Li and Quon *Genome Biology* (2019) 20:193
https://doi.org/10.1186/s13059-019-1806-0

Genome Biology

METHOD Open Access

# scBFA: modeling detection patterns to mitigate technical noise in large-scale single-cell genomics data

Ruoxin Li[1,2] and Gerald Quon[1,2,3*]

## Abstract

Technical variation in feature measurements, such as gene expression and locus accessibility, is a key challenge of large-scale single-cell genomic datasets. We show that this technical variation in both scRNA-seq and scATAC-seq datasets can be mitigated by analyzing feature detection patterns alone and ignoring feature quantification measurements. This result holds when datasets have low detection noise relative to quantification noise. We demonstrate state-of-the-art performance of detection pattern models using our new framework, scBFA, for both cell type identification and trajectory inference. Performance gains can also be realized in one line of R code in existing pipelines.

**Keywords:** scRNA-seq, Dimensionality reduction, scATAC-seq, Technical noise, Gene detection, Gene quantification, Cell type identification, Trajectory inference, Variable gene selection

## Background

Single-cell genomics technologies have become a widely used technique for investigating diverse problems related to gene regulation, including the identification of novel cell types and their regulatory signatures, trajectory inference for the analysis of continuous processes such as differentiation, high-resolution analysis of transcriptional dynamics, and characterization of transcriptional heterogeneity within populations of cells [1]. Of the different modalities that can be profiled, single-cell RNA sequencing (scRNA-seq) is currently the most mature; diverse scRNA-seq technologies are now available to cater towards specific applications. For instance, droplet-based methods, such as Drop-seq, currently have some of the highest throughput capture of cells and are suitable for rare cell type identification and characterization of tissue heterogeneity. The so-called full-length transcript methods are able to measure alternative splicing and sequence individual cells more deeply, with the limitation of typically sequencing fewer cells.

scRNA-seq technologies are still rapidly evolving [2], and one of the most pressing challenges today is to address a large amount of technical noise that can drive approximately 50% of the cell-cell variation in expression measurements [3–5]. Two such expression measurements of interest are gene detection (the identification of the set of all genes truly expressed in a given cell) and gene quantification (the estimation of the relative number of transcripts per gene and cell, also referred to as counts); the fidelity of these measurements for a given technology is termed its sensitivity and accuracy, respectively. Both sensitivity and accuracy vary widely between scRNA-seq technologies [6], which is the result of the small quantities of RNA sequenced per cell, reverse transcriptase inefficiency, and amplification bias [5], among other features of the scRNA-seq protocols.

Independent of technology choice, scRNA-seq experimental design necessitates a cost trade-off between deeper sequencing of individual cells and sequencing more cells overall. We have observed that as the number of cells sequenced increases, the average gene detection rate decreases, as does the average number of molecules sequenced per cell (Additional file 1: Figure S1), due to both choice of technology and cost trade-off. We reasoned that when the number of unique molecules drops

* Correspondence: gquon@ucdavis.edu
[1]Graduate Group in Biostatistics, University of California, Davis, Davis, CA, USA
[2]Genome Center, University of California, Davis, Davis, CA, USA
Full list of author information is available at the end of the article

too low, the signal-noise ratio of the data may be too low to make gene quantification informative [7], and therefore, downstream analyses should be adapted to primarily consider only gene detection patterns.

In this paper, we make the key observation that on scRNA-seq datasets exhibiting high technical noise, dimensionality reduction using only the gene detection measurements is superior to the existing state-of-the-art methods that use both detection and quantification measurements [8, 9]. We show that our new detection-based model, single-cell binary factor analysis (scBFA), leads to better cell type identification and trajectory inference, more accurate recovery of cell type-specific markers, and is much faster to perform compared to several quantification-based methods. Through simulation experiments, we demonstrate that our gene detection model is superior precisely when quantification noise exceeds detection noise, providing a principled explanation for when and why discarding quantification estimates is advantageous. Finally, we demonstrate the superiority of our detection model in the analysis of single-cell chromatin accessibility data, suggesting detection models may improve downstream analysis of other single-cell genomic modalities in high-throughput datasets.

## Results

### scBFA achieves superior performance in cell type identification across diverse benchmarks

We first hypothesized that the performance of scRNA-seq analysis tools that model gene counts (quantification) could be improved by instead modeling only the gene detection patterns when analyzing datasets that have a high degree of technical noise. Our intuition is that it is well established that poorly expressed genes are hard to accurately quantify using single-cell genomics technologies due to technical noise [10, 11]. Extrapolating to an entire dataset, we then reasoned that for datasets in which technical noise leads to low gene detection and noisy quantification, modeling differences in small gene counts is challenging and prone to error, and therefore, focusing only on gene detection would be more robust.

To test our hypothesis, we developed single-cell binary factor analysis (scBFA), a method for dimensionality reduction that only uses gene detection patterns. We compared scBFA against seven other approaches that model gene counts and represent the spectrum of approaches to identifying cell types within scRNA-seq datasets (see the “Methods” section): scVI [12], SAVER [13], sctransform [14], scrna2019 [15], PCA, ZINB-WaVE [8], and scImpute [9]. scBFA is designed as a gene detection-based analog of ZINB-WaVE, and so, comparison of scBFA versus ZINB-WaVE is the most direct comparison of gene detection versus quantification-based approaches. In this study, we focus on the task of dimensionality reduction, as it is a nearly ubiquitous first step both for data visualization and analysis [16–18] and many analysis tools have been developed to address it [8, 9, 12, 19, 20]. Furthermore, previous work has shown that cell type identification and dimensionality reduction are still possible in scRNA-seq experimental designs favoring high cell counts, with low coverage per cell [4, 21–23].

We evaluated the methods using 14 benchmark datasets for which experimentally defined cell type labels were available (Additional file 1: Table S1) by first learning low dimensional embeddings, then using the embeddings to predict cell type labels in a supervised setting. When using highly variable genes (HVGs) as a gene selection criterion during data preprocessing, we found scBFA was the best, or tied for best, in 13 out of 14 benchmarks (Figs. 1 and 2, Additional file 1: Figures S2-S3). This result was robust to the selection of the hyperparameters of scBFA (Additional file 1: Figure S4). Surprisingly, we found that the choice of gene selection had a significant impact on our results. Under an alternative gene selection procedure that biases towards highly expressed genes (HEGs) and yields minimal overlap with HVG (Additional file 1: Table S2), scBFA was a top performer in only 9 of 14 of the benchmarks (Additional file 1: Figure S5).

### Gene selection shapes cell type identification performance by modulating detection rate and dispersion

We hypothesized the stark difference in performance between the HVG and HEG selection criteria was due to the differences in the overall technical noise in the resulting selected gene sets. For both the HVG and HEG versions of each benchmark, we computed two indirect measures of technical noise, the gene detection rate (GDR) and gene-wise dispersion. Existing approaches to directly estimating technical noise require spike-in standards [24, 25], and not all datasets we analyzed had incorporated spike-in standards in their protocol. GDR is the average fraction of genes that are detected as expressed in a given cell. Gene-wise dispersion is intuitively the excess variation in the gene expression observed beyond what is expected based on a Poisson model of sampling noise and is driven by both technical noise and biological factors of interest.

When considering individual benchmarks in isolation, HVG selection leads to a systematically lower GDR and higher gene-wise dispersion compared to HEG selection (Fig. 3). Furthermore, HEG selection consistently leads to higher performance in cell type identification for all methods tested (Fig. 3c), suggesting that HEG selection may be more sensible for cell type identification. This result is intuitive, as the HVG selection procedure identifies genes whose variance is in excess of that predicted by sampling noise, and therefore is likely to be enriched in poorly expressed genes that exhibit significant

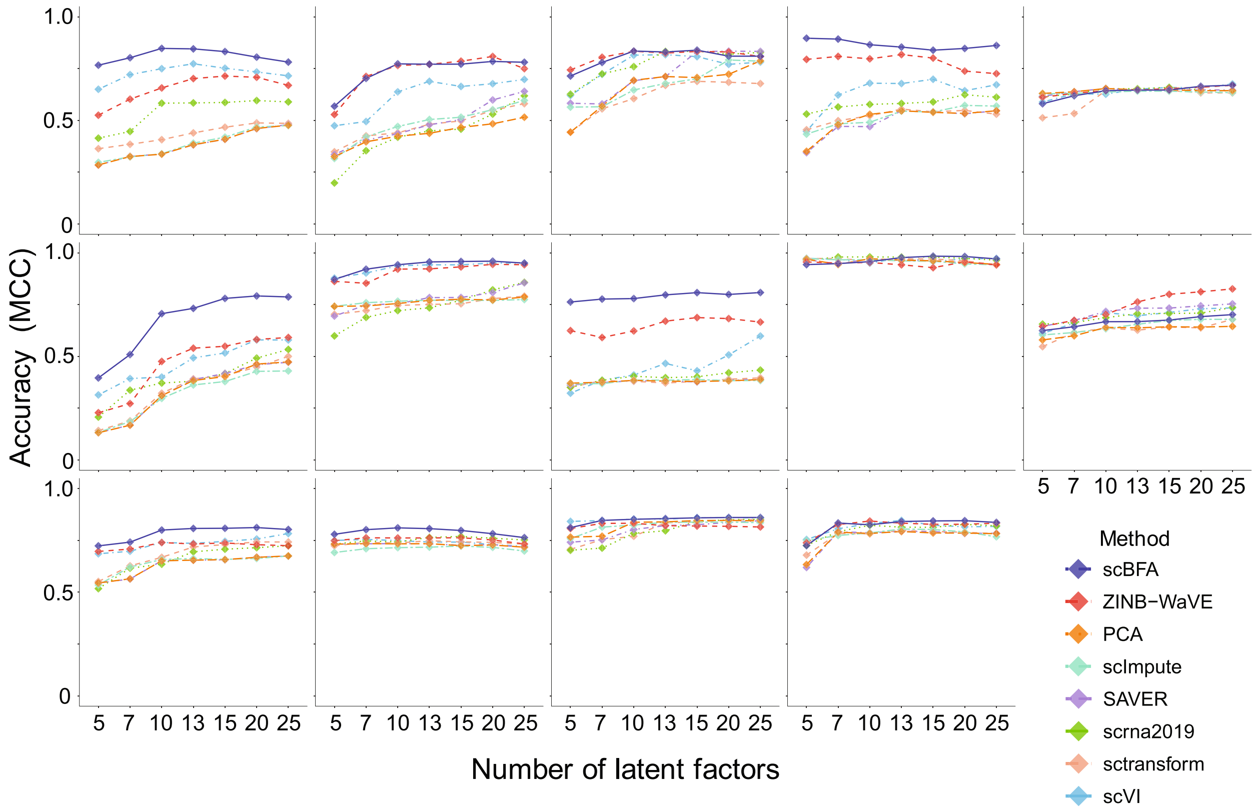

**Fig. 1** Single-cell binary factor analysis (scBFA) outperforms quantification models. Performance is measured via cross-validation of cell type classifiers trained on scRNA-seq benchmark data in the respective embedding spaces of each method, as a function of the number of latent dimensions specified. scBFA is a top performer in 13 out of 14 datasets. Datasets from left to right, top to bottom: Dendritic, Pancreatic, DC, mESCs, HSPCs, MGE, Intestinal, MEM-T, H7-ESC, LSK, Myeloid, HSCs, PBMC, and LPS (see Additional file 1: Table S1 for a detailed description of benchmarks)

dropout noise. For scBFA, specifically, the three benchmarks for which HVG outperformed HEG correspond to the benchmarks for which HEG selection led to the highest GDR and HVG selection led to low GDR (Fig. 3c, d). The poor performance of scBFA combined with HEG selection can therefore be explained by the high GDR.

