed independently and in parallel for each cell $i = 1, \ldots, n$. For this purpose, we consider a subroutine `solveZinbRegression`$(y, A_\mu, B_\mu, C_\mu, A_\pi, B_\pi, C_\pi, C_\theta)$ to find a set of vectors (a_μ, a_π, b) that locally maximize the log-likelihood of a ZINB model for a vector of counts y parametrized as follows:
$$\begin{aligned} \ln(\mu) &= A_\mu a_\mu + B_\mu b + C_\mu , \\ \mathrm{logit}(\pi) &= A_\pi a_\pi + B_\pi b + C_\pi , \\ \ln(\theta) &= C_\theta . \end{aligned}$$

We give more details on how to solve `solveZinbRegression` in the next section. To solve (11) for cell i we call `solveZinbRegression` with the following parameters:

$$\begin{cases} a_\mu &= \gamma_\mu[.,i] \\ a_\pi &= \gamma_\pi[.,i] \\ b &= W[i,.]^\top \\ y &= Y[i,]^\top \\ A_\mu &= V_\mu \\ B_\mu &= \alpha_\mu^\top \\ C_\mu &= (X_\mu[i,.]\beta_\mu + O_\mu[i,.])^\top \\ A_\pi &= V_\pi \\ B_\pi &= \alpha_\pi^\top \\ C_\pi &= (X_\pi[i,.]\beta_\pi + O_\pi[i,.])^\top \\ C_\theta &= \zeta \end{cases} .$$

3. Right-factor (gene-specific) optimization:

$$\left(\hat\beta, \hat\alpha\right) \leftarrow \arg\max_{(\beta,\alpha)} \left\{ \ell(\beta, \hat\gamma, \hat W, \alpha, \hat\zeta) - \frac{\epsilon_\beta}{2}\|\beta^0\|^2 - \frac{\epsilon_\alpha}{2}\|\alpha\|^2 \right\} .$$

Note that this optimization can be performed independently and in parallel for each gene $j = 1, \ldots, J$, by calling `solveZinbRegression` with the following paramters:

$$\begin{cases} a_\mu &= (\beta_\mu[.,j]; \alpha_\mu[.,j]) \\ a_\pi &= (\beta_\pi[.,j]; \alpha_\pi[.,j]) \\ b &= \emptyset \\ y &= Y[.,j] \\ A_\mu &= [X_\mu, W] \\ B_\mu &= \emptyset \\ C_\mu &= (V_\mu[j,.]\gamma_\mu)^\top + O_\mu[.,j] \\ A_\pi &= [X_\pi, W] \\ B_\pi &= \emptyset \\ C_\pi &= (V_\pi[j,.]\gamma_\pi)^\top + O_\pi[.,j] \\ C_\theta &= \zeta_j \mathbf{1}_n \end{cases} .$$

4. Orthogonalization:

$$\left(\hat W, \hat\alpha\right) \leftarrow \arg \min_{(W,\alpha)\,:\, W\alpha = \hat W \hat\alpha} \frac{1}{2}\left(\epsilon_W \|W\|^2 + \epsilon_\alpha \|\alpha\|^2\right) .$$

This is obtained by applying Lemma 1, starting from an SVD decomposition of the current $\hat W \hat\alpha$. Note that this step not only allows to maximize locally the penalized log-likelihood, but also ensures that the columns of W stay orthogonal to each other during optimization.

Solving `solveZinbRegression`

Given a N-dimensional matrix of counts $y \in \mathbb{N}^N$, and matrix $A_\mu \in \mathbb{R}^{N\times p}$, $B_\mu \in \mathbb{R}^{N\times r}$, $C_\mu \in \mathbb{R}^N$, $A_\pi \in \mathbb{R}^{N\times q}$, $B_\pi \in \mathbb{R}^{N\times r}$, $C_\pi \in \mathbb{R}^N$ and $C_\theta \in \mathbb{R}^N$, for some integers p, q, r, the function `solveZinbRegression`$(y, A_\mu, B_\mu, C_\mu, A_\pi, B_\pi, C_\pi, C_\theta)$ attempts to find parameters $(a_\mu, a_\pi, b) \in \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^r$ that maximize the ZINB log-likelihood of y with parameters:

$$\begin{aligned} \ln(\mu) &= A_\mu a_\mu + B_\mu b + C_\mu , \\ \text{logit}(\pi) &= A_\pi a_\pi + B_\pi b + C_\pi , \\ \ln(\theta) &= C_\theta . \end{aligned}$$

Starting from an initial guess (as explained in the different steps above), we perform a local minimization of this function $F(a_\mu, a_\pi, b)$ using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method. Let us now give more details on how the gradient of F is computed.

Given a single count y (i.e., $N = 1$), we first explicit the derivatives of the log-likelihood of y with respect to the (μ, π) parameters of the ZINB distribution. We first observe that

$$\frac{\partial}{\partial \mu} \ln f_{NB}(y; \mu, \theta) = \frac{y}{\mu} - \frac{y+\theta}{\mu+\theta} \,,$$

and that by definition of the ZINB distribution the following holds:

$$\begin{aligned} \frac{\partial}{\partial \mu} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{(1-\pi) f_{NB}(y; \mu, \theta) \frac{\partial}{\partial \mu} \ln f_{NB}(y; \mu, \theta)}{f_{ZINB}(y; \mu, \theta, \pi)} \,, \\ \frac{\partial}{\partial \pi} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{\delta_0(y) - f_{NB}(y; \mu, \theta)}{f_{ZINB}(y; \mu, \theta, \pi)} \,. \end{aligned}$$

Let us explicit these expressions, depending on whether or not y is null:

- If $y > 0$, then $\delta_0(y) = 0$ and $f_{ZINB}(y; \mu, \theta, \pi) = (1-\pi) f_{NB}(y; \mu, \theta)$, so we obtain:

$$\begin{aligned} \frac{\partial}{\partial \mu} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{y}{\mu} - \frac{y+\theta}{\mu+\theta} \,, \\ \frac{\partial}{\partial \pi} \ln f_{ZINB}(y; \mu, \theta, \pi) &= \frac{-1}{1-\pi} \,. \end{aligned}$$

- If $y = 0$, then $\delta_0(y) = 0$, and we get

$$\begin{aligned} \frac{\partial}{\partial \mu} \ln f_{ZINB}(0; \mu, \theta, \pi) &= \frac{-(1-\pi)\left(\frac{\theta}{\mu+\theta}\right)^{\theta+1}}{\pi + (1-\pi)\left(\frac{\theta}{\mu+\theta}\right)^{\theta}} \,, \\ \frac{\partial}{\partial \pi} \ln f_{ZINB}(0; \mu, \theta, \pi) &= \frac{1 - \left(\frac{\theta}{\mu+\theta}\right)^{\theta}}{\pi + (1-\pi)\left(\frac{\theta}{\mu+\theta}\right)^{\theta}} \,. \end{aligned}$$

When $N \geq 1$, using standard calculus for the differentiation of compositions and the facts that:

$$\begin{aligned} \left(\ln^{-1}\right)' (\ln \mu) &= \mu \,, \\ \left(\text{logit}^{-1}\right)' (\text{logit}\pi) &= \pi(1-\pi) \,, \end{aligned}$$

we finally get that

$$\begin{aligned} \nabla_{a_\mu} F &= A_\mu^\top G \,, \\ \nabla_{a_\pi} F &= A_\pi^\top H \,, \\ \nabla_b F &= B_\mu^\top G + B_\pi^\top H \,. \end{aligned}$$

where G and H are the N-dimensional vectors given by

$$\begin{aligned} \forall i \in [1, N], \quad G_i &= \mu_i \frac{\partial}{\partial \mu} \ln f_{ZINB}(y_i; \mu_i, \theta_i, \pi_i) \,, \\ H_i &= \pi_i (1-\pi_i) \frac{\partial}{\partial \pi} \ln f_{ZINB}(y_i; \mu_i, \theta_i, \pi_i) \,. \end{aligned}$$

Software implementation

The ZINB-WaVE method is implemented in the R package `zinbwave`, available from the GitHub repository `https://github.com/drisso/zinbwave`. See the package vignette for a detailed example of a typical use. The code to reproduce all the analyses and figures of this article is available at `https://github.com/drisso/zinb_analysis`.

Real datasets

V1 dataset. Tasic *et al.* [3] characterized more than 1,600 cells from the primary visual cortex (V1) in adult male mice, using a set of established Cre lines. Single cells were isolated by FACS into 96-well plates and RNA was reverse transcribed and amplified using the SMARTer kit. Sequencing was performed using the Illumina HiSeq platform, yielding 100 bp-long reads. We selected a subset of three Cre lines, Ntsr1-Cre, Rbp4-Cre, and Scnn1a-Tg3-Cre, that label Layer 4, Layer 5, and Layer 6 excitatory neurons, respectively. This subset consists of 379 cells, grouped by the authors into 17 clusters; we excluded the cells that did not pass the authors' quality control filters and that were classified by the authors as "intermediate" cells between two clusters, retaining a total of 285 cells. Gene expression was quantified by gene-level read counts. Raw gene-level read counts and QC metrics (see below) are available as part of the scRNAseq Bioconductor package (`https://bioconductor.org/packages/scRNAseq`). We applied the dimensionality reduction methods to the 1,000 most variable genes.

S1/CA1 dataset. Zeisel *et al.* [4] characterized 3,005 cells from the primary somatosensory cortex (S1) and the hippocampal CA1 region, using the Fluidigm C1 microfluidics cell capture platform followed by Illumina sequencing. Gene expression was quantified by unique molecular identifier (UMI) counts. In addition to gene expression measures, we have access to metadata that can be used to assess the methods: batch, sex, number of mRNA molecules. Raw UMI counts and metadata were downloaded from `http://linnarssonlab.org/cortex/`.

mESC dataset. Kolodziejczyk *et al.* [30] sequenced the transcriptome of 704 mouse embryonic stem cells (mESCs), across three culture conditions (serum, 2i, and a2i), using the Fluidigm C1 system followed by Illumina sequencing. We selected only the cells from the second and third batch, after excluding the samples that did not pass the authors' QC filters. This allowed us to have cells from each culture condition in each batch and resulted in a total of 169 serum cells, 141 2i cells, and 159 a2i cells. In addition to gene expression measures, we have access to batch and plate information that can be included as covariates in our model. Raw gene-level read counts were downloaded from `http://www.ebi.ac.uk/teichmann-srv/espresso/`. Batch and plate information was extracted from the sample names, as done in Lun & Marioni [31]. We applied the dimensionality reduction methods to the 1,000 most variable genes.

Glioblastoma dataset. Patel *et al.* [6] collected 672 cells from five dissociated human glioblastomas. Transcriptional profiles were generated using the SMART-Seq protocol. We analyzed only the cells that passed the authors' QC filtering. The raw data were downloaded from the NCBI GEO database (accession GSE57872). Reads were aligned using TopHat with the following parameters: –rg-library Illumina –rg-platform Illumina –keep-fasta-order -G -N 3 –read-edit-dist 3 –no-coverage-search -x 1 -M -p 12. Counts were obtained using `htseq-count` with the following parameters (`http://www-huber.embl.de/HTSeq/doc/count.html`): -a 10 -q -s no -m union. We applied the dimensionality reduction methods to the 1,000 most variable genes.

