Gene detection models outperform gene expression for large-scale scRNA-seq analysis

Ruoxin Li[1,2], Gerald Quon[1,2,3,*]
[1]Graduate Group in Biostatistics, [2]Genome Center, [3]Department of Molecular and Cellular Biology, University of California, Davis, Davis, CA

*To whom correspondence should be addressed: gquon@ucdavis.edu

Abstract

The number of cells to be sequenced is one of the most important considerations in single cell RNA experiment design, and we find that experiments designed to maximize the number of sequenced cells tends to also increase the measurement noise within individual cells. This noise is detrimental to the performance of downstream scRNA-seq analysis tools, but can be mitigated by ignoring precise expression measurements and focusing on the detection of a gene. This observation that throwing away data can increase signal is observed across multiple capture technologies, can be addressed using as little as one line of code, and results in execution times that are up to three orders of magnitude faster, therefore making this observation important as newer capture technologies continue to drive the number of sequenced cells higher.

Main text

Single cell RNA sequencing (scRNA-seq) is rapidly being adopted due to the diverse problems in gene regulation that can now be more easily addressed. These include the identification of novel cell types, trajectory inference for analysis of continuous processes such as differentiation, and analysis of transcriptional heterogeneity within populations of cells[1]. The power of scRNA-seq analysis comes from the number of cells that are sequenced, as higher cell counts enable identification and characterization of transcriptional heterogeneity with more certainty. As a result, newer technologies keep pushing the boundaries of throughput capabilities: those based on 3' tagging and unique molecular identifiers (UMI) have now been used to generate datasets totaling over one million cells[2].

While capture technologies are widely acknowledged to produce scRNA-seq data with systematically different characteristics[3], analysis tools for scRNA-seq are largely agnostic to the underlying capture technology. We have observed that globally, more recent studies tend to sequence more cells, but at the expense of the gene detection rate (**Fig. 1a, Supplementary Table 1**) and per-cell library size. Gene detection rate (GDR) is the per-cell average fraction of genes for which at least one read (or UMI) has been mapped to it, and per-cell library size is the average total number of reads (or UMI) that map to a cell. Clearly, when the GDR or per-cell library size fall too low, downstream analysis tools that explicitly model the counts of molecules mapping to each gene (relative expression) will be confounded by the noisy measurements and perform poorly at their respective tasks.

We hypothesized that we can mitigate the noise associated with low GDR by designing analysis tools that only model the gene detection pattern. That is, we propose to transform scRNA-seq data such that we only measure whether at least one molecule has mapped to a gene in a cell, instead of counting how many molecules mapped to a gene. Here we focus on the effects of such a transformation specifically on the task of dimensionality

reduction of scRNA-seq data, as it is a nearly ubiquitous first step in data analysis[4,5,6] and many analysis tools have been developed to address it[7,8,9,10].

To test our hypothesis, we designed Binary Factor Analysis (BFA), a scRNA-seq dimensionality reduction tool that only models gene detection, and benchmarked it against three other approaches that model gene counts: PCA, ZINB-WaVE[8] and scImpute[11]. We benchmark these methods by using their low dimensional embeddings to predict cell type labels in a supervised setting. We identified nine datasets for which cell types were established experimentally without any computational inference (**Supplementary Table 2**), in contrast to other published datasets in which computational techniques were used in the pipeline to label cells.

Of the nine datasets we tested, BFA was a top performer in six of them (Group I) and performed poorly in the remaining three (Group II) (**Fig. 1b**). The difference in performance of BFA and ZINB-WaVE is particularly striking because the ZINB-WaVE model has two components: one that models gene detection, and the other that models gene counts. The model structure and parameter learning algorithm of BFA is designed to match the gene detection component of ZINB-WaVE as closely as possible, making the difference in their performance primarily due to whether gene counts (ZINB-WaVE) or gene detection patterns (BFA) are modeled.

BFA outperforms ZINB-WaVE when the GDR is low, which occurs when the number of cells sequenced is large (**Fig. 1c**). We found that ZINB-WaVE and other methods are less robust in this regime because lower GDR is correlated with increased dispersion (noise) of gene expression (**Fig. 1d**), therefore forcing gene count models and their low dimensional embeddings to explain more outliers in the data. Some gene counting models even share variance parameters across genes[8,12], making them even more susceptible to noise generated from low GDR. The gene detection pattern is more robust to outliers in the counting data because genes that are expressed at low levels are treated equally as genes that are expressed highly. Notably, BFA performs much more poorly when the GDR reaches close to 100%, since in this situation every gene is detected in every cell, so there is no variance across cells for BFA's embedding spaces to capture.