Across all benchmarks and gene selection criteria, we found that scBFA outperforms count-based methods for benchmarks with low GDR and high gene-wise dispersion (Figs. 1 and 3; Additional file 1: Figure S5, group I benchmarks). This is likely because higher dispersion increases the noise within the gene counts, forcing count-based models and their low-dimensional embeddings to explain more outliers and noise in the data; this is particularly true for count models that share variance parameters across genes [8, 26]. The gene detection pattern is more robust to noise than counts because moderately to highly expressed genes are likely to be equally well detected even in the presence of technical noise. Interestingly, low GDR of a dataset in particular is associated with more sequenced cells regardless of the experimental protocol used (Additional file 1: Figure S1) and is likely a result of investigators trading off sequencing many cells at the cost of sequencing fewer reads per cell. These results together suggest that scBFA is more appropriate for large-scale dataset analysis than quantification-based methods.

Inversely, high GDR is more typical of smaller datasets (Additional file 1: Figure S1) and yields poor performance of scBFA. This is because when the GDR reaches close to 100%, every gene is detected in nearly every cell, so there is a limited variation for scBFA to capture in its embedding space. Consistent with these results, we found that the performance of scBFA decreases after imputation (SAVER-scBFA, scImpute-scBFA) relative to before imputation (scBFA) (Additional file 1: Figure S6), in part because imputation increases GDR by imputing false-positive zero expression. On average, SAVER increased the GDR by 9.6% and scImpute increases GDR by 232.6%.

### Balance of detection and quantification noise determines the relative performance of detection and count models

We next sought to identify precisely which types of technical noise were responsible for the relative performance of scBFA versus the gene count models.

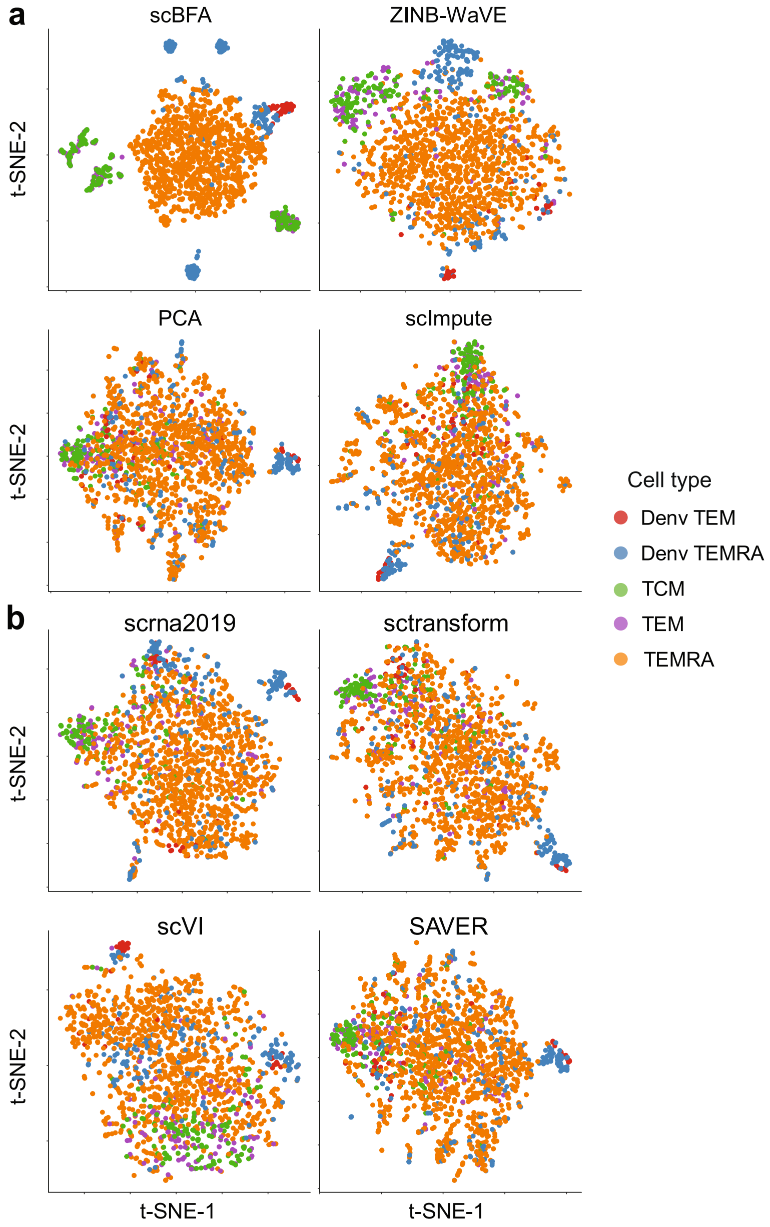

**Fig. 2** scBFA improves visualization of cell identity in the MEM-T benchmark of Patil et al. 2D t-SNE visualization of 10-dimensional embeddings generated by the eight methods on the MEM-T benchmark. Cells are colored according to their corresponding cell types and states.

Previous studies found that sensitivity and accuracy (gene detection and quantification) can be affected differently by sequencing depth and other features of the protocols [7, 27]. We hypothesized that differences in detection and quantification noise might explain the performance difference between scBFA and quantification-based methods. Because technical noise is difficult to estimate in real datasets without the spike-in standards, we instead generated thousands of simulated scRNA-seq datasets that systematically vary in the relative amount of noise in gene detection and gene counts (quantification).

Our simulation framework extends the ZINB-WaVE statistical model [8] to include the parameters that separately influence the noise added to either the gene detection pattern $(\sigma_\pi^2)$ or the gene counts $(\sigma_\mu^2)$ in the simulated datasets. We also tuned the global level of gene dispersion that drives variation in gene counts via the parameter $r$, which adds noise specifically to the UMI counts in the dataset and is a key parameter of many dimensionality reduction models [8, 12, 28]. Finally, we also tuned the global level of gene dropout observed in the dataset via the parameter $\delta$, to

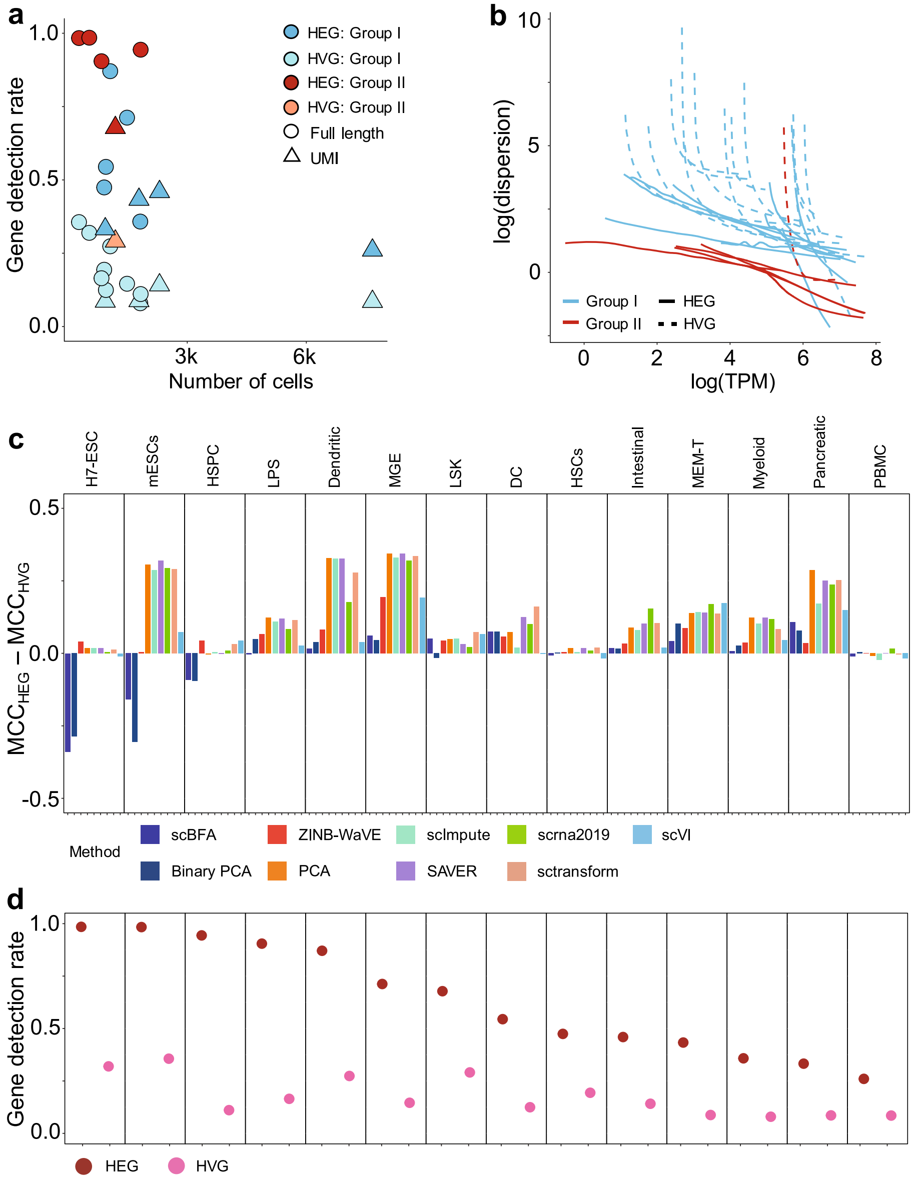

**Fig. 3** Relative scBFA performance is positively correlated with dataset size and high technical noise. **a** Gene detection rate as a function of the number of cells for the 14 benchmarks, when processed using either HVG or HEG selection. Group I benchmarks refers to those datasets in which scBFA is a top performer, and group II benchmarks refers to datasets in which scBFA is a poor performer. Note that each of the 14 benchmarks are represented twice (once under each of HEG and HVG selection). Additional file 1: Table S6 indicates the membership of each benchmark within group I and group II. **b** Same as **a**, except mean-dispersion trends are estimated and visualized for each of the datasets from group I and group II, under HVG and HEG selection criteria. **c** Difference in performance (MCC) of cell type classifiers trained on individual benchmarks and for each method, either using HVG or HEG selection. Performance is assessed through cross-validation of cell type classifiers trained on scRNA-seq data in the respective embedding spaces of each method. Performance is averaged across all number of latent dimensions tested. **d** The corresponding gene detection rate under the two gene selection criteria. Note HEG yields systematically higher GDR compared to HVG.

simulate global differences in gene detection typically observed between different protocols and technologies [6].

We first confirmed that our simulation framework generates datasets with similar characteristics to real datasets. For each of the LSK, HSPC, and LPS benchmarks, we first applied the HVG selection procedure and fit the ZINB-WaVE model. Using the ZINB-WaVE-learned parameters and after setting our additional framework parameters $(\sigma_\mu^2 = 0.5, \sigma_\pi^2 = 0.5, r = 1, \delta = -0.5)$, we then simulated the exact same number of cells as was in the original dataset. Upon performing dimensionality reduction and visualization of both simulated and measured cells simultaneously, we found cells clustered by cell type regardless of whether they were from the real or simulated dataset (Additional file 1: Figures S7-S9), confirming our simulation framework generates realistic datasets.

scBFA consistently outperforms the count-based methods in classifying cell types precisely when the gene detection noise is less than the gene count noise ($\sigma_\pi^2 < \sigma_\mu^2$) (Fig. 4). This observation is robust to the choice of gene dispersion parameter $r$ (Additional file 1: Figures S10-S11) and gene selection procedure (Fig. 4, Additional file 1: Figures S12-S14). On real datasets, we found that scBFA performance increases as the gene detection rate decreases (Fig. 3a), suggesting that in the real datasets for which GDR is low, the count noise may exceed the detection noise.