Simulated datasets

Simulating from the ZINB-WaVE model

Setting the simulation parameters. In order to simulate realistic data, we fitted our ZINB-WaVE model to two real datasets (V1 and S1/CA1) and used the resulting parameter estimates as the truth to be estimated in the simulation. Genes that did not have at least 5 reads in at least 5 cells were filtered out and J ($J = 1,000$) genes were then sampled at random for each dataset. The ZINB-WaVE model was fitted to the count matrix Y with the number of unknown cell-level covariates set to $K = 2$, genewise dispersion ($\zeta = \ln\theta = -\ln\phi$), X_μ, X_π, V_μ, and V_π as columns of ones (i.e., intercept only), and no offset matrices, to get estimates for W, α_μ, α_π, β_μ, β_π, γ_μ, γ_π, and ζ. The parameters which were varied in the simulations are as follows.

- The number of cells: $n = 100, 1,000, 10,000$.
- The proportion of zero counts, $zfrac = \sum_{i,j} 1(Y_{ij} = 0)/nJ$, via the parameter γ_π: $zfrac \approx 0.25, 0.50, 0.75$. Since $\mathrm{logit}(\pi) = X\beta_\pi + (V\gamma_\pi)^T + W\alpha_\pi$, the value of γ_π is directly linked to the dropout probability π, thus to the zero fraction. Note that by changing only γ_π but not γ_μ, we change the dropout rate but not the underlying, unobserved mean expression; i.e., this increases the number of *technical* zeros but not *biological* zeros.
- The ratio of within to between-cluster sums of squares for W. Specifically, let C denote the number of clusters and n_c the number of cells in cluster c. For a given column of W (out of K columns), let W_{ic} denote the value for cell i in cluster c, $\bar{W}$ the overall average across all n cells, $\bar{W}_c$ the average for cells in cluster c, and TSS the total sum of squares. Then,

$$\begin{aligned} TSS &= \sum_{c=1}^{C}\sum_{i=1}^{n_c}(W_{ic} - \bar{W})^2 \\ &= \sum_{c=1}^{C}\sum_{i=1}^{n_c}(W_{ic} - \bar{W}_c)^2 + \sum_{c=1}^{C} n_c(\bar{W}_c - \bar{W})^2 \\ &= WSS + BSS, \end{aligned}$$

  with $WSS = \sum_{c=1}^{C}\sum_{i=1}^{n_c}(W_{ic} - \bar{W}_c)^2$ and $BSS = \sum_{c=1}^{C} n_c(\bar{W}_c - \bar{W})^2$ the within and between-cluster sums of squares, respectively. The level of difficulty of the clustering problem can be controlled by the ratio of within to between-cluster sums of squares. However, we want to keep the overall mean $\bar{W}$ and overall variance (i.e., TSS) constant, so that the simulated values of W stay in the same range as the estimated W from the real dataset; this prevents us from simulating an unrealistic count matrix Y.

  Let us scale the between-cluster sum of squares by a^2 and the within-cluster sum of squares by a^2b^2, i.e., replace $(\bar{W}_c - \bar{W})$ by $a(\bar{W}_c - \bar{W})$ and $(W_{ic} - \bar{W}_c)$ by $ab(W_{ic} - \bar{W}_c)$, with $a \geq 0$ and $b \geq 0$ such that TSS and $\bar{W}$ are constant. The total sum of squares TSS remains constant, i.e.,

$$TSS = a^2b^2WSS + a^2BSS,$$

  provided

$$a^2 = \frac{TSS}{b^2WSS + BSS}.$$

  Requiring the overall mean $\bar{W}$ to remain constant implies that

$$\bar{W}_c^* = (1-a)\bar{W} + a\bar{W}_c.$$

  Thus,

$$\begin{aligned} W_{ic}^* &= \bar{W}_c^* + ab(W_{ic} - \bar{W}_c) \\ &= (1-a)\bar{W} + a\bar{W}_c + ab(W_{ic} - \bar{W}_c) \\ &= (1-a)\bar{W} + a(1-b)\bar{W}_c + abW_{ic}. \end{aligned} \tag{12}$$

  The above transformation results in a scaling of the ratio of within to between-cluster sums of squares by b^2, while keeping the overall mean and variance constant.

  In our simulations, we fixed $C = 3$ clusters and considered three values for b^2, where the same value of b^2 is applied to each $k \in \{1, \ldots, K\}$: $b^2 = 1$, corresponding to the case where the within and between-cluster sums of squares are the same as the ones of the fitted W from the real dataset; $b^2 = 5$, corresponding to a larger ratio of within to between-cluster sums of squares and hence a harder clustering problem; $b^2 = 10$, corresponding to a very large ratio of within to between-cluster sums of squares and hence almost no clustering.

Overall, 2 (real datasets) $\times$ 3 (n) $\times$ 3 ($zfrac$) $\times$ 3 (clustering) = 54 scenarios were considered in the simulation.

Simulating the datasets. For each of the 54 scenarios, we simulated $B = 10$ datasets, resulting in a total of $54 \times 10 = 540$ datasets. Using the fitted W, α_μ, α_π, β_μ, β_π, γ_μ, γ_π, and ζ from one of the two real datasets, the datasets were simulated according to the following steps.

1. Simulate W with desired clustering strength. First fit a K-variate Gaussian mixture distribution to W inferred from one of the real datasets using the R function `Mclust` from the mclust package and specifying the number of clusters C. Then, for each of $B = 10$ datasets, simulate W cluster by cluster from K-variate Gaussian distributions using the `mvrnorm` function from the MASS package, with the cluster means, covariance matrices, and frequencies output by `Mclust`. Transform W as in Equation (12) to get the desired ratio of within to between-cluster sums of squares.
2. Simulate γ_μ and γ_π to get the desired zero fraction. We only considered cell-level intercept n-vectors γ_μ and γ_π, i.e., $L = 1$ and a matrix V of gene-level covariates consisting of a single column of ones. As the fitted γ_μ and γ_π from the original datasets are correlated n-vectors, fit a bivariate Gaussian distribution to γ_μ and γ_π using the function `Mclust` from the mclust package with $C = 1$ cluster. Then, for each of $B = 10$ datasets, simulate γ_μ and γ_π from a bivariate Gaussian distribution using the `mvrnorm` function from the MASS package, with the mean and covariance matrix output by `Mclust`. To increase/decrease the zero fraction, increase/decrease each element of the mean vector for γ_π inferred from `Mclust` (shifts of $\{0, 2, 5\}$ for V1 dataset and $\{-1.5, 0.5, 2\}$ for S1/CA1 dataset).
3. Create the ZINB-WaVE model using the function `zinbModel` from the package zinbwave.
4. Simulate counts using the function `zinbSim` from the package zinbwave.

Simulating from the Lun & Marioni [31] model

To simulate datasets from a different model than our ZINB-WaVE model, we simulated counts using the procedure described in Lun & Marioni [31] (details in Supplementary Materials of original publication and code available from the Github repository `https://github.com/MarioniLab/PlateEffects2016`). Although the Lun & Marioni [31] model is also based on a zero-inflated negative binomial distribution, the distribution is parameterized differently and also fit differently, gene by gene. In particular, the negative binomial mean is parameterized as a product of the expression level of interest and two nuisance technical effects, a gene-level effect assumed to have a log-normal distribution and a cell-level effect (cf. library size) whose distribution is empirically derived. The zero inflation probability is assumed to be constant across cells for each gene and is estimated independently of the negative binomial mean. The ZINB distribution is fit gene by gene using the `zeroinfl` function from the R package pscl. We used the raw gene-level read counts for the mESC dataset as input to the script `reference/submitter.sh`, to create a simulation function constructed by fitting a ZINB distribution to these counts. The script `simulations/submitter.sh` was then run to simulate counts based on the estimated parameters (negative binomial mean, zero inflation probability, and genewise dispersion parameter). We simulated $C = 3$ clusters with equal number of cells per cluster. The parameters which were varied in the simulations are as follows.

- The number of cells: $n = 100, 1,000, 10,000$.
- The proportion of zero counts, $zfrac = \sum_{i,j} 1(Y_{ij} = 0)/nJ$, via the zero inflation probability: $zfrac \approx 0.4, 0.6, 0.8$. For $zfrac = 0.4$, we did not modify the code in Lun & Marioni [31]. However, to simulate datasets with greater zero fractions, namely $zfrac = 0.6$ and $zfrac = 0.8$, we added respectively 0.3 and 0.6 to the zero inflation probability (p'_i, in their notation).

For each of the 3 (n) $\times$ 3 ($zfrac$) = 9 scenarios, we simulated B ($B = 10$) datasets, resulting in $9 \times 10 = 90$ simulated datasets in total.

Dimensionality reduction methods

Four different methods were applied to the real and simulated datasets. For all the methods, we selected $K = 2$ components, unless specified otherwise. A notable exception is the S1/CA1 dataset, for which, given the large number of cells and the complexity of the signal, we specified $K = 3$ components.

ZINB-WaVE

We applied the ZINB-WaVE procedure using the function `zinbFit` from our R package zinbwave, with the following parameter choices.

- Number of unknown cell-level covariates K: $K = 1, 2, 3, 4$.
- Gene-level covariate matrix V: not included or set to a column of ones $\mathbf{1}_J$.
- Cell-level covariate matrix X: set to a column of ones $\mathbf{1}_n$. For the mESC dataset, we also considered including batch covariates in X.
- Dispersion parameter ζ: ζ the same for all genes (common dispersion) or specific to each gene (genewise dispersion).

Negative binomial fit

For the V1 dataset, after full-quantile normalization (function `betweenLaneNormalization` from the R package EDASeq [38]), we used the R package edgeR [39] (Version 3.16.5) to fit a negative binomial distribution gene by gene with only an intercept (i.e., default value for the `design` argument as a single column of ones) and genewise dispersion. The parameters estimated by edgeR are

- the mean $E[Y_{ij}] = \mu_{ij}$ for the NB distribution, for each cell i and gene j;
- the dispersion parameter $\exp(-\zeta_j)$, for each gene j.

Zero-inflated factor analysis

We used the zero-inflated factor analysis (ZIFA) method [26], as implemented in the `ZIFA` python package (Version 0.1) available at `https://github.com/epierson9/ZIFA`, with the block algorithm (function `block_ZIFA.fitModel`, with default parameters). The output of ZIFA is an $n \times K$ matrix corresponding to a projection of the counts onto a latent low-dimensional space of dimension K, where we chose $K = 2$.

Principal component analysis

We used the function `prcomp` from the R package stats for the simulation study and, for computational efficiency, the function `jsvds` from the R package rARPACK for the real datasets.

Normalization

As normalization is essential, especially for zero-inflated distributions, PCA and ZIFA were applied to both raw and normalized counts. The following normalization methods were used.

- Total-count normalization (TC). Counts are divided by the total number of reads sequenced for each sample and multiplied by the mean total count across all the samples. This method is related to the popular Transcripts Per Million (TPM) [40] and Fragments Per Kilobase Million (FPKM) [41] methods.
- Full-quantile normalization (FQ) [42]. The quantiles of the distributions of the gene-level read counts are matched across samples. We used the function `betweenLaneNormalization` from the R package EDASeq.
- Trimmed mean of M values (TMM) [43]. The TMM global-scaling factor is computed as the weighted mean of log-ratios between each sample and a reference sample. If the majority of the genes are not differentially expressed (DE), TMM should be close to 1, otherwise, it provides an estimate of the correction factor that must be applied to the library sizes in order to fulfill this hypothesis. We used the function `calcNormFactors` from edgeR to compute these scaling factors.