Modeling gene detection makes scRNA-seq analysis robust to other forms of noise besides GDR, such as inclusion of cells of lower quality. **Figure 2a** illustrates the 2D embedding of stimulated dendritic cells[13], while **Supplementary Figures 1-3** illustrate other datasets included in this study. What is exceptional about the dendritic cell dataset is that the raw data has a large number of cells that map many reads to the mitochondrial genes (**Supplementary Fig. 4**). Many reads mapping to mitochondrial genes is an indication of cell death[14], and these cells are typically discarded for downstream analysis, as they were for **Figures 1b** and **2a**. When we include those cells back in the analysis, we found that mitochondrial content drives the embedding of counting based methods much more than BFA, leading to a decrease in ability to group cells of the same type (**Supplementary Fig. 5**). BFA is more robust to this technical artifact again because high counts of specific genes are treated equally with low counts of other genes.

We hypothesized that the embedding spaces learned from gene detection patterns capture cell type-specific markers more readily than the embedding spaces learned from gene counts. To validate our hypothesis, we focused on two datasets, hematopoietic stem cells and dendritic cells, where the marker genes for those cell surface markers are well-studied and we could identify 23 and 60 markers respectively from the literature

(**Supplementary Tables 3, 4**). **Figure 2b** demonstrates that for both datasets, the embeddings of BFA are driven by cell type markers more than the other tested methods, despite the fact that the cell type markers are not used when calculating the embeddings. For BFA, genes that are expressed to some extent in all cells are ignored because they do not vary in their gene detection pattern across cells (the gene is detected in all cells), so intuitively BFA tends to select genes in its embeddings that must be turned completely off in at least some subset of cells, such as cell type markers. Because ZINB-WaVE has a gene detection and gene count component, we compared the performance of each component individually with respect to cell type marker identification. In ZINB-WaVE's model, the gene detection pattern is statistically inferred to account for noise in the gene counts, as opposed to BFA, which treats undetected genes as true observations. **Figure 2b** illustrates that BFA outperforms the gene detection pattern component of ZINB-WaVE, suggesting it is difficult to infer a more meaningful gene detection pattern.

With newer datasets occasionally exceeding one million cells, computational efficiency of scRNA-seq analyses becomes challenging as ideally these tools can be run on local machines. We found that ZINB-WaVE requires a median of 9-fold more execution time (**Fig. 2c**), and in the largest experiment with 500k cells, ZINB-WaVE did not finish running within two weeks. The difference in execution time between BFA and ZINB-WaVE is due primarily to the additional burden of modeling gene counts because the BFA model structure and parameter learning algorithm was designed to match the gene detection pattern component of ZINB-WaVE as closely as possible.

Remarkably, modeling gene detection patterns instead of gene counts can be performed with as little as one line of code in R. In our study thus far, we implemented BFA using a model appropriate for analyzing gene detection patterns, and show it is faster and often more accurate than a corresponding model that also uses gene count information. However, PCA is still used most often because of both its speed and its implementation in frequently-cited pipelines such as Seurat[15]. We have found that simply transforming the gene counts into gene detection patterns as a preprocessing step before use of PCA provides immediate benefits over standard PCA with respect to cell type identification (**Supplementary Fig. 6**) and only requires users to execute one line of R code to perform the transformation. While this so-called Binary PCA approach is not as accurate as BFA, it requires far less execution time than BFA (**Supplementary Fig. 7**) and it does not require installation of new software packages for users. For instance, for 100,000 cells, Binary PCA only requires approximately two minutes of execution time, compared to more than a day for ZINB-WaVE.