## scBFA mitigates technical and biological noise in noisy scRNA-seq data

We next tested each method's ability to reduce the effect of technical variation on the learned low-dimensional embeddings by training them on an ERCC-based dataset [29] with no variation due to biological factors. In this dataset, ERCC synthetic spike-in RNAs were diluted to a single concentration (1:10) and loaded into the 10× platform in place of biological cells during the generation of the GEMs. This dataset therefore consists of a single "cell type," with only technical variation present (since the spike-in RNAs were diluted to the same concentration). Additional file 1: Figure S15 illustrates that both scBFA and Binary PCA yield a low-dimensional embedding with minimal variation between "cells" compared to the other methods, suggesting that gene detection models are systematically more robust to technical noise compared to count models.

We also found that modeling gene detection patterns helps to mitigate the effect of biological confounding factors in the scRNA-seq data. For example, a common data normalization step is to remove low-quality cells for which many reads map to mitochondrial genes, as these cells are suspected of undergoing apoptosis [30]. However, finding a clear threshold for discarding cells based on mitochondrial RNA content is challenging (Additional file 1: Figure S16). We found that low dimensional embeddings learned by count-based methods are clearly influenced by mitochondrial RNA content, but this is not true for scBFA (Additional file 1: Figures

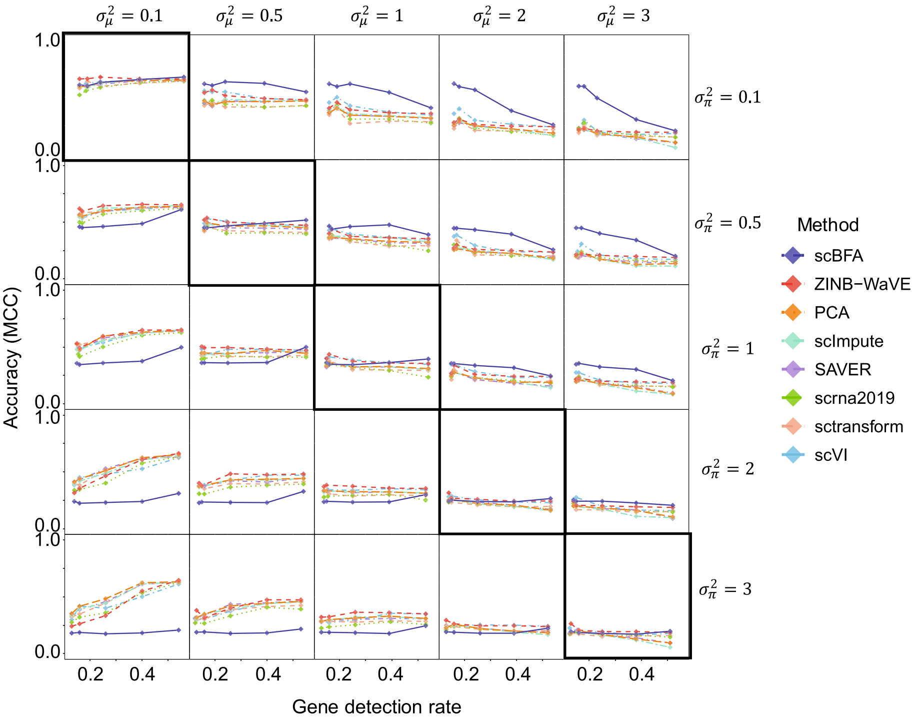

**Fig. 4** scBFA outperforms quantification models when the gene detection noise is less than gene quantification noise. Rows represent different settings of (gene) detection noise ($\sigma_\pi^2$), and columns represent different settings of (gene) quantification noise ($\sigma_\mu^2$). The diagonal represents simulations where the detection noise is equal to the quantification noise ($\sigma_\mu^2 = \sigma_\pi^2$), and the plots above the diagonal represent simulations where the detection noise is less than the quantification noise. Each *y*-axis indicates the cross-validation performance (MCC) of cell type predictors trained on embeddings learned from the simulated data, while each *x*-axis represents the gene detection rate that is manipulated by the parameter $\delta$. Here, the ground truth embedding matrix is obtained by fitting ZINB-WaVE to the LPS benchmark under HVG selection. The dispersion parameter $r$ is set to be 1 in these simulations.

S17-S18), suggesting that scBFA analysis of data will make the downstream analysis more robust to the inclusion of lower-quality cells.

### scBFA embedding space captures cell type-specific markers

We further hypothesized that scBFA performs well at cell type classification in high-quantification noise data because detection pattern embeddings are purely driven by genes only detected in subsets of cells such as marker genes, while this is less true for count models. Marker genes should always be turned off in unrelated cell types and always be expressed at some measurable level in the relevant cells.

To test our hypothesis, we measured the extent to which learned factor loadings capture established cell type markers on the PBMC, HSCs, and Pancreatic benchmarks, for which clear markers could be identified. For these 3 datasets, we identified 41, 43, and 73 markers, respectively, from the literature (Additional file 1: Tables S3-S5). Gene selection reduced the marker sets further to 30, 24, and 43 markers for HVG and 20, 28, and 47 for HEG, respectively. Figure 5 demonstrates that for these 3 datasets, the embeddings of scBFA are driven by cell type markers more than the quantification-based methods, despite the fact that the cell type markers are not used when learning the embeddings. These results also hold when HEG selection is used instead of HVG (Additional file 1: Figure S19).

An important conceptual difference between scBFA and quantification-based methods, such as ZINB-WaVE, is that scBFA treats all zero-count measurements as true observations in which a specific gene is truly not expressed in a given cell. In contrast, ZINB-WaVE and others try to statistically distinguish dropout events from true zero-count measurements. As a result, the ZINB-WaVE model has a gene detection-specific feature matrix and gene count-specific feature matrix component, and we compared the performance of each component individually with respect to the cell type marker identification. Figure 5 illustrates that scBFA factor loading matrix outperforms both components of ZINB-WaVE, suggesting the proportion of false-positive (undetected) zero-count measurements is relatively small and hard to infer statistically.

### Trajectory inference improves with detection modeling

One of the most tantalizing applications of scRNA-seq is trajectory inference for identifying changes in the gene expression during continuous processes such as differentiation [31]. There are on the order of at least 45 methods for trajectory inference [32]. The first step to many trajectory inference methods is dimensionality reduction, of which PCA is a commonly used method [31]. Using a recent benchmark of trajectory inference methods, we identified Slingshot, a top-performing method that uses dimensionality reduction [33]. We evaluated Slingshot's performance on a set of 18 "gold standard" trajectory inference benchmarks after we replaced its PCA step with one of the dimensionality reduction methods we have benchmarked [32]. We found that substituting scBFA in place of PCA led to a systematically higher performance compared to the other methods (ZINB-WaVE, PCA, scImpute, SAVER, scrna2019, sctransform, scVI) (Fig. 6). These results are robust to the performance metric (Fig. 6, Additional file 1: Figure S20).

### Detection pattern models are also superior for scATAC-seq data analysis

Several of the features of scRNA-seq protocols thought to drive technical noise are also shared among other single-cell genomic technologies, such as small starting material and amplification bias. We therefore reasoned that detection-based approaches, such as scBFA, are applicable to other types of single-cell genomic data. We measured the ability of scBFA to cluster cells into cell types using scATAC-seq datasets, which also typically produce highly sparse datasets. scATAC-seq datasets are not typically suitable for input into scRNA-seq analysis tools, because the largest values observed in scATAC-seq data correspond to the ploidy of the genome (e.g., two for humans). However, such sparse, small count data means that transformation into a detection pattern matrix suitable for input into a method such as scBFA will not significantly alter the input data, making scBFA potentially more generalizable than other scRNA-seq analysis tools.

We performed dimensionality reduction and cell type classification experiments on several scATAC-seq datasets, analogous to our scRNA-seq analyses above. We benchmarked scBFA against PCA, Binary PCA, Scasat [34], Destin [35], scABC [36], chromVAR [37], and SCRAT [38]. scBFA systematically outperformed all other methods in our benchmark datasets (Fig. 7, Additional file 1: Figures S21-S24). An important advantage of scBFA over the other scATAC-seq methods is that only scBFA can systematically adjust for the cell-level covariates such as QC measurements (e.g., cell cycle stage) and batch effects. In contrast, other methods, such as Scasat, are unable to remove batch effect in all features since Scasat removes batch effects through removing batch-specific loci, which can be confounded with cell type-specific loci depending on the experimental design. Methods such as Binary PCA cannot directly regress out continuous covariates.

### Detection pattern models can be trained efficiently

The size of scRNA-seq datasets is regularly climbing to new scales each year [2], as newer technologies increase the throughput of cells. With current datasets

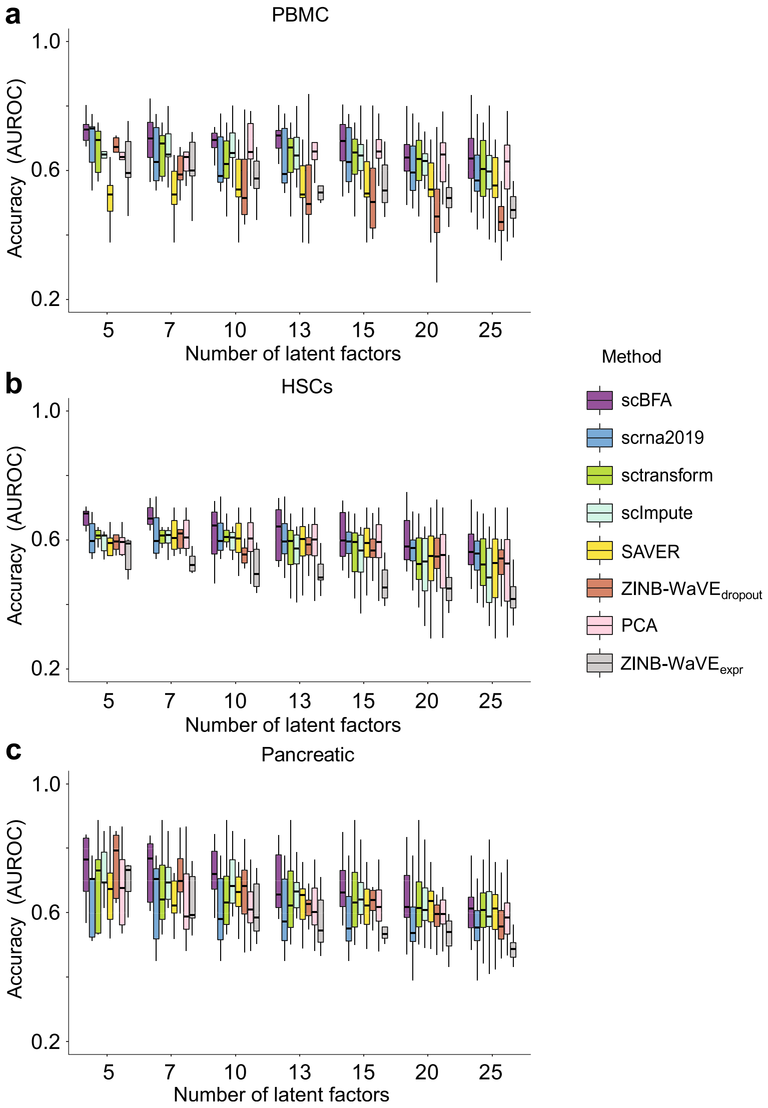

**Fig. 5** scBFA is better informed by cell type markers than quantification models. Each latent factor learned from each method was evaluated based on how much influence established cell type markers exerted on its embeddings, as measured by the area under the curve (AUROC) metric. Each boxplot represents the AUROC of all latent factors for a given method and benchmark. ZINB-WaVE is represented twice, once for the latent dimensions inferred by their gene detection pattern (ZINB-WaVE$_{dropout}$) and once for the latent dimensions inferred from the gene counts (ZINB-WaVE$_{expr}$). **a** PBMC benchmark. **b** HSCs benchmark. **c** Pancreatic benchmark.