Evaluation criteria

Bias and MSE of the ZINB-WaVE estimators

For data simulated from the ZINB-WaVE model, let θ and $\hat{\theta}_b$, $b = 1, \ldots, B$, respectively denote the true parameter and an estimator of this parameter for the b^{th} simulated dataset. Then, for each scenario, the performance of the estimator $\hat{\theta}$ can be assessed in terms of bias and mean squared error (MSE) as follows:

$$\begin{aligned} \text{Bias} &= \frac{1}{B}\sum_{b=1}^{B}(\hat{\theta}_b - \theta), \\ \text{MSE} &= \frac{1}{B}\sum_{b=1}^{B}(\hat{\theta}_b - \theta)^2. \end{aligned}$$

Bias and MSE were computed for β_μ, β_π, γ_μ, γ_π, ζ, $W\alpha_\mu$, $W\alpha_\pi$, $\ln(\mu)$, and $\text{logit}(\pi)$. When the parameter to be estimated was an $n \times J$ matrix, the matrix was converted to a $1 \times nJ$ row vector.

Goodness-of-fit of the ZINB-WaVE model

For the ZINB-WaVE model, the overall mean, variance, and zero probability are

$$\begin{aligned} E[Y_{ij}] &= (1-\pi_{ij})\mu_{ij}, \\ \text{Var}(Y_{ij}) &= (1-\pi_{ij})\mu_{ij}(1+\mu_{ij}(\phi_j+\pi_{ij})), \\ P(Y_{ij}=0) &= \pi_{ij} + (1-\pi_{ij})(1+\phi_j\mu_{ij})^{\frac{1}{\phi_j}}. \end{aligned}$$

When $\pi_{ij} = 0$, the ZINB model reduces to a negative binomial model. We compared the goodness-of-fit of the ZINB and NB models on real data, using mean-difference plots of estimated vs. observed mean count and zero probability, as well as plots of the estimated dispersion parameter against the observed zero fraction.

Correlation of pairwise distances between observations

For simulated data, we assessed different dimensionality reduction methods (ZINB-WaVE, PCA, and ZIFA) in terms of the correlation between pairwise distances between observations in the true and in the estimated reduced-dimensional space. In particular, we monitored the influence of the number of unknown cell-level covariates K when the other parameters are set correctly (V is a column of ones $\mathbf{1}_J$ and genewise dispersion). For each simulation scenario, the correlation between true and estimated pairwise distances was computed and averaged over B datasets.

Silhouette width

Given a set of labels for the samples (e.g., biological condition, batch) silhouette widths provide a measure of goodness of the clustering of the samples with respect to these labels. Silhouette widths may be averaged within clusters or across all observations to assess clustering strength at the level of clusters or overall, respectively. The silhouette width s_i of a sample i is defined as follows:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}},$$

where $a_i = d(i, C_{cl(i)})$, $b_i = \min_{l \neq cl(i)} d(i, C_l)$, $C_{cl(i)}$ is the cluster to which i belongs, and $d(i, C_l)$ is the average distance between sample i and the samples in cluster C_l.

Average silhouette widths were used to compare ZINB-WaVE, PCA, and ZIFA on both real and simulated datasets. For simulated data, the cluster labels correspond to the true simulated W and, for each scenario, silhouette widths were computed and averaged over B datasets. For real data, the authors' cluster labels or known cell types were used.

Correlation with QC measures

To evaluate the dependence of the inferred low-dimensional signal on unwanted variation, we computed the absolute correlation between each dimension (e.g., principal component) and a set of quality control (QC) measures. For the V1 dataset, FastQC (`http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`) and Picard tools (`https://broadinstitute.github.io/picard/`) were used to compute a set of 16 QC measures. These measures are available as part of the scRNAseq Bioconductor package (`https://bioconductor.org/packages/scRNAseq`). For the Glioblastoma and mESC datasets, we used the scater Bioconductor package [44] (Version 1.2.0) to compute a set of 7 QC measures. For the S1/CA1 dataset, we used a set of 6 QC measures provided by the authors (`http://linnarssonlab.org/cortex/`).

Acknowledgments

We would like to thank Olivier Mirabeau from the Curie Institute for the pre-processing of the Glioblastoma dataset. We also thank Aaron Lun and the Marioni Lab for making the code needed for the simulations available online. DR and SD are supported by the National Institutes of Health BRAIN Initiative (grant U01 MH105979, PI: John Ngai). SG and JPV are supported by the French National Research Agency (grant ABS4NGS ANR-11-BINF-0001). JPV is supported by the European Research Council (grant ERC-SMAC-280032), the Miller Institute for Basic Research in Science, and the Fulbright Foundation.

Contributions

DR, SD and JPV formulated the statistical model. SG and JPV conceived and implemented the optimization algorithm. DR, SG, and JPV wrote the R package. DR performed the real data analysis. DR, FP, SD, and JPV designed the simulation study. FP performed the simulations and analyzed the simulated data. DR, FP, SD, and JPV wrote the manuscript.

References

1. Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C. & Teichmann, S. A. The technology and biology of single-cell RNA sequencing. *Molecular Cell* **58,** 610–620 (2015).
2. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161,** 1202–1214 (2015).
3. Tasic, B. *et al.* Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature Neuroscience* **19,** 335–346 (2016).
4. Zeisel, A. *et al.* Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347,** 1138–42 (2015).
5. Deng, Q., Ramsköld, D., Reinius, B. & Sandberg, R. Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science* **343,** 193–6 (2014).
6. Patel, A. P. *et al.* Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* **344,** 1396–1401 (2014).
7. Bacher, R. & Kendziorski, C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biology* **17,** 1 (2016).
8. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nature methods* **11,** 740–2 (2014).
9. Islam, S. *et al.* Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature methods* **11,** 163–166 (2014).
10. Tung, P.-Y. *et al.* Batch effects and the effective design of single-cell gene expression studies. *Scientific Reports* **7,** 39921 (Jan. 2017).
11. Vallejos, C. A., Risso, D., Scialdone, A., Dudoit, S. & Marioni, J. C. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature Methods,* Under review (2017).
12. Marinov, G. K. *et al.* From single-cell to cell-pool transcriptomes: stochasticity in gene expression and RNA splicing. *Genome research* **24,** 496–510 (Mar. 2014).
13. Pollen, A. A. *et al.* Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology* **32,** 1053–8 (2014).
14. Buettner, F. *et al.* Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology* **33,** 155–160 (2015).
15. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32,** 381–6 (2014).
16. Ji, Z. & Ji, H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Research* **44,** e117 (2016).
17. Shin, J. *et al.* Single-Cell RNA-Seq with Waterfall Reveals Molecular Cascades underlying Adult Neurogenesis. *Cell Stem Cell* **17,** 360–372 (2015).
18. Campbell, K., Ponting, C. P. & Webber, C. Laplacian eigenmaps and principal curves for high resolution pseudotemporal ordering of single-cell RNA-seq profiles. *bioRxiv,* 027219 (2015).
19. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology* **33,** 495–502 (2015).
20. Shalek, A. K. *et al.* Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* **510,** 363–9 (2014).
21. Gaublomme, J. T. *et al.* Single-Cell Genomics Unveils Critical Regulators of Th17 Cell Pathogenicity. *Cell* **163,** 1400–1412 (2015).
22. Hicks, S. C., Teng, M. & Irizarry, R. A. On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data. *bioRxiv,* 025528 (2015).
23. Finak, G. *et al.* MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biology* **16,** 1 (2015).

24. Belkin, M. & Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation* **15,** 1373–1396 (2003).

25. Van Der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).

26. Pierson, E. & Yau, C. Dimensionality reduction for zero-inflated single cell gene expression analysis. *Genome Biology* **16** (2015).

27. Gagnon-Bartsch, J. a. & Speed, T. P. Using control genes to correct for unwanted variation in microarray data. *Biostatistics* **13,** 539–52 (July 2012).

28. Risso, D., Ngai, J., Speed, T. P. & Dudoit, S. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nat Biotech* **32,** 896–902 (2014).

29. Cole, M. & Risso, D. scone: Single Cell Overview of Normalized Expression data. *R package* (2016).

30. Kolodziejczyk, A. A. *et al.* Single Cell RNA-Sequencing of Pluripotent States Unlocks Modular Transcriptional Variation. *Cell Stem Cell* **17,** 471–485 (2015).

31. Lun, A. T. L. & Marioni, J. C. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. *bioRxiv* (2016).

32. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8,** 118–127 (2007).

33. Leek, J. T. & Storey, J. D. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet* **3,** e161 (2007).

34. Lin, Z. *et al.* Simultaneous dimension reduction and adjustment for confounding variation. *Proceedings of the National Academy of Sciences* **113,** 14662–14667 (2016).

35. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Meth* **advance on** (Mar. 2017).

36. Srebro, N., Rennie, J. D. M. & Jaakkola, T. S. Maximum-Margin Matrix Factorization. *Advances in Neural Information Processing Systems 17,* 1329–1336 (2005).

37. Mazumder, R., Hastie, T. & Tibshirani, R. *Spectral Regularization Algorithms for Learning Large Incomplete Matrices.* 2010.

38. Risso, D., Schwartz, K., Sherlock, G. & Dudoit, S. GC-content normalization for RNA-Seq data. *BMC Bioinformatics* **12,** 480 (2011).

39. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26,** 139 (2010).

40. Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* **12,** 323 (2011).

41. Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology* **28,** 511–515 (2010).

42. Bullard, J. H., Purdom, E., Hansen, K. D. & Dudoit, S. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics* **11,** 94 (2010).

43. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* **11,** R25 (2010).

44. Mccarthy, D. J., Campbell, K. R., Lun, A. T. L. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics,* 1–8 (2017).

Supplementary figures

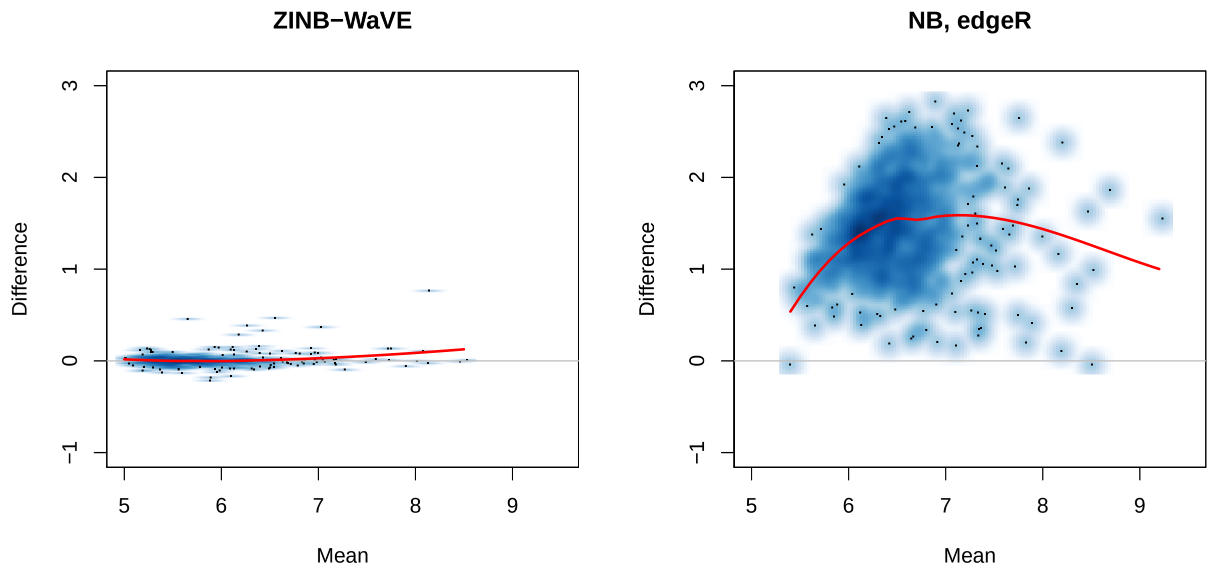


Figure S1: *Goodness-of-fit of ZINB and NB models: Mean-difference plot of estimated vs. observed mean count, V1 dataset.* Left panel: the estimated count from ZINB-WaVE for cell i and gene j is $(1 - \hat{\pi}_{ij})\hat{\mu}_{ij}$. Right panel: the estimated count from edgeR's NB model for cell i and gene j is $\hat{\mu}_{ij}$. Observed and estimated counts were averaged over n cells and plotted on a log scale.