Our contribution has several important implications for the field. First, we demonstrate that modeling the gene detection pattern is a strategy to mitigate the high dispersion associated with high cell count data, while maintaining our ability to distinguish cell types. This strategy is not tied to a specific model structure; the performance improvements in BFA compared to ZINB-WaVE, and Binary PCA compared to PCA, demonstrate our results hold across model structures and loss functions. This modification makes sense only when the GDR is below a threshold of 0.9 in our experiments. Second, the success of modeling gene detection patterns shapes our understanding of "dropout". In previous scRNA-seq studies[16,17,11] the gene detection pattern is understood as a mixture of genes truly turned off and genes not detected due to technical artifacts[16]. The underlying assumption of the mixture model is that a significant proportion of undetected genes is due to technical factors, and they therefore try to distinguish biological versus technical occurrences of undetected genes. BFA's assumption that all zeroes are biological, and its

corresponding superior performance when cell counts are high, suggests enough zeroes are true signal that mixture modeling is unhelpful. We therefore propose modeling scRNA-seq datasets by their gene detection pattern as a novel approach to address challenges in scRNA-seq datasets with higher cell counts.

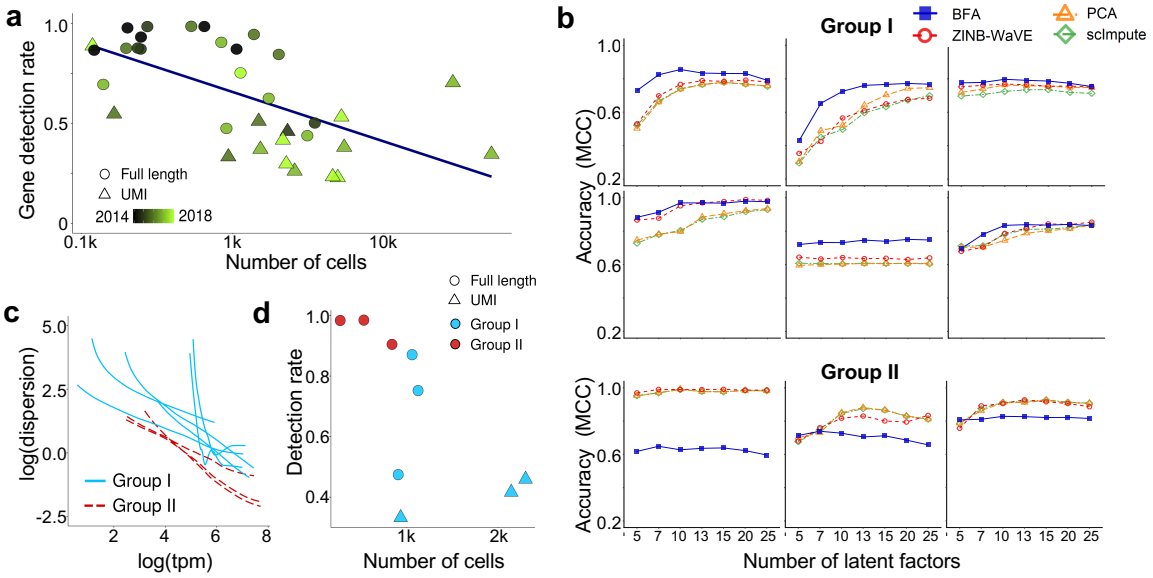

Figure 1: Binary Factor Analysis (BFA) outperforms other approaches when cell counts are high and transcriptional measurements are noisy. (**a**) Gene detection rate as a function of the number of cells sequenced in the study, across 36 diverse datasets (see **Supplementary Table 1**). Datasets are colored by date of publication. (**b**) Cross-validation performance of cell type predictors trained on scRNA-seq data that has been reduced in dimensionality, as a function of the number of dimensions specified. Datasets from left to right, top to bottom: dendritic, MGE, HSCs, mouse intestinal, CD4-CTL/CD4+ memory T, pancreatic, H7-ESC, mESCs and LPS (see **Supplementary Table 2**). Group I datasets are those in which BFA is a top performer, Group II datasets are those where BFA performs poorly relative to other methods. (**c**) Mean-dispersion trends estimated for each of the datasets from (a). Datasets are colored by their corresponding group. (**d**) The gene detection rate as a function of the number of cells from each dataset in (b).