occasionally exceeding one million cells, computational efficiency of scRNA-seq analyses becomes challenging as ideally these tools could be run on local machines. We therefore benchmarked methods in terms of their speed of computation. In our comparisons, we also included a fast approximation of scBFA, which we term Binary PCA. Binary PCA is easy to implement in one line of R code (we simply transform the gene counts into the gene detection patterns as a preprocessing step before use of PCA) and provides immediate benefits over standard PCA and other methods with respect to cell type identification (Additional file 1: Figures S25-S26). We found that Binary PCA is tied for the fastest of all methods, while scBFA is still faster than several competing count-based methods (Additional file 1: Figure S27). More specifically, scBFA is a median of ten times faster than ZINB-WaVE. The difference in the execution time between scBFA and ZINB-WaVE is due primarily to the additional burden of modeling gene quantification because the scBFA model structure and parameter learning algorithm were designed to match the gene detection pattern component of ZINB-WaVE as closely as possible. This suggests that gene detection models may help analysis tools scale to larger datasets in the future.

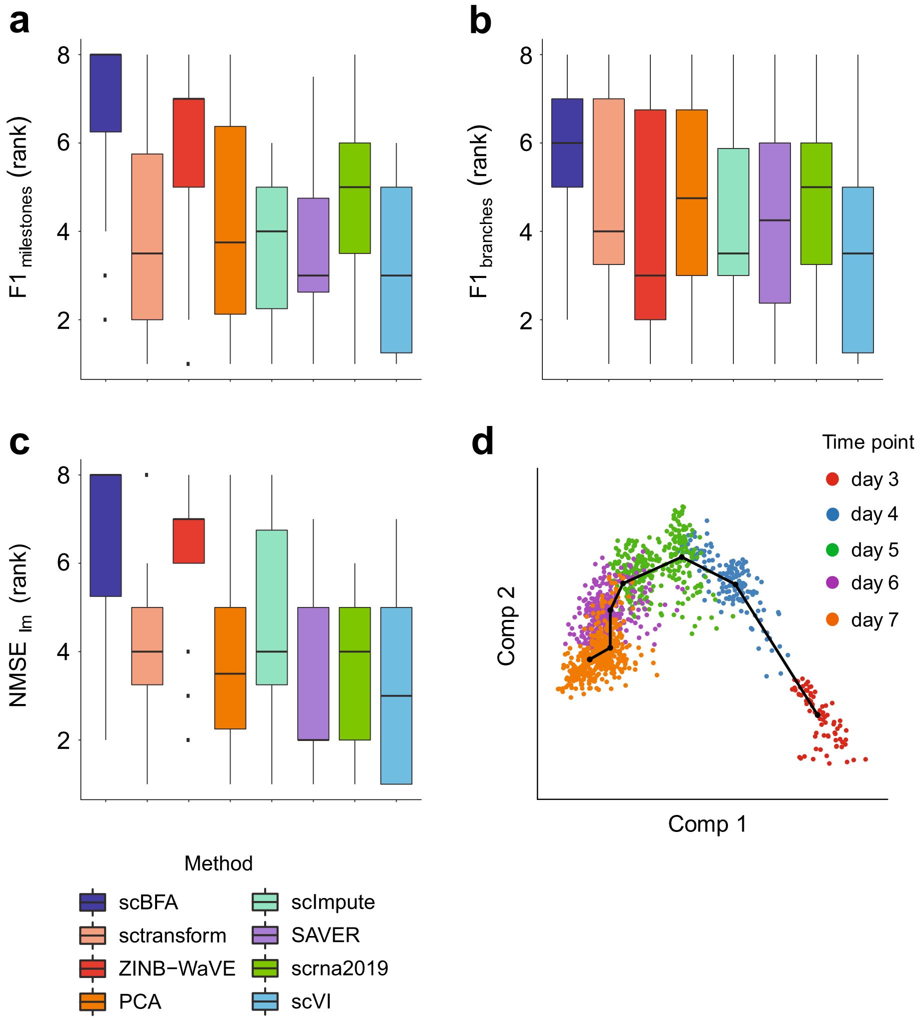

**Fig. 6** scBFA leads to the most improvement in trajectory inference performance of Slingshot. **a**–**c** Slingshot was modified by replacing the PCA step of the original Slingshot method with each of the dimensionality reduction methods tested. The *y*-axis shows the distribution of the overall ranks (higher rank is better) of the modified versions of Slingshot. Methods were evaluated across 18 "gold standard" benchmarks and using 3 different performance metrics, $F1_{milestone}$, $NMSE_{lm}$, and $F1_{branch}$, that measure how well the inferred trajectory matches the ground truth trajectory. $F1_{milestone}$ and $F1_{branch}$ are based on the quality of clustering of cells in the trajectory, while $NMSE_{lm}$ assesses how well the position of a cell in the inferred trajectory predicts the position of the cell in the ground truth trajectory. Across the three evaluation metrics and 18 benchmarks, scBFA yields better overall performance (rank). **d** A 2D scatter plot of scBFA's first two components, visualizing the inferred trajectory corresponding to the embryo development time in the H-embryos dataset.

## Discussion

Our primary result is that when the count (quantification) noise is relatively high in a dataset as is typical in larger datasets, the effects of this noise on the downstream analysis can be mitigated by modeling detection patterns instead of counts. The improvement in the performance of scBFA over ZINB-WaVE in this regime (Figs. 1 and 3) is particularly informative because the ZINB-WaVE model has two components: one that models gene detection and the other that models gene counts (quantification). The model structure and parameter learning algorithm of scBFA are 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 quantification (ZINB-WaVE) or gene detection (scBFA) is modeled.

We show that as the number of cells sequenced increases within a dataset, the technical noise in the data (as measured indirectly by the gene detection rate and gene-wise dispersion) and the relative performance of scBFA generally increase as well. Because many scRNA-seq applications benefit from higher numbers of sequenced cells, there is a steep upward trend of scRNA-seq dataset sizes, with some recent datasets containing nearly a million cells [2]. Our results therefore suggest

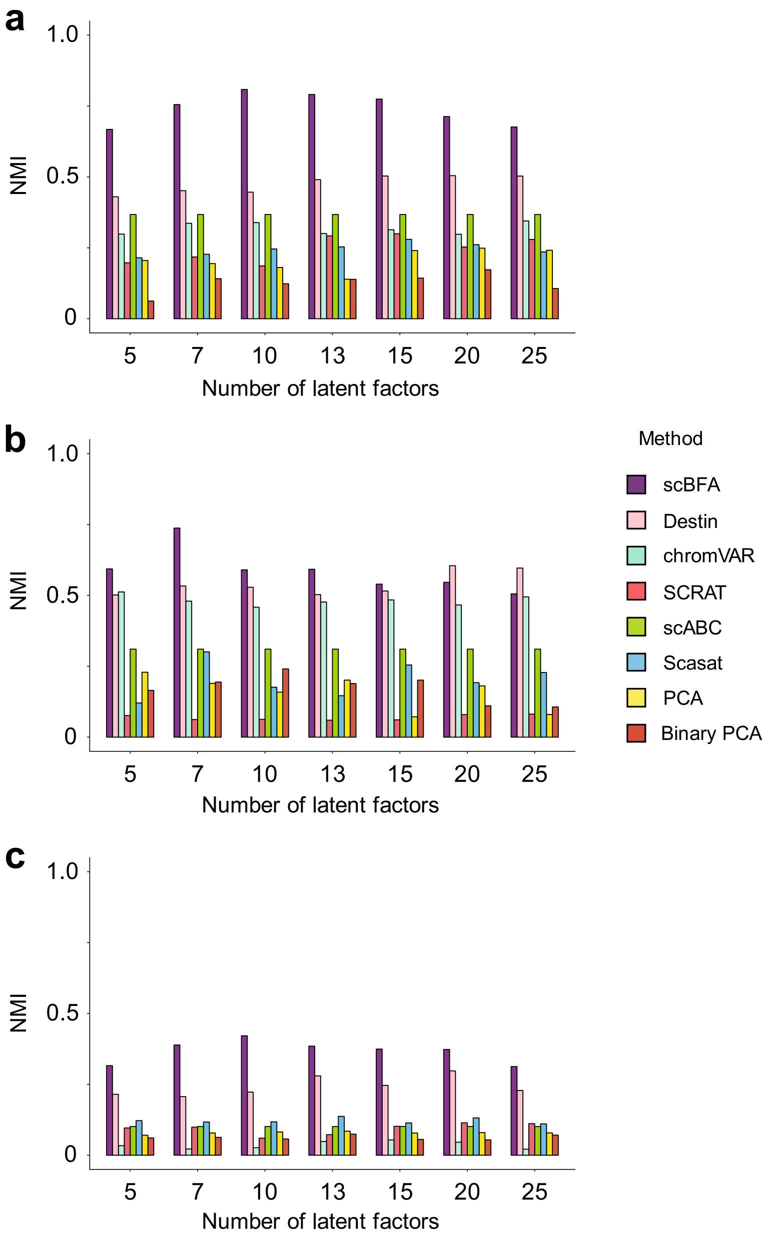

**Fig. 7** scBFA more accurately recovers cell type identity in scATAC-seq datasets. Clustering accuracy (NMI) of each scATAC-seq method trained on a given benchmark, as a function of the number of latent dimensions specified. **a** GSE96769. **b** GSE74310. **c** GSE107816.

that it is increasingly important that next-generation scRNA-seq analysis tools exploit the advantages of gene detection-only modeling in order to mitigate the effects of technical noise within the data. Also, given the influence of technology and protocol choice on technical noise in scRNA-seq data [6], our results imply that future scRNA-seq tools could be designed to take advantage of the specific noise structure implied by different scRNA-seq protocols, as opposed to being relatively protocol-agnostic as they are today.

While it is challenging to measure technical noise in real datasets, we show that the gene detection rate and gene-wise dispersion are easily calculated and serve as good proxies for measuring technical noise. In our results, we found that the performance of scBFA exceeds that of gene count analysis tools in cell type classification when the gene detection rate falls below 90% and dispersion estimates are high, therefore providing the community with a general guideline for when detection-based tools, such as scBFA, should be used instead of quantification-based tools. Our R package also has implemented a function, diagnose, to assist users in determining whether scBFA is appropriate for their data. Our results are also consistent with previous work that shows tasks, such as dimensionality reduction, cell type identification, and abundance estimation, can be performed successfully when individual cells are sequenced to shallow depth [4, 21–23] and further provide a complementary analysis approach suitable for these datasets with low per-cell sequencing depth.