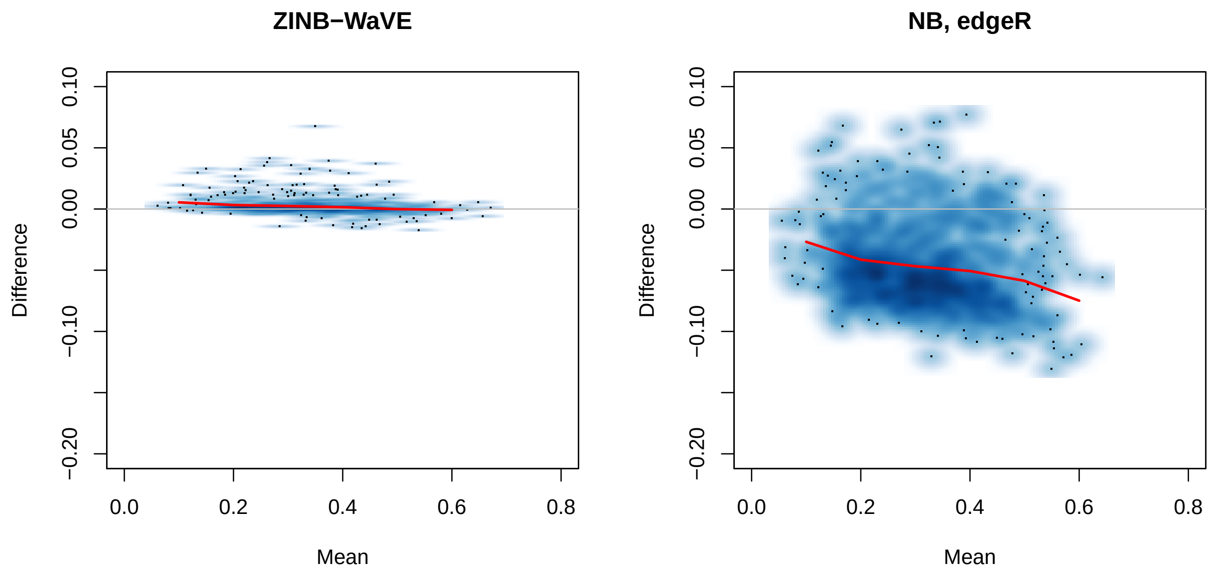


Figure S2: *Goodness-of-fit of ZINB and NB models: Mean-difference plot of estimated vs. observed zero probability, V1 dataset.* Left panel: the estimated zero probability from ZINB-WaVE for cell i and gene j is $\hat{\pi}_{ij} + (1 - \hat{\pi}_{ij})(1 + \hat{\phi}_j \hat{\mu}_{ij})^{\frac{1}{\hat{\phi}_j}}$. Right panel: the estimated zero probability from edgeR's NB model for cell i and gene j is $(1 + \hat{\phi}_j \hat{\mu}_{ij})^{\frac{1}{\hat{\phi}_j}}$. Observed and estimated zero probabilities were averaged over n cells.

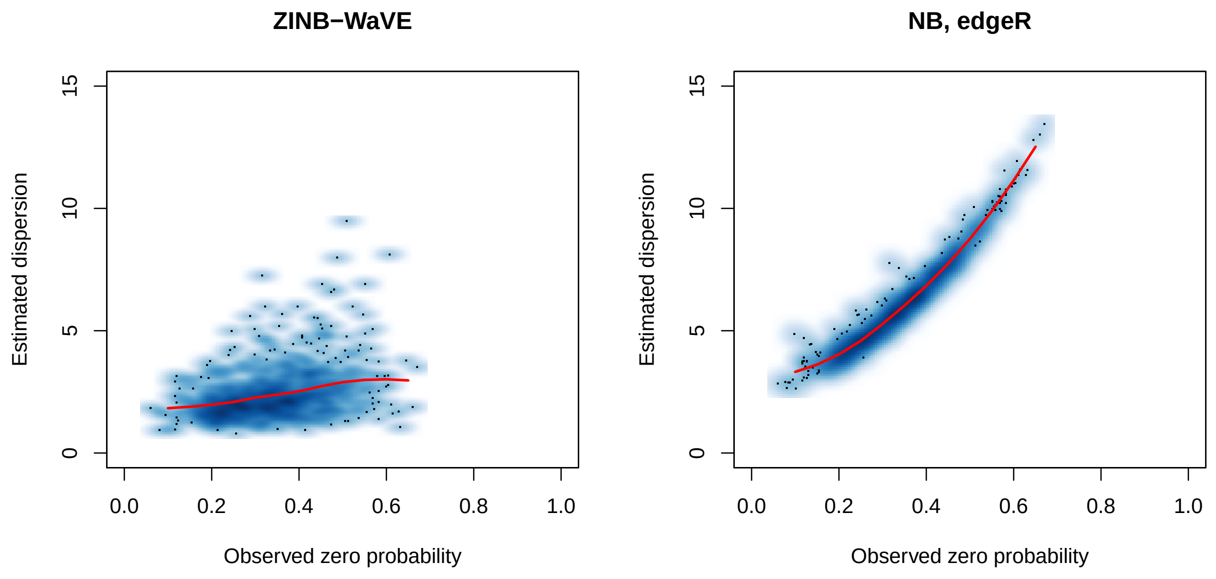


Figure S3: *Goodness-of-fit of ZINB and NB models: Estimated dispersion parameter vs. observed proportion of zero counts, V1 dataset.* Left panel: genewise dispersion parameters ϕ_j estimated using ZINB-WaVE. Right panel: genewise dispersion parameters ϕ_j estimated using edgeR's NB procedure.

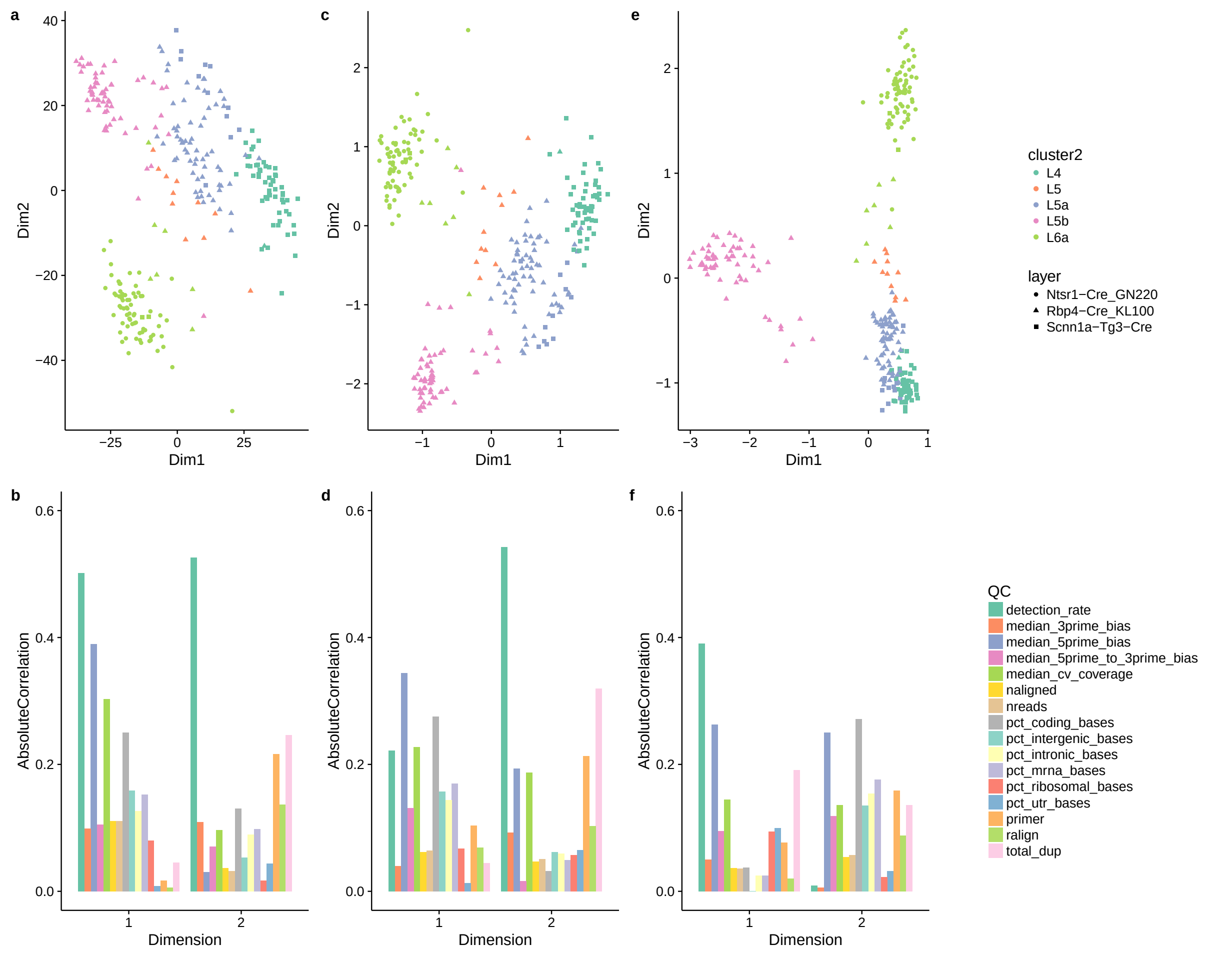


Figure S4: *Low-dimensional representation of the V1 dataset.* Upper panels provide two-dimensional representations of the data, after selecting the 1,000 most variable genes. Lower panels provide barplots of the absolute correlation between the first two components and a set of QC measures (see Methods). **(a, b)** PCA (on TC-normalized counts); **(c, d)** ZIFA (on TC-normalized counts); **(e, f)** ZINB-WaVE (no normalization needed). ZINB-WaVE leads to a low-dimensional representation that is less influenced by technical variation and to tighter, biologically meaningful clusters.