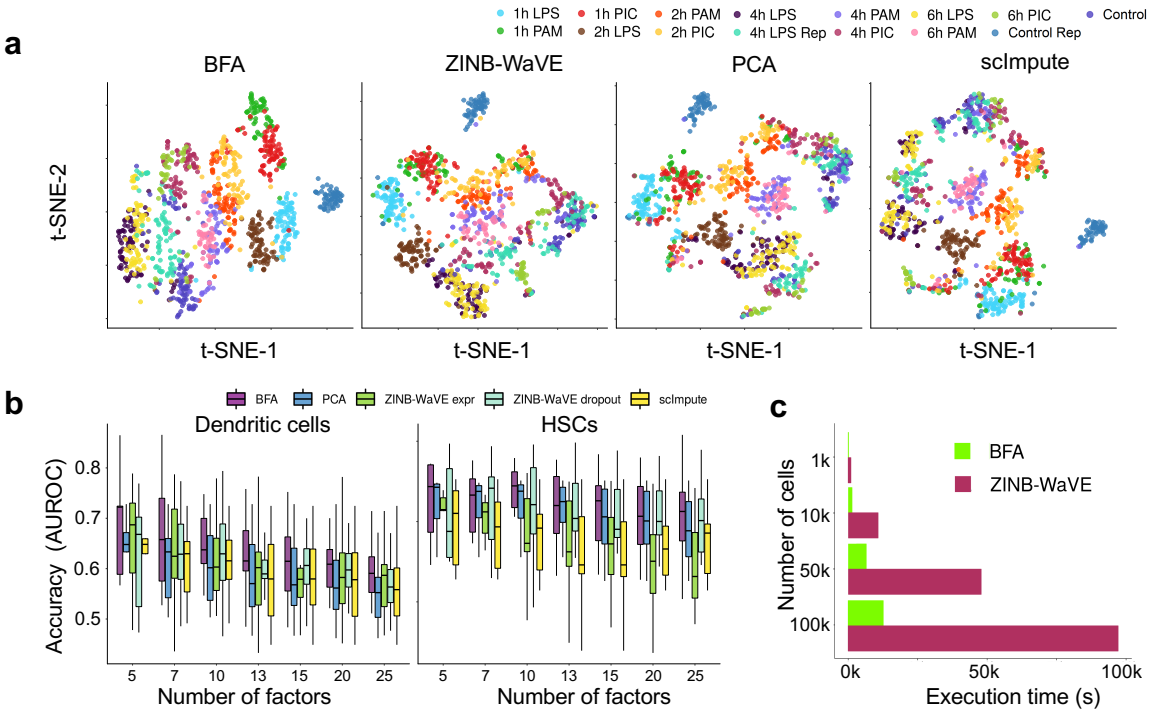

Figure 2: BFA embeddings recover cell type markers more efficiently. (**a**) 2D tSNE visualization of 10-dimensional embeddings generated by the four methods on the dendritic dataset. Cells are colored according to their corresponding cell types and states. (**b**) Comparison of the methods in terms of how enriched the inferred dimensions are with respect to the known cell type markers, as measured by AUROC (area under the curve). ZINB-WaVE is represented twice, once for the latent dimensions inferred by their gene detection pattern (ZINB-WaVE$_{dropout}$), and the other are the latent dimensions inferred from the gene counts (ZINB-WaVE$_{expr}$). (**c**) Execution time of BFA versus ZINB-WaVE on differently-sized subsets of cells subsampled from the 1.3 million scRNA-seq dataset generated from 10x Genomics.

References

1. Hwang, B., Lee, J. H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.* **50,** 96 (2018).
2. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8,** 14049 (2017).
3. Ziegenhain, C. *et al.* Comparative Analysis of Single-Cell RNA Sequencing Methods. *Mol. Cell* **65,** 631–643.e4 (2017).
4. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.* **32,** 381–386 (2014).
5. Haghverdi, L., Buettner, F. & Theis, F. J. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinforma. Oxf. Engl.* **31,** 2989–2998 (2015).
6. Giecold, G., Marco, E., Garcia, S. P., Trippa, L. & Yuan, G.-C. Robust lineage reconstruction from high-dimensional single-cell data. *Nucleic Acids Res.* gkw452 (2016). doi:10.1093/nar/gkw452
7. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **16,** 241 (2015).
8. A general and flexible method for signal extraction from single-cell RNA-seq data | Nature Communications. Available at: https://www.nature.com/articles/s41467-017-02554-5#Abs1. (Accessed: 2nd October 2018)
9. 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. Methods* **14,** 414–416 (2017).
10. Ding, J., Condon, A. & Shah, S. P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat. Commun.* **9,** 2002 (2018).

11. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat. Commun.* **9,** 997 (2018).