There is a plethora of data normalization methods that have been, and continue to be, designed to decrease technical noise within and across cells, in order to better perform both gene detection and quantification, and to make these quantities comparable across cells (see [5, 39, 40] for an overview). The challenge we address here is not mitigated by data normalization methods; however, as we argue that when the number of UMIs sequenced per cell decreases drastically, gene quantification information specifically is not present (or useable) in the data, which is a problem that data normalization cannot mitigate. Data normalization works complementarily to gene detection pattern analysis however, as illustrated by our use of data normalization before gene detection modeling in this work.

Our results also imply that models that statistically distinguish dropout events from genes truly not expressed in a cell may be less fruitful for large datasets. Many gene count-based methods [9, 24, 41] model zero counts as a mixture of genes truly turned off (biological signal) and genes that are truly expressed but not detected due to technical artifacts from the experimental protocol (technical noise) [41]. On the contrary, gene detection pattern methods, such as scBFA, treat all zeroes as a biological signal, a key feature motivated by the observation that zero measurements driven by technical noise tend to occur for genes that are poorly expressed [10, 11]. The superior performance of scBFA when the gene detection rate is low suggests that for these datasets, there is not enough information in the gene counts to reliably detect technical dropout events, and therefore, traditional mixture modeling can be unhelpful for high-throughput datasets where gene detection rates are low overall.

The success of modeling gene detection patterns in scRNA-seq is not tied to a specific model structure. The performance improvement of scBFA over ZINB-WaVE, and Binary PCA over PCA, demonstrates our results hold across multiple model structures and loss functions. In both cases, not only do we observe performance gains for large scRNA-seq datasets, but there is also a substantial speed improvement because detection modeling avoids complex parametric modeling of gene counts, making detection models scalable to larger datasets. Within the class of gene detection-based models, Binary PCA provides a moderately accurate but much faster and simpler implementation scheme that can be achieved in one line of code, making our results readily achievable by the current analysis pipelines.

A surprising finding was that HEG gene selection led to a systematically better cell type identification for every tested method in almost all datasets, compared to HVG selection (Fig. 3c). HVG selection anecdotally is the standard criterion upon which variable genes are typically selected during preprocessing [42], suggesting at least for cell type identification, HEG selection may lead to improved performance regardless of the method used.

While single-cell genomic data from different modalities, such as scATAC-seq, have similar data structure as scRNA-seq data, the analysis tools and pipelines developed to date for those two technologies are largely mutually exclusive. Here, we show that scBFA generalizes to other single-cell genomic modalities and outperforms the existing methods for cell type identification for scATAC-seq datasets as well, even those that take advantage of auxiliary data such as transcription factor motifs and distance to transcription start sites [35]. We expect our results to generalize to other single-cell genomic modalities such as single-cell methylation or histone modification data.

## Methods

### Single-cell binary factor analysis (scBFA) model

scBFA is available as an R package on Bioconductor at https://bioconductor.org/packages/devel/bioc/html/scBFA.html, as well as on GitHub (https://github.com/quon-titative-biology/scBFA). The main function to run scBFA is scBFA().

In our notation below, matrices are represented by upper case bold letters, vectors by lower case bold letters, and numeric constants as upper case non-bold letters. Square brackets also indicate a matrix, though represented as a series of column vectors. A matrix subscript with round brackets indicates the index of the corresponding column vector.

The schematic of our single-cell binary factor analysis (scBFA) model is shown in Additional file 1: Figure S28. The input data to scBFA consists of two matrices, $\boldsymbol{O}$ and $\boldsymbol{X}$. $\boldsymbol{O}$ is a matrix of counts, consisting of $G$ features (genes in the case of scRNA-seq data, or loci in the case of scATAC-seq data) measured in each of $N$ samples (cells). From the input data $\boldsymbol{O}$, we compute a matrix $\boldsymbol{B}$, where $B_{ij}$ represents the detection pattern observed for cell $i$ ($i = 1, \ldots, N$) and feature $j$ ($j = 1, \ldots, G$). For scRNA-seq inputs, $B_{ij} = 1$ when $O_{ij} \geq 1$, otherwise $B_{ij} = 0$. Therefore, $B_{ij} = 1$ indicates that at least one read (or UMI) maps to gene $j$ in cell $i$ and therefore suggests gene expression. Similarly, for scATAC-seq input data, $B_{ij} = 1$ when $O_{ij} \geq 1$, in other words, when at least one read maps to locus $j$ in cell $i$ (and therefore suggests locus accessibility), otherwise $B_{ij} = 0$. scBFA is adapted from a generalized linear model framework and is therefore capable of adjusting for batch effects and other nuisance cell-level covariates. Input $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2,$

…, $\boldsymbol{x}_N]^T$ is a $N \times C$ cell-level covariate matrix that enables correction for $C$ observed nuisance factors such as batch effects or other cell-specific quality control measurements. If there are no such cell-level covariates that need to be adjusted for, $\boldsymbol{X}$ is the null matrix by default.

The intuition behind scBFA is that it performs dimensionality reduction to explain the high-dimensional detection pattern matrix $\boldsymbol{B}$ by estimating two lower-dimensional matrices: a $N \times K$ embedding matrix $\boldsymbol{Z} = [\ \boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_N}]^T$,and a $K \times G$ loading matrix $\boldsymbol{A} = [\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_G}]$. Here, $K$ is the number of latent dimensions used to approximate $\boldsymbol{B_{ij}}$, where where $K \ll G$. $u_i$ and $v_i$ and represent the $i$th cell-level intercept and $j$th feature-specific intercept, respectively. $\boldsymbol{u}$ is therefore a vector of length $N$, and $\boldsymbol{v}$ is a vector of length $G$. For scRNA-seq, for example, we expect $\boldsymbol{u}$ and $\boldsymbol{v}$ will implicitly model the variation of gene expression caused by library size. $\mu_{\text{ij}}$ is the mean of the Bernoulli distribution governing whether feature $j$ is detected in cell $i$ or not.

Formally, scBFA is defined by the following model:

$$\operatorname{logit}(\mu_{ij}) = \boldsymbol{x}_i^T \boldsymbol{\beta}_j + \boldsymbol{z}_i^T \boldsymbol{a}_j + u_i + v_j$$

$$P(\boldsymbol{B}; \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{u}, \boldsymbol{v}) = \prod_{i,j} \text{Bernoulli}\left(B_{ij} | \mu_{ij}, \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{u}, \boldsymbol{v}\right)$$

We train the scBFA model by optimizing the following penalized likelihood function:

$$f(\boldsymbol{B}, \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{u}, \boldsymbol{v}) = \left[\sum\nolimits_{i,j} \ln P\left(B_{ij}; \boldsymbol{A}, \boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{X}, \boldsymbol{u}, \boldsymbol{v}\right)\right] - \epsilon_1 \|\boldsymbol{A}\|_2^2 - \epsilon_2 \|\boldsymbol{Z}\|_2^2 - \epsilon_3 \|\boldsymbol{\beta}\|_2^2$$

Here, $\epsilon_1$, $\epsilon_2$, and $\epsilon_3$ are the tunable parameters that control the regularization of the model parameters, where by default $\epsilon_1 = \frac{\epsilon_0}{G}, \epsilon_2 = \frac{\epsilon_0}{N}$, $\epsilon_3 = \frac{\epsilon_0}{G}$, and $\epsilon_0 = \max\{N, G\}$. The optimization is carried out using the L-BFGS-B optimization routine available in the R optim() function. After completing the optimization, we orthogonalize $\boldsymbol{Z}$ and $\boldsymbol{A}$ using the orthogonalizeTraceNorm() function available in the ZINB-WaVE [8] package.

### Binary PCA model and calculation of the gene detection pattern matrix

Binary PCA describes our fast approximation to scBFA by simply running PCA, with the exception of converting the input count matrix into a detection matrix by converting non-zero values to one. We implemented Binary PCA through the addition of a single line of R code. Suppose that countMatrix is the name of the matrix in R that stores, for example, the UMI counts for each gene in each cell. To run Binary PCA, we first convert the countMatrix into the gene detection pattern matrix before running PCA via the R command:

```
countMatrix[which(countMatrix>0)] <- 1
```

We then call PCA using the following command in R:

```
PCA_results <- prcomp(countMatrix, center=TRUE, scale.=FALSE);
```

Note that typically, scale is set to TRUE when calling PCA. For Binary PCA, we set it to FALSE because the variance in gene detection is potentially associated with cell types (e.g., genes with higher detection variance are more likely to be marker genes, and therefore should contribute more to the embedding).

### Execution of scRNA-seq analysis methods

We compared scBFA against scVI [12], SAVER [13], sctransform [14], scrna2019 [15], PCA, ZINB-WaVE [8], and scImpute [9]. These seven methods were selected to represent diverse classes of approaches to scRNA-seq data analysis, including dimensionality reduction methods (PCA, ZINB-WaVE, scVI), preprocessing approaches that can be applied before dimensionality reduction (sctransform, scrna2019), and imputation methods that can be applied before dimensionality reduction (SAVER, scImpute). Of the dimensionality reduction methods, PCA was chosen because of its implementation in popular packages such as Seurat [42], and scVI [12] is a leading deep learning-based dimensionality reduction method. ZINB-WaVE was chosen specifically because it is a recently developed method, and scBFA is designed as a gene detection-based analog of ZINB-WaVE; therefore, comparing scBFA with ZINB-WaVE is the fairest comparison of gene detection versus quantification-based approaches.

We ran most of the scRNA-seq analysis methods with their default parameter settings, with the exception of scVI and scrna2019.

scVI requires specification of a learning rate and the number of iterations before convergence. During training, we found that scVI performance was heavily influenced by these two parameters. We therefore performed an unbiased grid search by setting the number of iterations to be either 2000 or 4000 and setting the learning rate to be either 1e−2, 1e−3, or 1e−4. We then trained the model with all 6 possible combinations of learning rate and number of iterations, and selected the combination of parameters that gave the lowest loss on the hold-out set. The loss value is provided by scVI during training. During training, the size of the training set is fixed to be 75% of the entire dataset, and the remaining parameters are fixed at their default values. We repeated the above parameter search for the same number of factors as the other methods for all scRNA-seq datasets.

For the simulated datasets, given the large number of scenarios tested, we fixed the learning rate to be 0.001 and number of iterations to be 2000.

scrna2019 is a method developed to perform feature selection and GLM-based factor analysis on scRNA-seq [15]. The scrna2019 R package (obtained on May 6, 2019, from https://github.com/willtownes/scrna2019) offers both a GLM factor analysis model and a corresponding deviance score approximation. We used the deviance score approximation instead of the GLM framework for our experiments because several benchmarks required batch effect correction, which should be addressed using the deviance score approximation as per scrna2019's authors' recommendations [15]. Also, at the time of the writing of this paper, the GLM implementation produced errors for three of our datasets that prevented us from completing our experiments.