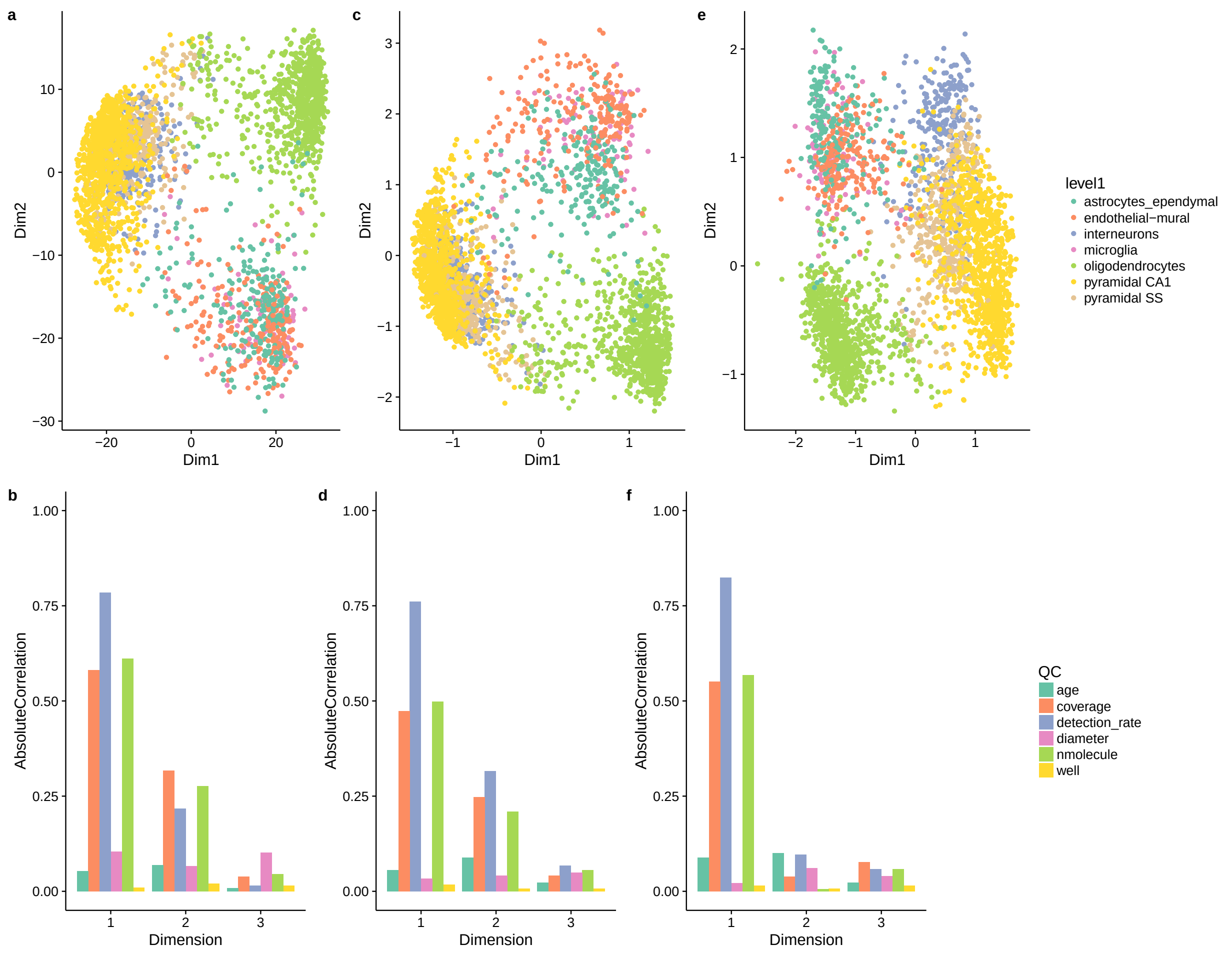


Figure S5: *Low-dimensional representation of the S1/CA1 dataset.* Upper panels provide two-dimensional representations of the data. Lower panels provide barplots of the absolute correlation between the first three components and a set of QC measures (see Methods). **(a, b)** PCA (on TC-normalized counts); **(c, d)** ZIFA (on TC-normalized counts); **(e, f)** ZINB-WaVE (no normalization needed). ZINB-WaVE leads to a low-dimensional representation that is less influenced by technical variation and to tighter, biologically meaningful clusters.

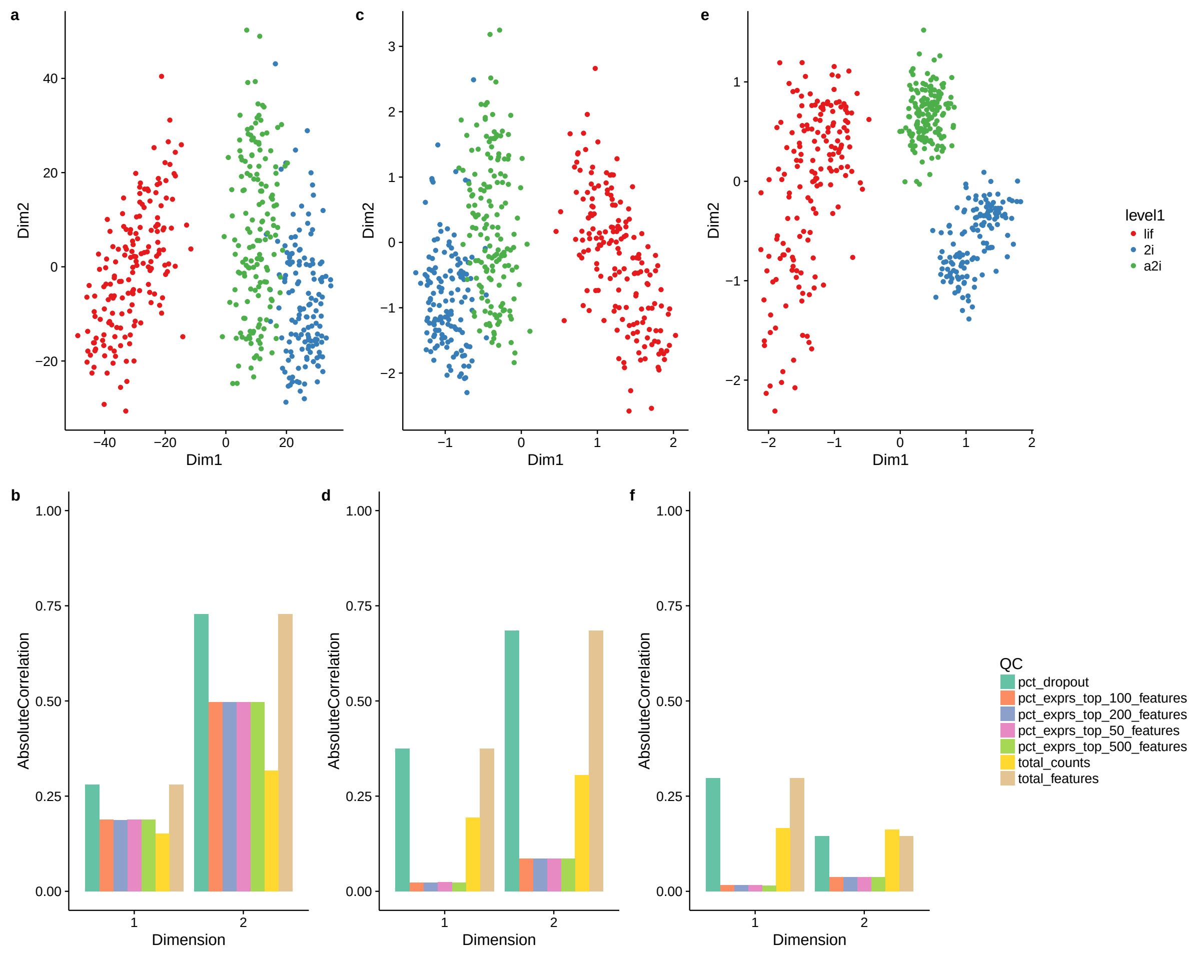


Figure S6: *Low-dimensional representation of the mESC dataset.* Upper panels provide two-dimensional representations of the data, after selecting the 1,000 most variable genes. Lower panels provide barplots of the absolute correlation between the first two components and a set of QC measures (see Methods). **(a, b)** PCA (on TC-normalized counts); **(c, d)** ZIFA (on TC-normalized counts); **(e, f)** ZINB-WaVE (no normalization needed). ZINB-WaVE leads to a low-dimensional representation that is less influenced by technical variation and to tighter, biologically meaningful clusters.

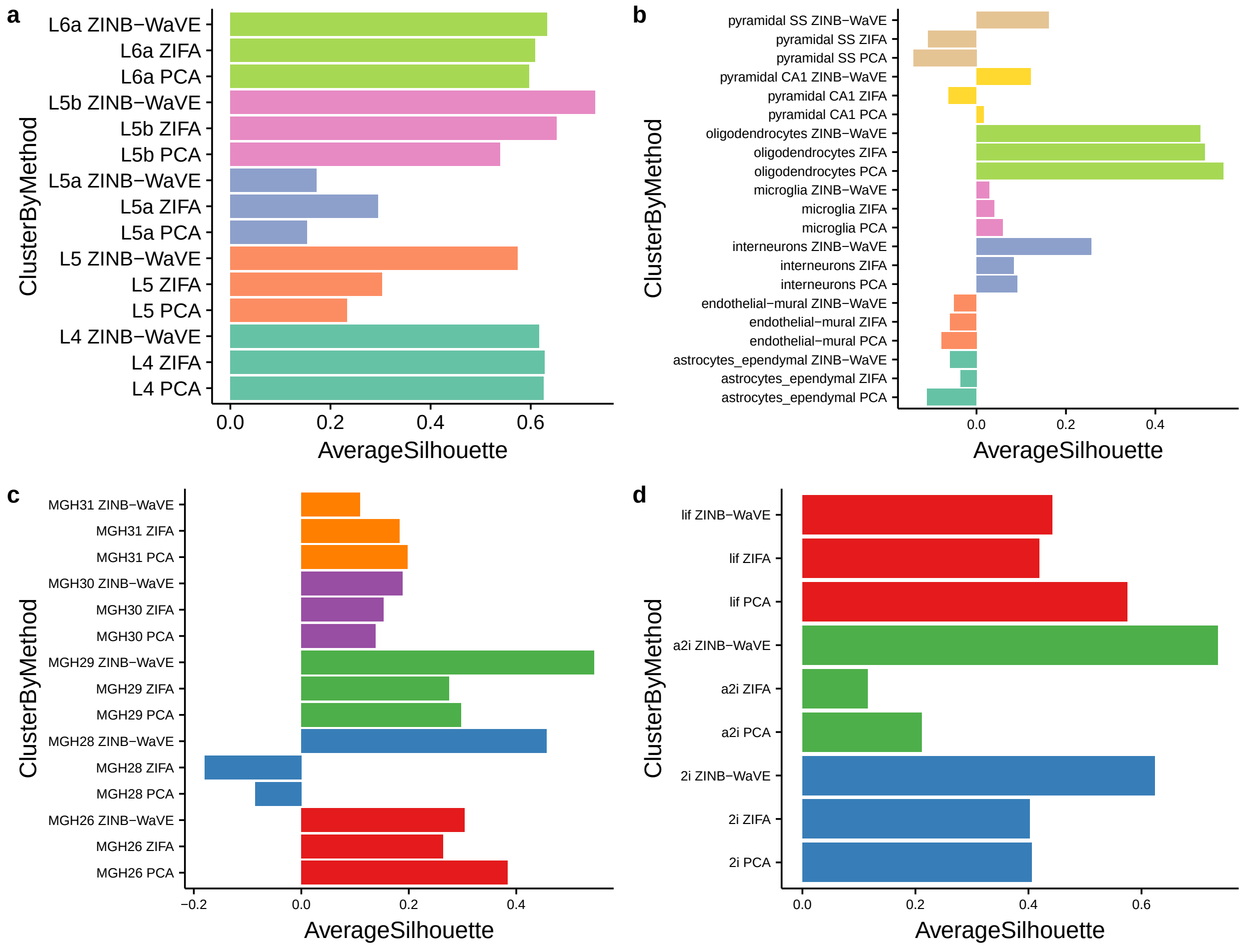


Figure S7: *Per-cluster average silhouette widths, real datasets.* **(a)** V1 dataset; **(b)** S1/CA1 dataset; **(c)** mESC dataset; **(d)** Glioblastoma dataset. For each of the four scRNA-seq datasets of Figures S4, S5, S6, and 2, barplots of the per-cluster average silhouette widths for ZINB-WaVE, ZIFA, and PCA (the best normalization method was used for ZIFA and PCA). Silhouette widths were computed in the low-dimensional space, using the groupings provided by the authors of the original publications: unsupervised clustering procedure **(a-b)**, observed characteristics of the samples, such as culture condition **(c)** and patient **(d)** .