12. Finak, G. *et al.* MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.* **16,** 278 (2015).

13. Shalek, A. K. *et al.* Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* **510,** 363–369 (2014).

14. Tait, S. W. G. & Green, D. R. Mitochondrial Regulation of Cell Death. *Cold Spring Harb. Perspect. Biol.* **5,** (2013).

15. Spatial reconstruction of single-cell gene expression data | Nature Biotechnology. Available at: https://www.nature.com/articles/nbt.3192. (Accessed: 23rd October 2018)

16. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nat. Methods* **11,** 740–742 (2014).

17. Kim, J. K. *et al.* Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nat. Commun.* **6,** 8687 (2015).

18. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9,** 2579–2605 (2008).

Online Methods

BFA model

Our Binary Factor Analysis (BFA) model is adapted from a generalized linear model framework, and is therefore capable of adjusting for batch effects and other nuisance cell level covariates. The workflow is shown in **Supplementary Figure 8**. Let B_{ij} refer to the gene detection pattern observed for cell i and gene j, where $B_{ij} = 1$ when the gene is expressed and at least one read (or UMI) maps to gene j in cell i, otherwise $B_{ij} = 0$. Let N represent the number of cells in the dataset, G the number of genes measured, and K the number of latent dimensions to infer. BFA is defined by the following model:

$$B_{ij} \sim \text{Bernoulli}(\mu_{ij})$$
$$\text{logit}(\mu_{ij}) = X_i^T \beta_j + Z_i^T A_j + O_j + U_i$$

Here, X is an N by C cell-level covariate matrix that corrects for nuisance factors such as batch effect. X_i refers to the ith row of X. Let β be the C by G coefficient matrix, and β_j the jth column of β. Z is an N by K low dimensional embedding matrix, A is a K by G compressed feature matrix. Z_i and A_j are the ith row and j^{th} column of Z and A, respectively. Moreover, O_j and U_i stands for the jth gene level intercept and i^{th} cell level intercept, respectively. O is therefore a vector of length G, and U is a vector of length N.

We train the BFA model by optimizing the following likelihood function:

$$f(A, Z, \beta, O, U) = \sum_{i,j} \ln P(B_{ij}; A, Z, \beta, O, U) - \epsilon_1 ||A||_2^2 - \epsilon_2 ||Z||_2^2 - \epsilon_3 ||\beta||_2^2$$

Here ϵ_1, ϵ_2 and ϵ_3 are tunable parameters for controlling regularization of the model parameters, where $\epsilon_1 = \frac{\epsilon_0}{N}, \epsilon_2 = \frac{\epsilon_0}{G}$, and $\epsilon_3 = \frac{\epsilon_0}{G}$, here $\epsilon_0 = \max\{N, G\}$. The optimization is carried out using conjugate gradient descent. After completing optimization, we orthogonalize Z and A as was done in ZINB-WaVE[8].

Selection of representative datasets to measure gene detection. We obtained a total of 36 scRNA-seq datasets from which we calculated gene detection rates as a function of the number of cells in each dataset (Figure 1a). We obtained these datasets from two sources, the conquer database and the Gene Expression Omnibus (GEO). For GEO, we used the search term "((‘single cell rna-seq’ OR ‘single cell transcriptomic’ OR ‘10X’ OR ‘single cell transcriptome’) AND Expression profiling by high throughput sequencing[DataSet Type]) AND (Homo sapiens[Organism] OR Mus musculus[Organism])", sorted all datasets by size, then selected a similar number of datasets from both the top and bottom of the list.

Evaluation of dimensionality reduction methods

We evaluated each dimensionality reduction method based on how well their low dimensional embeddings could discriminate different cell types. For each dataset tested, we first performed dimensionality reduction on the entire dataset for a fixed number of dimensions, to obtain an embedding matrix representing each cell in K dimensions. We then performed 5-fold cross validation by repeatedly splitting the embedding dataset into training sets and testing sets, training a non-regularized multi-level logistic classifier on the training dataset using the *a priori* known cell type labels, then used the model to predict

cell type labels for the test dataset. For every prediction, using the known cell type labels, we computed a confusion matrix and the corresponding Matthews' correlation coefficient as a measure of classification accuracy:

$$\mathrm{MCC} = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)}\sqrt{(TP+FN)}\sqrt{(TN+FP)}\sqrt{(TN+FN)}}$$

Quality control

For each dataset tested, we performed a standardized quality control process. We first remove cells where mitochondrial genes accounted for over 50% of the total read count. Then we filtered out genes that are expressed in fewer than 1% of cells, and removed cells with library size (total read or UMI count) less than one eighth quantile of all cell library sizes.