### Execution of scATAC-seq analysis methods

We compared scBFA against PCA, Binary PCA, Scasat [34], Destin [35], scABC [36], chromVAR [37], and SCRAT [38]. Scasat and Destin are scATAC-seq analysis tools primarily designed to identify cell types and differential accessibility analysis. Both methods treat dimensionality reduction as a prior step before further clustering distinct cell types. Scasat's embedding space is learned by performing multidimensional scaling (MDS) on a cell-cell Jaccard similarity matrix computed from a binarized chromatin accessibility matrix. Destin developed a weighted principal component analysis approach using distance to transcription start sites and reference regulomic data. scABC is an unsupervised clustering tool of single-cell epigenetic data and performs multi-stage clustering based on the input chromatin accessibility matrix directly. chromVAR aggregates motif position weight matrices (PWM) and chromatin accessibility to uncover cell populations and identify motifs associated with cell type-specific variation. SCRAT summarizes several distinct regulatory genomic data (including prior established gene sets and transcription factor binding motif sites, among others) to identify distinct cell populations from single-cell genomic data.

For SCRAT, we used the regulatory activity feature list provided by SCRATsummary() as the default input features. In addition, we also input the BED files corresponding to the scATAC-seq data as a custom feature as suggested by the SCRAT authors [38], which we found to improve the performance.

### Quantifying the effect of imputation on scBFA

We compared the performance of scBFA before and after imputation on our 14 benchmark datasets under HVG selection. We tested two state-of-the-art imputation methods, SAVER [13] and scImpute [9]. SAVER estimates library size-normalized posterior means of gene expression levels ($\hat{\lambda}_{ij}$), which are inappropriate for input into scBFA because they are not sparse. We therefore sampled counts from SAVER's generative model as follows:

$$O_{ij} \sim \mathrm{Poisson}\left(s_i \hat{\lambda}_{ij}\right)$$

where $\hat{\lambda}_{ij}$ is SAVER's imputed expression level and $s_i$ is the library size for cell $i$ divided by the mean library size across cells [13]. We generated five separate count matrices $O_{ij}$ based on the SAVER estimates $\hat{\lambda}_{ij}$. For scImpute, we used its imputed gene counts matrix directly as input for scBFA.

### Selection of representative datasets to measure gene detection rates

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 (Additional file 1: Figure S1). We obtained these datasets from two sources, the conquer database [43] and the Gene Expression Omnibus (GEO) [44]. 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 (Additional file 1: Table S7).

### Computing mean and dispersion curves

We use the DESeq2 [28] package to obtain gene-specific dispersion estimates for each dataset, where dispersion is measured across all cells in a dataset. Within the DESeq2 pipeline, gene-wise dispersions are first estimated, a trend line is fit to the gene-wise dispersion estimates, and finally, shrinkage is applied to the gene-wise dispersion estimates (MAP estimates). In Fig. 3b, we extracted the fitted gene-wise dispersion estimates from the trend line (second step), and we fit these dispersion estimates by local linear regression (LOESS) using the gene-wise mean of transcripts per million (TPM) across all cells as the explanatory variable. To address the border effect of LOESS, we removed the top and bottom 2.5% of genes as ranked by TPM. Note that using the MAP dispersion estimates (final DESeq2 step) or the fitted dispersion estimates from the trended fit (second step) does not materially change our conclusion. The exception is for the dataset PBMC where there are 455 genes with its MAP dispersion estimates staying at their initialized value of 1e−8 during optimization, while their fitted dispersion estimates are substantially different. We therefore chose the fitted dispersion estimates to generate Fig. 3b.

### Benchmarking dimensionality reduction methods for scRNA-seq

We evaluated each dimensionality reduction method by how well their low dimensional embeddings discriminate experimentally defined cell types. For each dataset and method tested, we first performed dimensionality reduction on the entire dataset to obtain an embedding matrix representing each cell in $K$ dimensions (the matrix $\boldsymbol{Z}$ described in the scBFA methods section). We then performed fivefold cross-validation in which we trained a non-regularized multi-level logistic classifier on the training embeddings from each method using the a priori known cell type labels, then used the model to predict cell type labels for the test embeddings. For every prediction, using the known cell type labels, we computed a confusion matrix and the corresponding Matthews' correlation coefficient (MCC) as a measure of classification accuracy. MCC was calculated using the R package mltools. We repeated the fivefold cross-validation a total of 15 times and reported the mean classification accuracy as the final accuracy.

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

In our analysis of the ERCC dataset, we used a different evaluation metric because each "cell" represents technical replicates of the spiked-in RNA diluted at a constant ratio (10:1). Under the assumption that the only variation between "cells" is due to technical factors, we therefore used averaged within-group sum of squares (AWSS) to measure how the low-dimensional embedding learned by each method captured such homogeneity. Given an $N$ by $K$ embedding matrix $Z$, AWSS was calculated as follows:

$$W = \left(Z - \overline{Z}\right)$$
$$\mathrm{AWSS} = \frac{\mathrm{trace}\left(W^T W\right)}{N - 1}$$

Here, $\overline{Z}$ is an $N$ by $K$ matrix for which every row is the column mean of $Z$.

### Benchmarking cell type identification methods for scATAC-seq

We benchmarked scBFA against existing scATAC-seq analysis tools by evaluating their ability to correctly cluster cell types. We used a different evaluation scheme from that used for the scRNA-seq experiments because one of the existing methods (scABC) does not produce low-dimensional embeddings and instead outputs cluster labels. The methods Scasat and Destin both provide cluster labels directly from their analysis pipeline. For scBFA, PCA, Binary PCA, chromVAR, and SCRAT, we clustered cells based on the learned embedding matrices using R's built-in hierarchical clustering function hclust() with Wald's distance. We compared the accuracy of the clustering results from each method using the metrics normalized mutual information (NMI) and Adjusted Rand Index (ARI), computed using the R package aricode.

### Simulation of scRNA-seq data

A variation of the ZINB-WaVE model was used to simulate scRNA-seq datasets and is defined as follows (Additional file 1: Figure S29):

$$\boldsymbol{z}_{\boldsymbol{\mu}(\boldsymbol{i})} \sim N\left(\hat{\boldsymbol{z}}_i, \sigma_\mu^2 \boldsymbol{I}_{\boldsymbol{K}}\right)$$

$$\boldsymbol{z}_{\boldsymbol{\pi}(\boldsymbol{i})} \sim N\left(\hat{\boldsymbol{z}}_{\boldsymbol{i}}, \sigma_{\boldsymbol{\pi}}^2 \boldsymbol{I}_{\boldsymbol{K}}\right)$$

$$\mathrm{logit}\left(\pi_{ij}\right) = \left(\boldsymbol{z}_{\boldsymbol{\pi}(\boldsymbol{i})}^{\boldsymbol{T}} \hat{\boldsymbol{a}}_{\boldsymbol{\pi}(\boldsymbol{j})} + \hat{u}_{\pi(i)} + \hat{\nu}_{\pi(j)} - \delta\right)$$

$$\log\left(\mu_{ij}\right) = \left(\boldsymbol{z}_{\boldsymbol{\mu}(\boldsymbol{i})}^{\boldsymbol{T}} \hat{\boldsymbol{a}}_{\boldsymbol{\mu}(\boldsymbol{j})} + \hat{\mu}_{\mu(i)} + \hat{\nu}_{\mu(j)}\right)$$

$$\Pi_{ij} \sim \mathrm{Bernoulli}(\pi_{ij})$$

$$O_{ij} \begin{cases} = 0 & \text{if } \Pi_{ij} = 1 \\ \sim \mathrm{NB}\left(\mu_{ij}, r\right) & \text{if } \Pi_{ij} = 0 \end{cases}$$

To keep the consistency of the notation, the parameters we used above $\{\hat{\boldsymbol{Z}}, \hat{\boldsymbol{A}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{A}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{u}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{u}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{\pi}}\}$ respectively correspond to the parameters $\{\hat{\boldsymbol{W}}, \hat{\boldsymbol{\alpha}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\alpha}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{\gamma}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{\gamma}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\pi}}\}$ used in the original ZINB-WaVE paper. In the first step of our simulations, all parameters with a hat accent are set a priori by fitting the ZINB-WaVE model using its R package [8] on a single scRNA-seq dataset in order to use realistic parameters for our simulation. The remaining parameters $\{\delta, \sigma_\mu^2, \sigma_\pi^2, r\}$ are then systematically varied in our simulations to determine their effect on downstream dimensionality reduction methods. $O_{ij}$ denotes the gene counts for cell $i$ and feature $j$. As is described in the original ZINB-WaVE paper, $\hat{\boldsymbol{Z}} = [\hat{\boldsymbol{z}}_{\boldsymbol{1}}, \hat{\boldsymbol{z}}_{\boldsymbol{2}}, \ldots, \hat{\boldsymbol{z}}_{\boldsymbol{N}}]^{\boldsymbol{T}}$ is a $N \times K$ embedding matrix, while $\hat{\boldsymbol{A}}_{\boldsymbol{\mu}} = [\hat{\boldsymbol{a}}_{\boldsymbol{\mu}(\boldsymbol{1})}, \hat{\boldsymbol{a}}_{\boldsymbol{\mu}(\boldsymbol{2})}, \ldots, \hat{\boldsymbol{a}}_{\boldsymbol{\mu}(\boldsymbol{G})}]$ and $\hat{\boldsymbol{A}}_{\boldsymbol{\pi}} = [\hat{\boldsymbol{a}}_{\boldsymbol{\pi}(\boldsymbol{1})}, \hat{\boldsymbol{a}}_{\boldsymbol{\pi}(\boldsymbol{2})}, \ldots, \hat{\boldsymbol{a}}_{\boldsymbol{\pi}(\boldsymbol{G})}]$ are the corresponding $K \times G$ regression coefficient matrices for the negative binomial and Bernoulli distributions governing the gene count and detection components, respectively. The output of the Bernoulli distribution is the latent variable $\Pi_{ij}$, which decides whether a gene is detected (in which case the observed value $O_{ij}$ is sampled from a negative binomial distribution), or not detected. $\hat{\boldsymbol{u}}_{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{u}}_{\boldsymbol{\pi}}$ are $N \times 1$ cell-specific intercepts for the count matrix and detection matrix, respectively. Similarly, $\hat{\boldsymbol{\nu}}_{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\nu}}_{\boldsymbol{\pi}}$ are $G \times 1$ gene-specific intercepts for the count matrix and detection matrix, respectively. The number of latent dimensions $K$ used to generate the gene expression values was fixed at 5, and we used a total of 2000 highly variable genes as in the original dataset. The LPS dataset does not provide any cell-level covariates, so in these simulations, there are no cell- or gene-wise covariate matrices. For quality control purposes, we filtered out genes that are expressed in fewer than 1% of the cells and then filtered out cells in which less than 1% of genes are expressed.

The distinction between our simulation framework and ZINB-WaVE is that ZINB-WaVE maintains the same cell embedding space $\hat{\boldsymbol{Z}}$ across both the gene detection and count spaces. In contrast, our framework relaxes this constraint by introducing individual embeddings $\boldsymbol{Z_\pi}$ and $\boldsymbol{Z_\mu}$ that are close to $\hat{\boldsymbol{Z}}$. Formally, $\boldsymbol{Z_\pi}$ and $\boldsymbol{Z_\mu}$ are $N \times K$ embedding matrices for the gene detection and count spaces, respectively. Each row $i$ of $\boldsymbol{Z_\pi}$ and $\boldsymbol{Z_\pi}$ are sampled from respective $K$-multivariate Gaussian distributions with the same mean defined by $\hat{\boldsymbol{z}}_i$ and spherical variance parameters $\sigma_{\boldsymbol{\pi}}^2$ and $\sigma_{\mu}^2$, respectively.