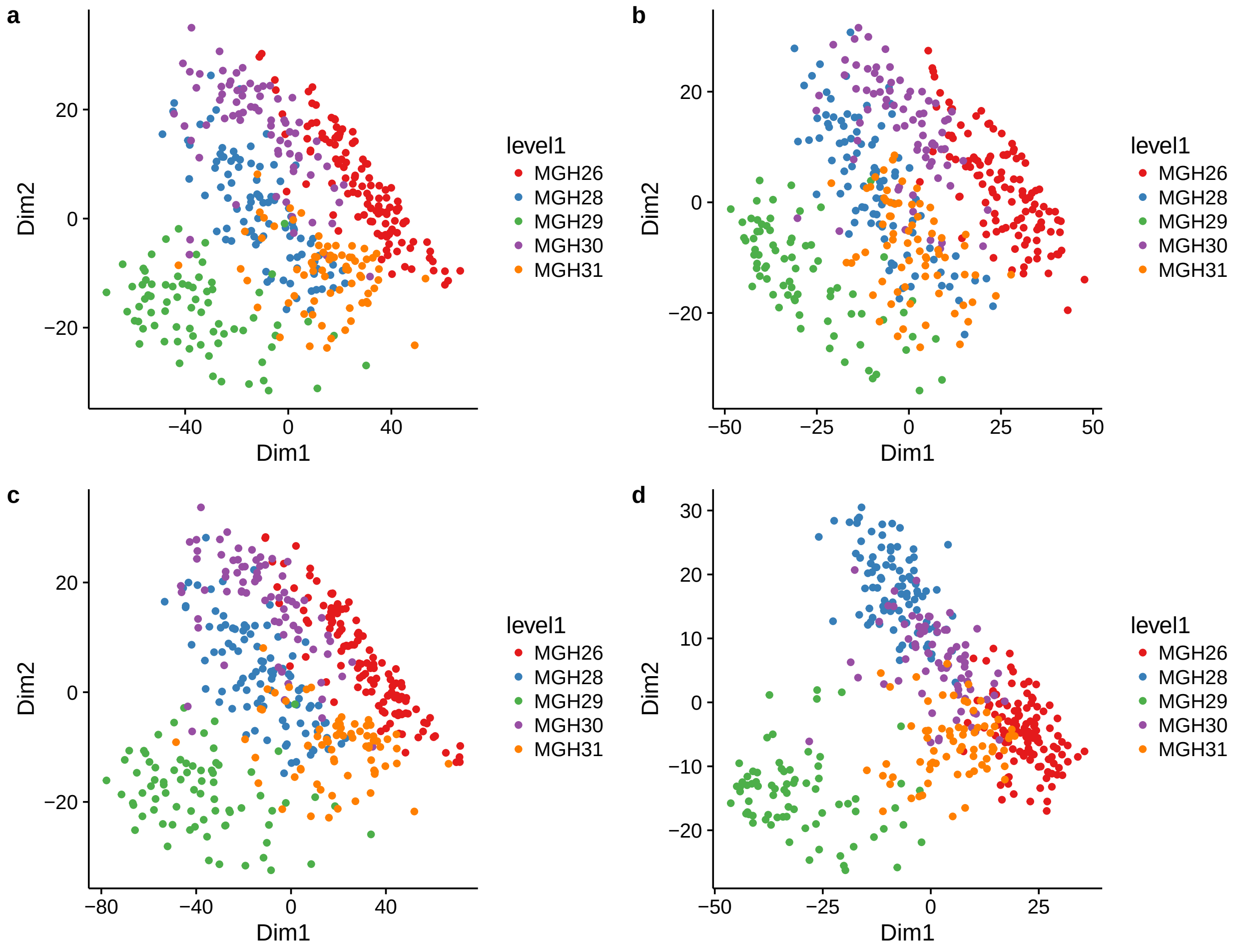


Figure S8: *Principal component analysis for Glioblastoma dataset.* **(a)** No normalization; **(b)** TC normalization; **(c)** TMM normalization; **(d)** FQ normalization.

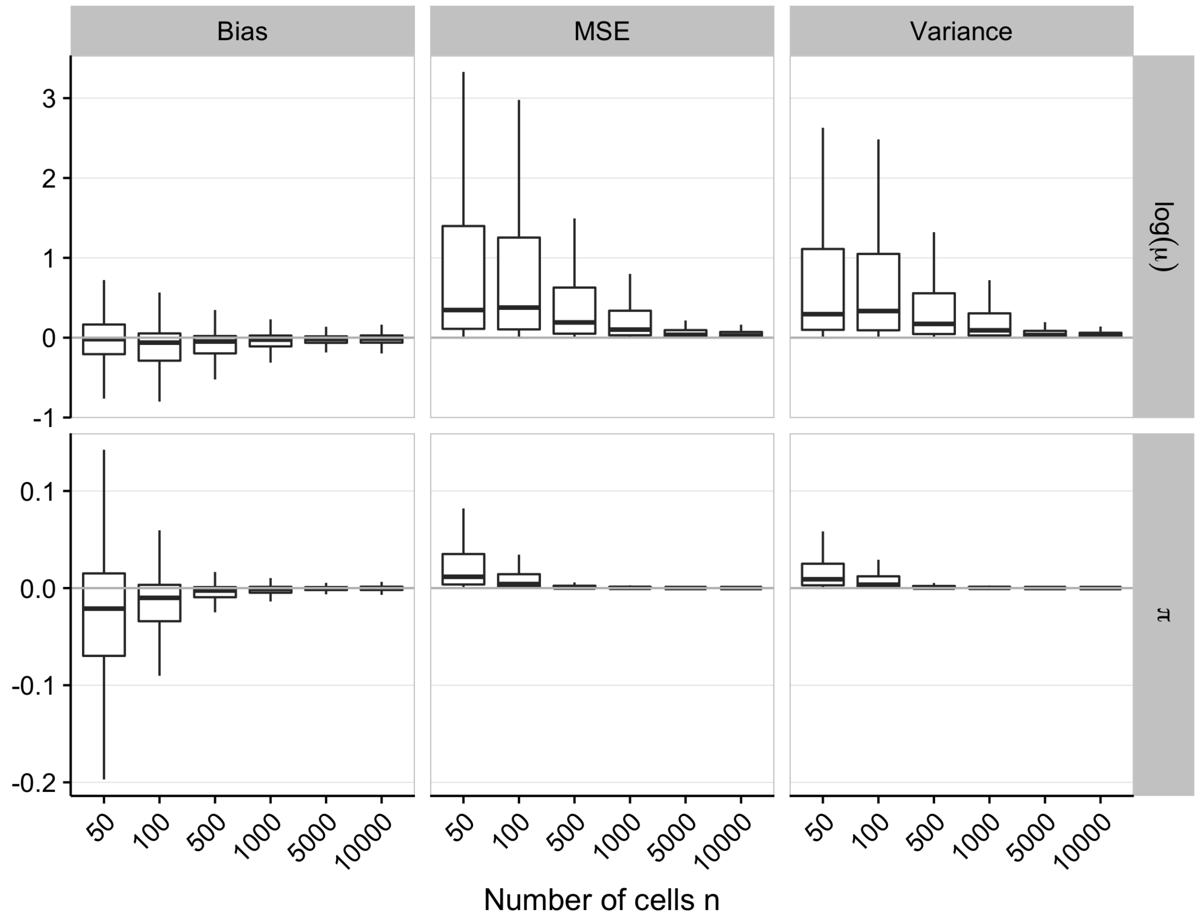


Figure S9: *Bias, MSE, and variance for ZINB-WaVE estimation procedure, ZINB-WaVE simulation model.* Boxplots of bias, MSE, and variance for $\ln(\mu)$ and π as a function of the number of cells n. For each gene and cell, bias, MSE, and variance were averaged over $B = 10$ datasets simulated from our ZINB-WaVE model, based on the S1/CA1 dataset and with $n \in \{50, 100, 500, 1,000, 5,000, 10,000\}$ cells, $J = 1,000$ genes, scaling of one for the ratio of within to between-cluster sums of squares ($b^2 = 1$), and zero fraction of about 80%. The following values were used for both simulating the data and fitting the ZINB-WaVE model to these data: $K = 2$ unknown factors, $X = \mathbf{1}_n$, cell-level intercept ($V = \mathbf{1}_J$), and genewise dispersion.

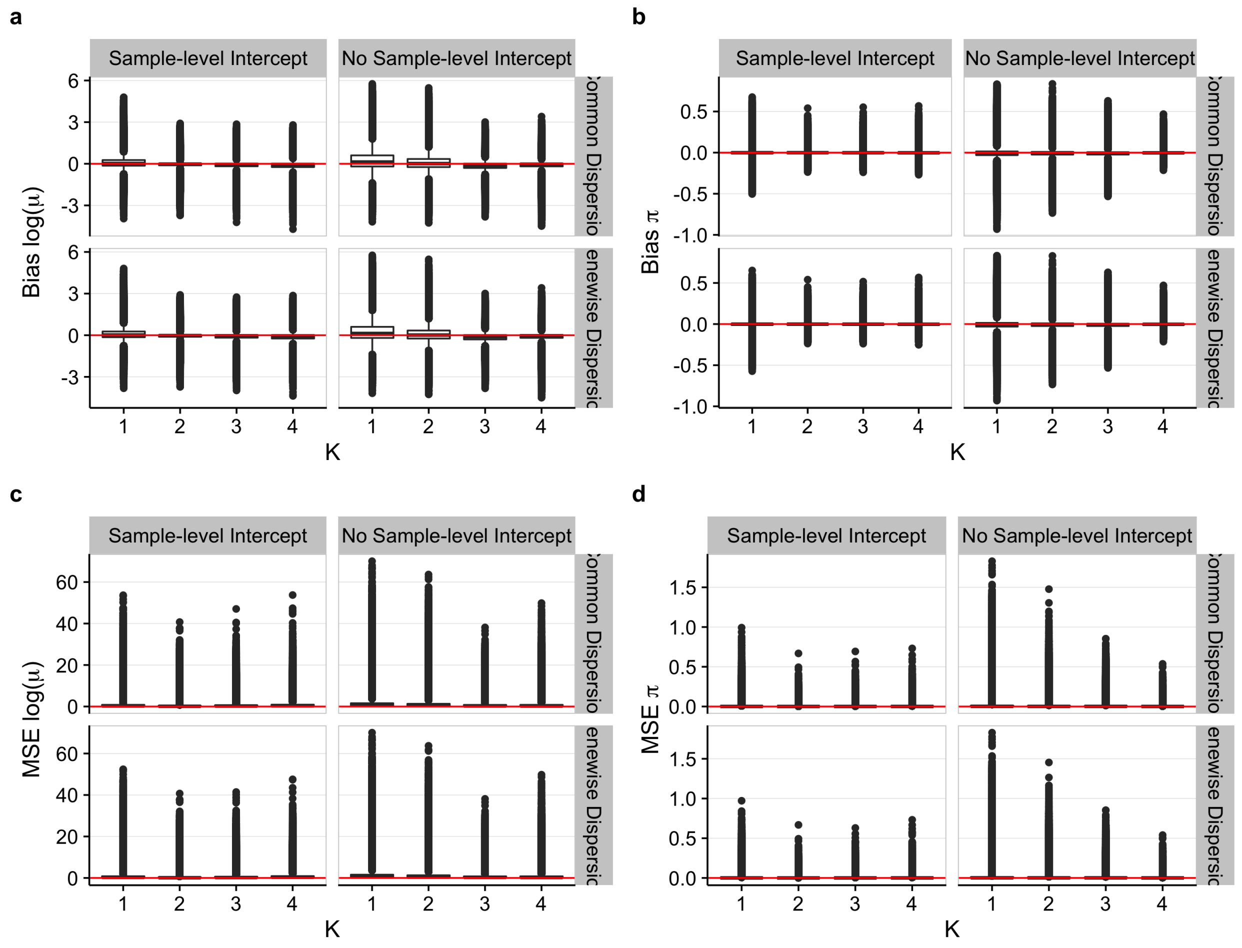


Figure S10: *Bias and MSE for ZINB-WaVE estimation procedure, ZINB-WaVE simulation model.* Same as Figure 5, but with outliers plotted individually (i.e., observations beyond the whiskers).