Normalization

For each method, we also normalized cells to control for differences in library size. For PCA, we normalize the count by $x_{ij} = \log\left(\frac{y_{ij}}{c_i} + 1\right)$, where x_{ij} is the normalized gene count for cell i and gene j, y_{ij} is the original gene count for cell i and gene j, and c_i is library size for cell i. ZINB-WaVE directly accounts for library size by their cell-wise intercept[8]. For scImpute, we use the total number of read counts in the imputed space as their corresponding library size and normalize in the same way as PCA. For BFA, we input library size as a cell covariate to allow it to decide whether to use library size to explain the gene detection pattern.

Gene selection

For all datasets, we subset the genes to the 2,000 most variant genes before dimensionality reduction, a common step[15]. Note that the gene detection rate of each dataset calculated in **Figure 1** is based on these 2,000 most variant genes. To be consistent across all datasets, we calculate the gene-wise variance in the gene count space. For the 68k PBMC dataset, we only selected the top 1,000 most varying genes for computational speed.

Batch effect correction

For both ZINB-WaVE and BFA, we model the batch effect as cell-level covariates and regress them out within the model structure. Since PCA does not offer a framework to regress out nuisance factors, we first regress batch effect as cell-level covariates directly from the normalized values x_{ij} using a linear model. Then we apply PCA on the residual matrix and obtain the corresponding factor scores and factor loading matrix. For Binary PCA and scImpute, we also regress out batch effect from the binary entries and imputed values respectively, then use the residual matrix in the same way as for PCA.

Identification of marker genes

We evaluated the extent to which the inferred dimensions for each method recovers known marker genes (**Fig. 2b)**. For every method, we first obtained the K by G factor loading matrix indicating which genes are contributing to each of the K factors. Then for every loading matrix and given number of factors, we rank the absolute value of each gene in each factor and calculate the AUROC (area under the receiver-operator curve) to measure the extent to which the known marker genes contribute more to a factor than expected by chance.

Note that ZINB-WaVE has two feature matrices corresponding to the gene detection and gene count components, respectively, and therefore appears twice in **Figure 2b**. In ZINB-WaVE, π_{ij} models whether a gene been detected or not, and μ_{ij} models the mean for the read counts under negative binomial distribution.

$$\mathrm{logit}(\pi_{ij}) = (W\alpha_\pi + X\beta_\pi + (V\gamma_\pi)^T + O_\pi)_{ij}$$

$$\log(\mu_{ij}) = \left(W\alpha_\mu + X\beta_\mu + (V\gamma_\mu)^T + O_\mu\right)_{ij}$$

The feature matrix α_π that models the gene detection component (π) is denoted *ZINB-WaVE*$_{\text{dropout}}$ and the feature matrix α_μ that model gene counts (μ) is denoted *ZINB-WaVE*$_{\text{mean}}$.

Visualization

After we obtain the factor score matrix from every method, we use the t-distributed stochastic embedding[18] method to project it on 2 dimensions for visualization as a scatterplot. In **Figure 2a**, the number of factors used as input to tSNE in each visualization is 10.

Timing experiments

In the timing experiment of **Figure 2c**, we randomly subsample 1k, 5k, 10k, 50k, 100k, and 500k cells from the 1.3 Million 10x dataset of Brain Cells from E18 Mice and record the single-core execution time (in seconds) of both ZINB-WaVE and BFA on the same machine. ZINB-WaVE did not complete the experiment with 500k cells, so we only present our comparison up to 100k cells. Due to the nonconvex nature of ZINBWAVE's objective function and different optimization scheme, we cannot strictly match the convergence criterion of ZINB-WaVE to BFA. Therefore, we use the same number of iterations for each method that was used to generate the results in **Figure 1b**.

Acknowledgements

Funding for this work was provided by the Chan Zuckerberg Initiative DAF and Silicon Valley Community Foundation Grant #182633.

Author contributions

R.L. and G.Q. conceptualized the study, analyzed data and wrote the manuscript. R.L. performed all experiments.