In our simulations, we varied the simulation parameters $\{\delta, \sigma_{\mu}^2, \sigma_{\boldsymbol{\pi}}^2, r\}$ as follows. To influence the total number of gene counts detected (total detection rate), we set $\delta \in \{-2, -0.5, 1, 2.5, 4\}$. To influence the variance in the gene detection and count embedding spaces, we set $\sigma_{\pi}^2, \sigma_{\mu}^2 \in \{0.1, 0.5, 1, 2, 3\}$. Finally, we varied the common gene dispersion parameter $r \in \{0.5, 1, 5\}$. In total, the number of unique parameter settings we used to simulate scRNA-seq data is $5 \times 5 \times 5 \times 3 = 375$. For each of those scenarios, we simulated 3 replicates, resulted in $375 \times 3 = 1125$ datasets.

### Quality control of scRNA-seq data

For each scRNA-seq dataset tested, we performed a standardized quality control process. We first removed cells for which mitochondrial genes accounted for over 50% of the total observed counts. Then, we filtered out genes that are expressed in fewer than 1% of cells and removed cells whose library size (total read or UMI count) was less than one-eighth quantile of all cell library sizes. One exception is the MEM-T cell dataset, where we removed an extra 361 cells from the batch labeled "subject16" to remove batches that were confounded with cell types.

### Preprocessing of scATAC-seq data

We followed the scATAC-seq pipeline for processing and aligning reads used by the Destin method [35], obtained from GitHub at https://github.com/urrutiag/destin on April 22, 2019. This preprocessing pipeline yielded 2779, 576, and 960 BAM files for GSE96769 [45], GSE74310 [46], and GSE107816 [47], respectively. These BAM files form the initial input of Destin, scABC, and SCRAT. The input chromatin accessibility matrix for chromVAR and Scasat was then obtained from Destin's preprocessing pipeline directly.

For GSE96769, we only kept cells and genomic loci that are used in the original paper's analysis. The indices for genome loci and cells that passed quality control are supplied in the supplementary files of the original paper. Beyond that, we selected a subset of frozen cells from five patients, excluding patient BM0106, and a subset of pDC cells from patient BM1137 to keep as many samples as possible while removing the part of the batches confounded with cell types. This enabled us to construct a design matrix to correct for patient-specific effects. We furthermore excluded cells that are labeled as unknown by the original author.

For each scATAC-seq dataset tested, we only kept genomic loci that are accessible in at least 1% of cells and then removed cells with a total number of accessible sites that deviates more than 3 standard errors to the mean (in either direction) across all cells. The number of retained cells used as input in our downstream analysis was 1358 for GSE96769, 572 for GSE74310, and 929 for GSE108716. We found that SCRAT and chromVAR's preprocessing pipeline generated NA values, and so for these tools, we filtered out additional cells. For SCRAT, this yielded 1375 cells for GSE96769, 534 cells for GSE74310, and 815 cells for GSE108716. For chromVAR, this yielded 1358 cells for GSE96769, 529 cells for GSE74310, and 811 cells for GSE108716.

### Defining cell type labels in benchmark datasets

Most benchmark datasets used in our analyses were selected because the cell types were already defined in the original study by either known experimental condition or via cell surface markers. However, for the PBMC dataset, Stoeckius et al. [48] collected single-cell antibody-derived tag (ADT) data as well as scRNA-seq using CITE-seq [48]. ADTs can be viewed as a digital readout of cell surface protein abundance. We defined the cell types within this dataset by performing Louvain clustering on the Jaccard similarity matrix constructed based on the normalized ADT levels, similar to Stoeckius et al. [48]. Louvain clustering was performed using the "cluster_louvain" function implemented in the igraph R package. Clustering identified 10 cell types automatically. Note that the quality control standard for this dataset is different compared to the other scRNA-seq datasets used in our analysis, as cells were required to pass both scRNA-seq-specific filters (minimum of 800 reads) and ADT-specific filters (minimum of 50 ADT counts).

### Normalization of scRNA-seq data

For each method, we also normalized cells to control for differences in library size. For PCA, we normalize the counts by setting $\tilde{O}_{ij} = \log(\frac{O_{ij}}{c_i} + 1)$, where $\tilde{O}_{ij}$ is the normalized gene count for cell $i$ and gene $j$, $O_{ij}$ is the original gene count for cell $i$ and gene $j$, and $c_i = \sum_j O_{ij}$ is library size for cell $i$. ZINB-WaVE directly accounts for library size via their cell-specific intercept. For scImpute, we used the total number of imputed counts per cell as their corresponding library size and normalized in the same way as PCA. For scBFA, we estimated the feature-specific intercepts and cell-specific intercepts to implicitly model the effect of library size. SAVER uses the library size divided by the median library size across all cells to adjust for cell size. sctransform uses the log of the library size in its model. scrna2019 outputs a transformed deviance score matrix that does not depend on library size as input.

### Normalization of scATAC-seq data

For PCA, we performed a log transformation $\tilde{O}_{ij} = \log(O_{ij} + 1)$ to adjust the counts within scATAC-seq, where $O_{ij}$ is the original read count for cell $i$ and locus $j$. For scBFA, Scasat, Destin, scABC, Binary PCA, chromVAR, and SCRAT, no extra normalization was applied.

### Gene selection in scRNA-seq data

Highly variable genes (HVG) selection was performed to identify the most overdispersed genes, that is, genes that exhibit more variance than expected based on their mean. The HVG selection was performed using the FindVariableFeatures command implemented in Seurat 3.0. By default, Seurat selected the top 2000 genes. Highly expressed gene (HEG) selection was performed to identify the genes that exhibit the highest variance across cells, regardless of their mean, and is therefore expected to capture genes with higher mean expression compared to HVGs. To identify HEGs, we calculated the gene-specific variance in the gene count space and select the top 2000 genes to make the set size comparable to HVGs.

The gene detection rate (the average fraction of cells in which a gene is detected as expressed) and gene-wise dispersion of each dataset calculated in Fig. 3b is based on these 2000 most variant genes under both the HVG and HEG selection criteria. For the timing experiment, we only selected the top 1000 genes under the HVG criterion for computational speed.

### Batch effect correction

For both scRNA-seq and scATAC-seq datasets, we performed two types of batch effect correction, depending on how the cell types are distributed across the batches in the dataset. For datasets where all cell types are represented in all batches (e.g., replicates, patients), such as the HSC dataset, we used those cell-level covariates to define the $N \times C$ design matrix $\boldsymbol{X}$ (see the scBFA model details above). For ZINB-WaVE, scBFA, and scVI, we regressed $\boldsymbol{X}$ out within the model structure. Since PCA does not offer a framework to regress out nuisance factors, we first regressed $\boldsymbol{X}$ directly from the normalized counts $\tilde{O}_{ij}$ using a linear model. We then applied PCA on the residual matrix and obtained the corresponding embeddings and factor loading matrix. For Binary PCA, scImpute, SAVER, sctransform, and scrna2019, we also regressed out $\boldsymbol{X}$ from the binary entries and imputed values separately, then used the residual matrix in the same way as for PCA.

For other datasets (MEM-T, Pancreatic, MGE in scRNA-seq, and GSE96769 and GSE74310 in scATAC-seq), some batches were missing a subset of cell types, resulting in a design matrix $\boldsymbol{X}$ that cannot be directly used to estimate all batch effects. In this scenario, our strategy for modifying the dataset to address batch effects is as follows. Note that we use the same parametrization used to define the scBFA model earlier, except that we define a new observation matrix $\boldsymbol{M}$ as a $N \times G$ matrix, where observations can either correspond to measured expressed levels $\boldsymbol{O}$, inferred binary detection pattern $\boldsymbol{B}$, or imputed read counts. Except for minor differences in parameterization, the GLM-based dimensionality reduction methods scBFA and ZINB-WaVE can be summarized in the following framework, where $g$ is the link function, $P$ is a probability measure, and $\mu$ is the expectation over the probability measure. In the case of ZINB-WaVE, $P$ is a zero-inflated negative binomial distribution. In the case of scBFA, $P$ is a Bernoulli distribution.

$$g\left(\mu_{ij}\right) = \left(\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{T}}\boldsymbol{\beta}_{\boldsymbol{j}} + \boldsymbol{z}_{\boldsymbol{i}}^{\boldsymbol{T}}\boldsymbol{a}_{\boldsymbol{j}} + u_i + \nu_j\right)$$

$$M_{ij} \sim P\left(\mu_{ij}\right)$$

We first identify the largest subset of cell types that are represented in all batches within a given dataset. Define $\boldsymbol{M}_{\text{sub}}$ as the submatrix of $N^{'}$ observations ($N^{'} < N$) corresponding to this subset of cell types and similarly define the submatrices $\boldsymbol{X}_{\text{sub}}$, $\boldsymbol{Z}_{\text{sub}}$, $\boldsymbol{A}_{\text{sub}}$ and $\boldsymbol{u}_{\text{sub}}$, where $i^{'} = 1, \ldots, N^{'}$. We ran each dimensionality reduction method once to obtain an estimate of $\hat{\boldsymbol{\beta}}$ by optimizing the likelihood of the following model:

$$\boldsymbol{M}_{\mathbf{sub}\left(i',j\right)} \sim P\left(g^{-1}\left(\boldsymbol{x}^{\boldsymbol{T}}_{\mathbf{sub}\left(i'\right)}\boldsymbol{\beta}_{\boldsymbol{j}} + \boldsymbol{z}^{\boldsymbol{T}}_{\mathbf{sub}\left(i'\right)}\boldsymbol{a}_{\boldsymbol{j}} + u_{\mathrm{sub}\left(i'\right)} + \nu_{j}\right)\right)$$

$\hat{\boldsymbol{\beta}}$ learns the variance induced by different batches only. Then, we use $\hat{\boldsymbol{\beta}}$ as a plug-in estimate of $\boldsymbol{\beta}$, and performed each dimensionality method on the full dataset to obtain estimates of all other parameters. Note since both ZINB-WaVE and scBFA regularize their coefficient matrix $\boldsymbol{\beta}$, $\boldsymbol{X}_{\mathrm{sub}}$ and $\boldsymbol{X}$ are both standardized. For PCA, scImpute, SAVER, sctransform, scrna2019, SCRAT, and chromVAR, we used a similar strategy to obtain $\hat{\boldsymbol{\beta}}$ by using linear regression to regress out $\boldsymbol{X}_{\mathrm{sub}}$ from the observation matrix corresponding to the largest subset of cell types represented in all batches. Then, we calculated the residuals $R_{ij} = M_{ij} - \boldsymbol{X}\boldsymbol{\beta}^{\boldsymbol{T}}$ on the full dataset with $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$ fixed and performed PCA on the residual matrix $\boldsymbol{R}$. For scVI, we were unable to modify the model framework to adjust for batch effects when they were confounded with cell types, as was the case in MEM-T, Pancreatic, and MGE. Therefore, we measured the scVI performance when we did not correct for batch effect, as well as when we performed naïve batch effect correction ignoring the confounding, and then reported the best performance for scVI.

Scasat handles batch effects through the removal of batch-specific loci. However, for datasets GSE96769 and GSE74310, the batch effect is confounded with cell types. Therefore, we ran Scasat without batch effect correction because batch-specific loci would be indistinguishable from cell type-specific loci. Destin and scABC cannot adjust batch effect on their own, and so we ran them without batch effect correction on these two datasets.