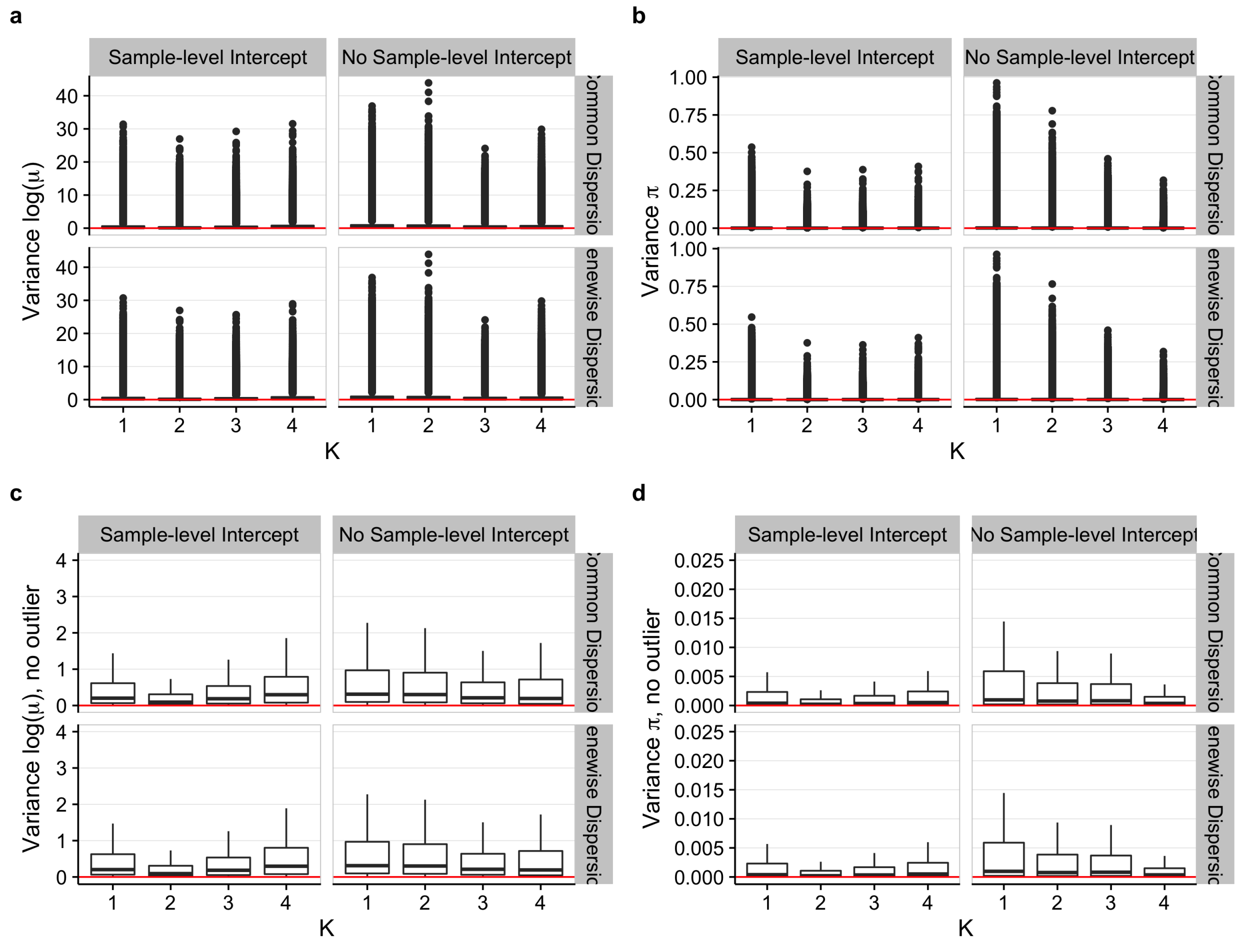
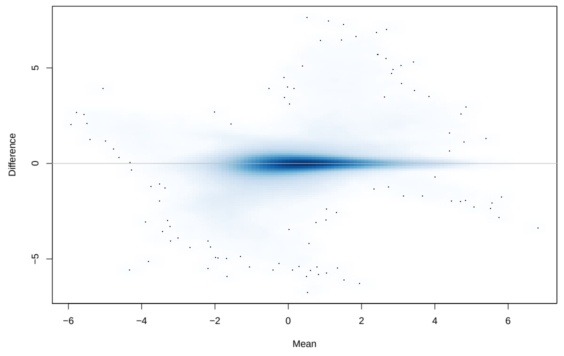
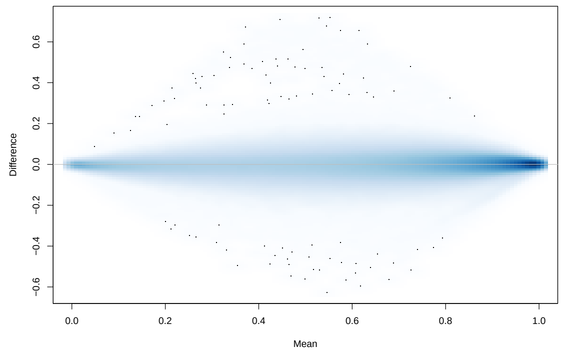


Figure S11: *Variance for ZINB-WaVE estimation procedure, ZINB-WaVE simulation model.* Panels show boxplots of variance (over $B = 10$ simulated datasets) for estimates of $\ln(\mu)$ **(a, c)** and π **(b, d)**. Outliers plotted in **(a, b)** and omitted in **(c, d)**. Simulation scenario as in Figure 5.



(a) $\ln(\mu)$



(b) π

Figure S12: *Mean-difference plots for ZINB-WaVE estimation procedure, ZINB-WaVE simulation model.* **(a)** Mean-difference plot of estimated vs. true negative binomial mean (log scale), $\ln(\hat{\mu}) - \ln(\mu)$ vs. $(\ln(\mu) + \ln(\hat{\mu}))/2$. **(b)** Mean-difference plot of estimated vs. true zero inflation probability, $\hat{\pi} - \pi$ vs. $(\pi + \hat{\pi})/2$. The estimates are based on one of the $B = 10$ datasets simulated from our ZINB-WaVE model, based on the S1/CA1 dataset and with $n = 1,000$ cells, $J = 1,000$ genes, scaling of one for the ratio of within to between-cluster sums of squares ($b^2 = 1$), and zero fraction of about 80%. The following values were used for both simulating the data and fitting the ZINB-WaVE model to these data: $K = 2$ unknown factors, $X = \mathbf{1}_n$, cell-level intercept ($V = \mathbf{1}_J$), and genewise dispersion (as in Fig. 5, S10, and S11).

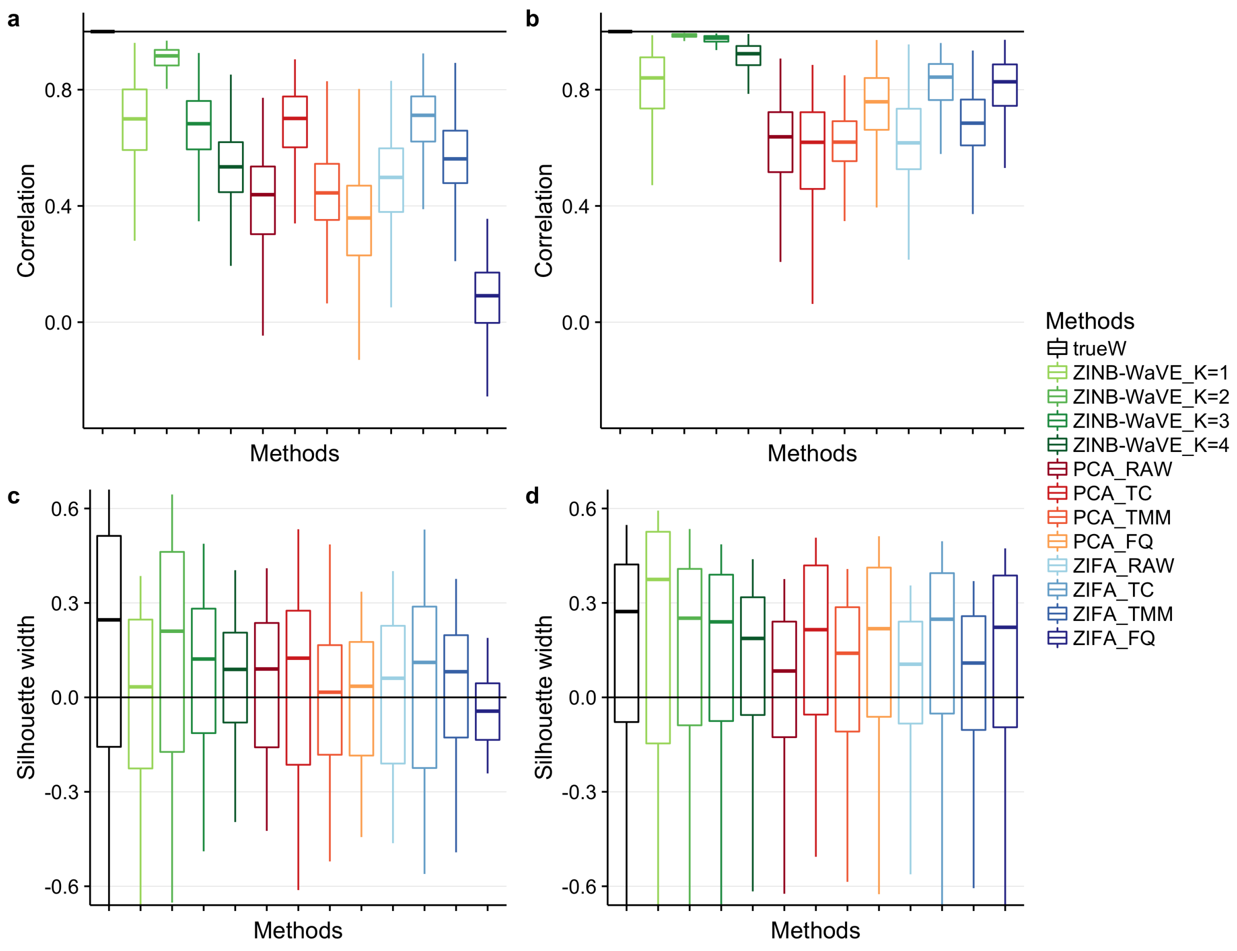


Figure S13: *Between-sample distances and silhouette widths for ZINB-WaVE, PCA, and ZIFA, ZINB-WaVE simulation model.* **(a)** Boxplots of correlations between between-sample distances based on true and estimated low-dimensional representations of the data for datasets simulated from the V1 dataset. **(b)** Same as **(a)** for simulations based on the S1/CA1 dataset. **(c)** Boxplots of silhouette widths for true clusters for datasets simulated from the V1 dataset. **(d)** Same as **(c)** for simulations based on the S1/CA1 dataset. All datasets were simulated from our ZINB-WaVE model with $n = 10,000$ cells, $J = 1,000$ genes, "harder" clustering ($b^2 = 5$), $K = 2$ unknown factors, zero fraction of about 80%, $X = \mathbf{1}_n$, cell-level intercept ($V = \mathbf{1}_J$), and genewise dispersion. Each boxplot is based on n values corresponding to each of the n samples and defined as averages of correlations **(a, b)** or silhouette widths **(c, d)** over $B = 10$ simulations. Between-sample distance matrices and silhouette widths were based on W for ZINB-WaVE, the first two principal components for PCA, and the first two latent variables for ZIFA. ZINB-WaVE was applied with $X = \mathbf{1}_n$, $V = \mathbf{1}_J$, genewise dispersion, and $K \in \{1, 2, 3, 4\}$. For PCA and ZIFA, different normalization methods were used. Colors correspond to the different methods. See Figure 6 **(a-d)** for the same scenario but with $n = 1,000$ cells and Supplementary Figure S14 for additional scenarios.

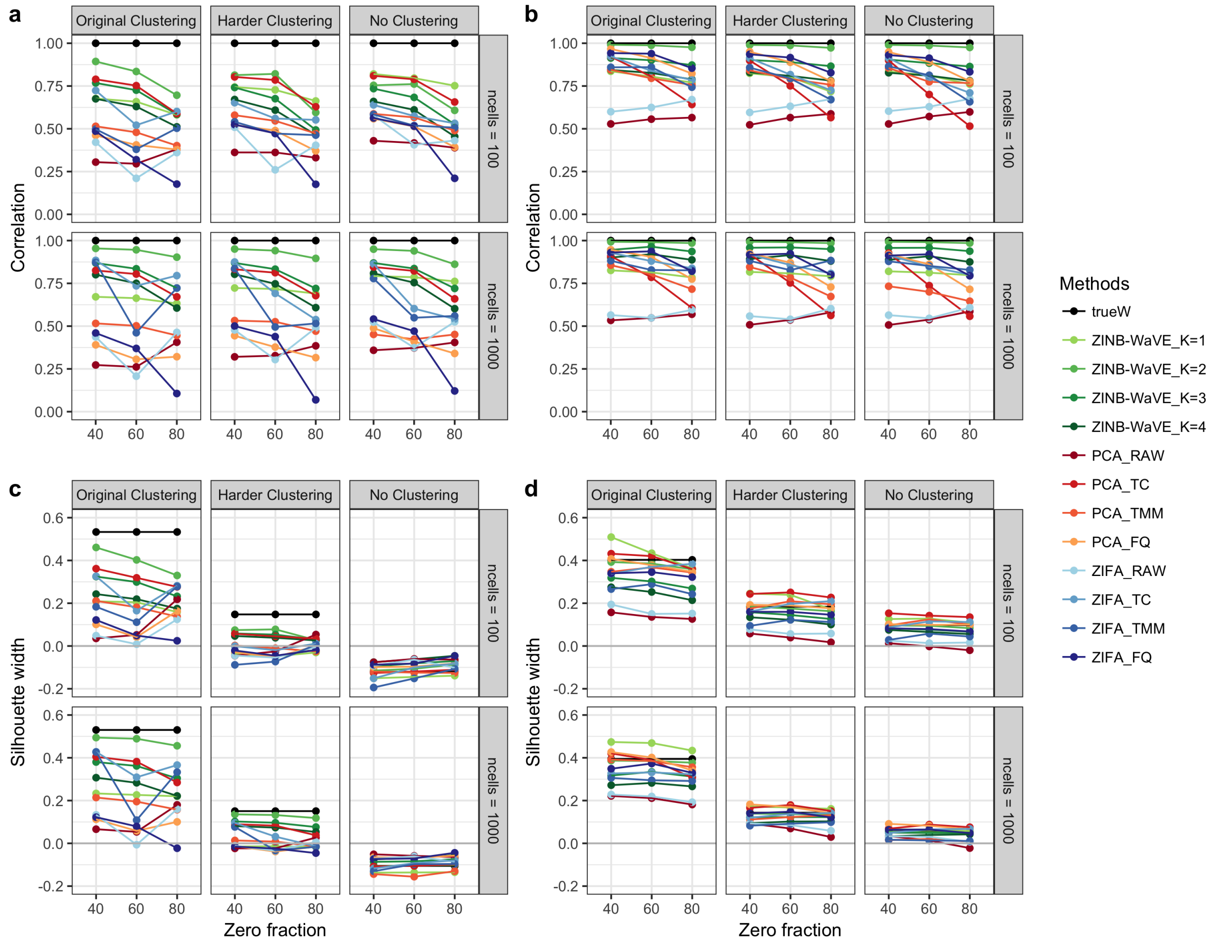


Figure S14: *Between-sample distances and silhouette widths for ZINB-WaVE, PCA, and ZIFA, ZINB-WaVE simulation model.* **(a)** Correlation between between-sample distances based on true and estimated low-dimensional representations of the data for datasets simulated from the V1 dataset. **(b)** Same as **(a)** for simulations based on the S1/CA1 dataset. **(c)** Silhouette width for true clusters for datasets simulated from the V1 dataset. **(d)** Same as **(c)** for simulations based on the S1/CA1 dataset. As in Figure 6, all datasets were simulated from our ZINB-WaVE model with $J = 1,000$ genes, $K = 2$ unknown factors, $X = \mathbf{1}_n$, cell-level intercept ($V = \mathbf{1}_J$), and genewise dispersion. Each point corresponds to a simulation scenario (zero fraction, clustering strength, sample size); correlations between true and estimated between-sample distances and silhouette widths are averaged over $B = 10$ simulated datasets and n cells. Column panels show three different clustering scenarios, where the scaling of the ratio of within to between-cluster sums of squares b^2 corresponds to the original clustering ($b^2 = 1$), a harder clustering ($b^2 = 5$), and no clustering ($b^2 = 10$). Row panels show different number of cells ($n \in \{100, 1,000, 10,000\}$). Between-sample distance matrices and silhouette widths were based on W for ZINB-WaVE, the first two principal components for PCA, and the first two latent variables for ZIFA. ZINB-WaVE was applied with $X = \mathbf{1}_n$, $V = \mathbf{1}_J$, genewise dispersion, and $K \in \{1, 2, 3, 4\}$. For PCA and ZIFA, different normalization methods were used. Colors correspond to the different methods.

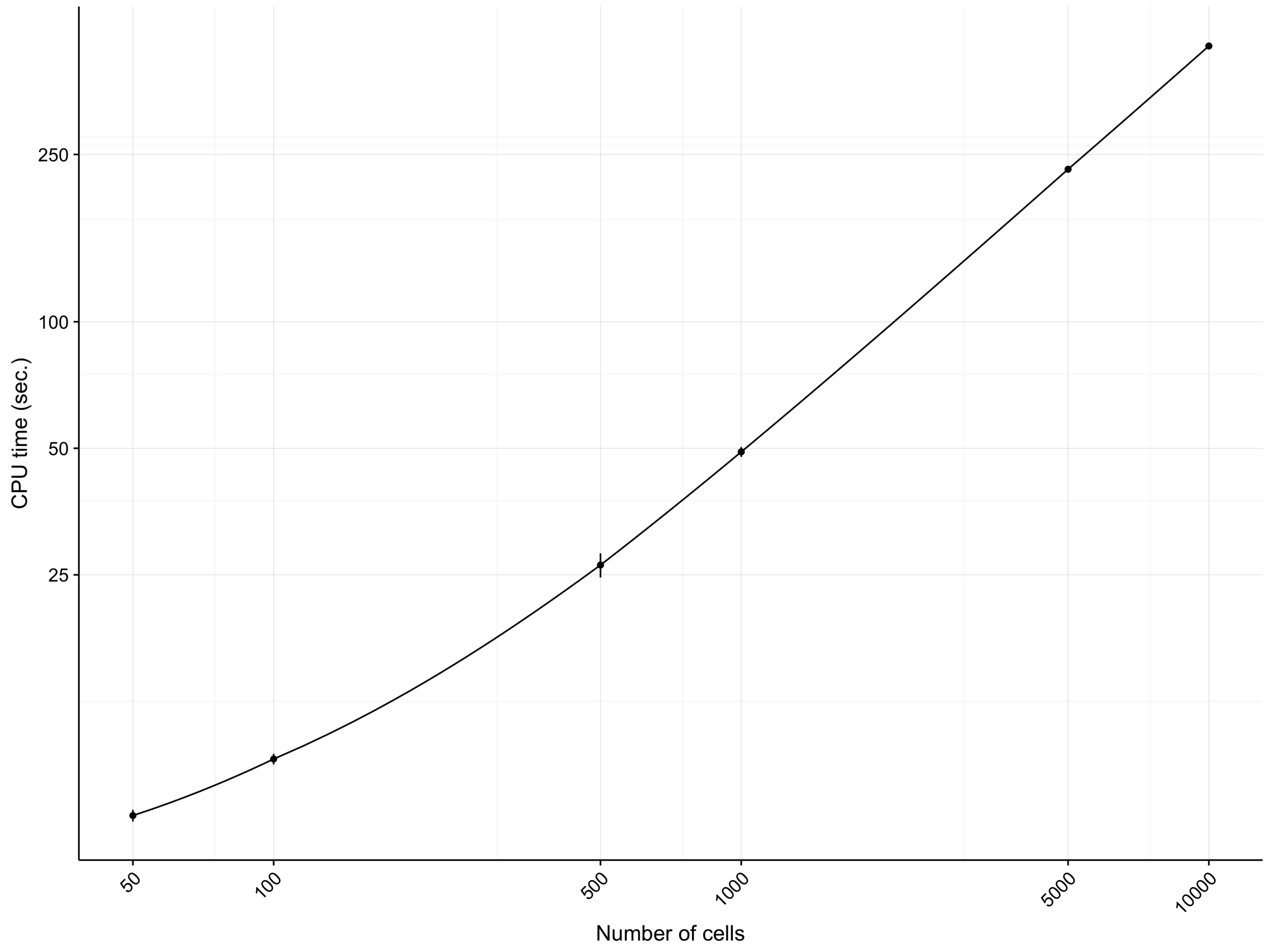


Figure S15: *CPU time vs. sample size for ZINB-WaVE estimation procedure.* Log-log scatterplot of mean CPU time (in seconds) vs. sample size n, for $B = 10$ datasets simulated from the Lun & Marioni [31] model with $n \in \{50, 100, 500, 1,000, 5,000, 10,000\}$ cells, $J = 1,000$ genes, zero fraction of about 60%. The following values were used to fit the ZINB-WaVE model to these data: $K = 2$ unknown factors, $X = \mathbf{1}_n$, cell-level intercept ($V = \mathbf{1}_J$), and common dispersion. CPU time were averaged over $B = 10$ simulated datasets and standard deviations are indicated by the vertical bars. Computations were done with 7 cores on a iMac with eight 4 GHz Intel Core i7 CPUs and 32 GB of RAM.