## Identification of marker genes

We evaluated the extent to which the inferred dimensions for each method recover known marker genes (Fig. 5). For each method, we first obtained the $K \times G$ 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 ranked the absolute value of each gene in each factor and calculated the area under the receiver-operator curve (AUROC) 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 loading matrices corresponding to the gene detection and gene count components, respectively, and therefore appears twice in Fig. 5. In ZINB-WaVE, $\pi_{ij}$ models whether a gene has been detected or not, and $\mu_{ij}$ models the mean for the read counts under negative binomial distribution. As in the previous section, we used the parameters $\{\hat{\boldsymbol{Z}}, \hat{\boldsymbol{A}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{A}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{u}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{u}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\nu}}_{\boldsymbol{\pi}}\}$ to replace the parameters $\{\hat{\boldsymbol{W}}, \hat{\boldsymbol{\alpha}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\alpha}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{\gamma}}_{\boldsymbol{\pi}}, \hat{\boldsymbol{\gamma}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\pi}}\}$ used y:

$$\begin{aligned} \mathrm{logit}\left(\pi_{ij}\right) &= \left(\boldsymbol{z}^{\boldsymbol{T}}_{\boldsymbol{\pi}(\boldsymbol{i})}\boldsymbol{a}_{\boldsymbol{\pi}(\boldsymbol{j})} + \boldsymbol{x}^{\boldsymbol{T}}_{\boldsymbol{i}}\boldsymbol{\beta}_{\boldsymbol{\pi}(\boldsymbol{j})} + u_{\pi(i)} + \nu_{\pi(j)}\right) \\ \log\left(\mu_{ij}\right) &= \left(\boldsymbol{z}^{\boldsymbol{T}}_{\boldsymbol{\mu}(\boldsymbol{i})}\boldsymbol{a}_{\boldsymbol{\mu}(\boldsymbol{j})} + \boldsymbol{x}^{\boldsymbol{T}}_{\boldsymbol{i}}\boldsymbol{\beta}_{\boldsymbol{\mu}(\boldsymbol{j})} + u_{\mu(i)} + \nu_{\mu(j)}\right) \end{aligned}$$

The loading matrix $\boldsymbol{a}_{\boldsymbol{\pi}}$ that models the gene detection component ($\pi$) is denoted as $ZINB\text{-}WaVE_{\mathrm{dropout}}$, and the loading matrix $\boldsymbol{a}_{\mu}$ that models gene counts is denoted $ZINB\text{-}WaVE_{\mathrm{mean}}$.

## Trajectory inference

To evaluate the performance gains of scBFA in the context of trajectory inference, we used a recently published platform, dynverse, for which "gold standard" scRNA-seq datasets were already obtained and preprocessed, and scripts and performance metrics have already been defined to evaluate trajectory inference [32]. Gold standard datasets refer to those datasets in which experimental (non-computational) methods were used to annotate a dataset with trajectory information such as cell type clusters (milestones) and connections between cell type clusters (milestone networks). From the 27 datasets available on June 12, 2019, that met this gold standard status and were real (not synthetic), we filtered out datasets that had less than 170 cells, yielding a total of 20 benchmark datasets. As with our previous experiments, we used the HVG selection criterion to identify the top 2000 varying genes for dimensionality reduction.

Our strategy for benchmarking trajectory inference was to identify an existing, top-performing trajectory inference method that also uses dimensionality reduction in its pipeline then replace that dimensionality reduction step with one of the methods we tested in our study. The dynverse paper identified Slingshot as a top performer [32]. To evaluate scBFA and the other methods, we substituted the PCA step of Slingshot with each dimensionality reduction method (scBFA, ZINB-WaVE, PCA, scImpute, SAVER, scrna2019, sctransform, scVI) and used dynverse to measure the performance of each modified version of Slingshot. The number of input latent dimensions was set to 10. Because the Slingshot implementation throws NA in cases where it is uncertain of the assignment of cells to a particular lineage, we removed two datasets from further evaluation because the number of NAs produced prevented calculation of the performance metrics (germline-human-both_guo.rds, mESC-differentiation_hayashi.rds). For each of the 18 benchmarks (Additional file 1: Table S8), we used dynverse to compute three performance metrics with respect to the experimentally gathered trajectory information: $\mathrm{F1}_{\mathrm{milestones}}$, $\mathrm{F1}_{\mathrm{branches}}$, and $\mathrm{NMSE}_{\mathrm{lm}}$. $\mathrm{F1}_{\mathrm{milestones}}$ measures the similarity between clustering membership of two trajectories. $\mathrm{F1}_{\mathrm{branches}}$ compares the similarity between the assignment of two branches. $\mathrm{NMSE}_{\mathrm{lm}}$ is a measurement of how well the position of a cell in the inferred trajectory predicts the position of the cell in the ground truth trajectory under linear regression. Larger values of $F1_{milestones}$, $F1_{branches}$, and $NMSE_{lm}$ correspond to better performance. We obtained $F1_{milestones}$, $F1_{branches}$, and $NMSE_{lm}$ via dynverse's calculate_mapping and calculate_position_predict functions within the dyneval package and converted raw values to ranks for Fig. 6. The wrapper function to obtain the results from Slingshot is adapted from the internal function https://github.com/dynverse/ti_slingshot/blob/master/package/R/ti_slingshot.R.

### Visualization

After we obtain the embedding matrix from every method, we use the *t*-distributed stochastic embedding [49] method to project the embedding matrix onto two dimensions for visualization as a scatterplot. In all visualizations, the number of factors used as input to t-SNE in each visualization is 10.

### Timing experiments

In the timing experiment (Additional file 1: Figure S27), we randomly subsampled 1k, 10k, 50k, and 100k cells from the 1.3 million 10× brain cell dataset from E18 mice and recorded the single-core execution time (in seconds) of all methods (PCA, ZINB-WaVE, scImpute, SAVER, sctransform, scrna2019, and Binary PCA) on the same machine. Due to the non-convex nature of ZINB-WaVE's objective function and different optimization scheme, we cannot strictly match the convergence criterion of ZINB-WaVE to scBFA. Therefore, we use the same number of iterations for each method that was used to generate the results in Fig. 1. Because scImpute requires specification of the number of cell clusters, we set the number of cell clusters to seven, similar to a previous study [50] that used seven as an underestimate of the true number of cell types.

## Additional files

**Additional file 1:** Contains supplementary figures and tables, **Figures S1–S29.**, **Tables S1–S8.** (DOCX 3090 kb)

**Additional file 2:** Review history. (DOCX 3950 kb)

### Acknowledgements

Not applicable.

### Review history

The review history is available as Additional file 2.

### Authors' contributions

RL and GQ conceived the study, analyzed and interpreted the data, and wrote the manuscript. RL wrote the code for scBFA. Both authors read and approved the final manuscript.

### Funding

This project has been made possible in part by grant number 2018-182633 from the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation. Funding was also provided by NSF CAREER award 1846559 to GQ.

### Availability of data and materials

The source code implementing scBFA is available on a Bioconductor at https://bioconductor.org/packages/devel/bioc/html/scBFA.html, as well as in a GitHub repository at https://github.com/quon-titative-biology/scBFA[51]. scBFA is released under Apache License 2.0.

All datasets analyzed in this study with accession numbers and references are included in the published article and Additional file 1. Benchmark datasets used for cell type identification in scRNA-seq are GSE48968 [52], GSE104157 [53], GSE100426 [54], GSE62270 [55], GSE106540 [56], GSE81076 [57], GSE100866 [48], GSE89232 [58], GSE123025 [59], GSE94383 [60], GSE100037 [61], GSE81682 [62], E-MTAB-2805 [63], and SRP073808 [64] (Additional file 1: Table S1). Benchmark datasets used for cell type identification in scATAC-seq are GSE96769 [45], GSE74310 [46], and GSE107816 [47]. Benchmark datasets used for trajectory inference in scRNA-seq are GSE59114 [65], E-MTAB-2805 [63], GSE60781 [66], GSE86146 [67], GSE70240 [68], GSE70243 [68], GSE70244 [68], GSE70236 [67], E-MTAB-3929 [69], GSE52529 [16], GSE74596 [70], GSE87375 [71], GSE99951 [72], GSE48968 [52], and GSE85066 [73] (Additional file 1: Table S8). Representative scRNA-seq datasets used for observational study in Additional file 1: Figure S1 are GSE101601 [74], GSE106707 [75], GSE110558 [76], GSE110692 [76], GSE119097 [77], GSE56638 [78], GSE72056 [79], GSE81682 [62], GSE85527 [80], GSE86977 [81], GSE95432 [82], GSE98816 [83], GSE95315 [84], GSE95752 [84], GSE76381 [85], GSE110679 [76], GSE99888 [86], GSE52529 [16], GSE60749 [87], GSE63818 [88], GSE71982 [89], GSE57872 [90], GSE102299, GSE48968 [52], GSE104157 [53], GSE100426 [54], GSE62270 [55], GSE106540 [56] (Additional file 1: Table S7).

### Ethics approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Competing interests

The authors declare that they have no competing interests.

### Author details

[1]Graduate Group in Biostatistics, University of California, Davis, Davis, CA, USA. [2]Genome Center, University of California, Davis, Davis, CA, USA. [3]Department of Molecular and Cellular Biology, University of California, Davis, Davis, CA, USA.

Received: 11 March 2019 Accepted: 28 August 2019

### References

1. Hwang B, Lee JH, Bang D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.* 2018;50:96.
2. Svensson V, Vento-Tormo R, Teichmann SA. Exponential scaling of single-cell RNA-seq in the past decade. *Nat Protoc.* 2018;13:599–604.
3. Hicks SC, Townes FW, Teng M, Irizarry RA. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics.* 2018;19:562–78.
4. Jaitin DA, et al. Massively parallel single cell RNA-Seq for marker-free decomposition of tissues into cell types. *Science.* 2014;343:776–9.
5. Vallejos CA, Risso D, Scialdone A, Dudoit S, Marioni JC. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nat. Methods.* 2017;14:565–71.
6. Ziegenhain, C. *et al.* Comparative analysis of single-cell RNA sequencing methods. *Mol. Cell* 65, 631-643.e4 (2017).
7. Dueck HR, et al. Assessing characteristics of RNA amplification methods for single cell RNA sequencing. *BMC Genomics.* 2016;17:966.
8. Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert J-P. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat Commun.* 2018;9.

9. Li WV, Li JJ. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat Commun*. 2018;9:997.
10. Pierson E, Yau C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*. 2015;16:241.
11. Ramsköld D, et al. Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nat. Biotechnol.* 2012;30:777–82.
12. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. *Nature Methods*. 2018;15:1053.
13. Huang M, et al. SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods*. 2018;15:539–42.
14. Hafemeister, C. & Satija, R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *bioRxiv* 576827 (2019). doi:10.1101/576827
15. Townes, F. W., Hicks, S. C., Aryee, M. J. & Irizarry, R. A. Feature selection and dimension reduction for single cell RNA-Seq based on a multinomial model. *bioRxiv* 574574 (2019). doi:10.1101/574574
16. Trapnell C, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotech*. 2014;32:381–6.
17. Haghverdi L, Buettner F, Theis FJ. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*. 2015;31:2989–98.
18. Giecold, G., Marco, E., Garcia, S. P., Trippa, L. & Yuan, G.-C. Robust lineage reconstruction from high-dimensional single-cell data. *Nucleic Acids Research* gkw452 (2016). doi:10.1093/nar/gkw452
19. 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*. 2017;14:414–6.
20. Ding J, Condon A, Shah SP. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat Commun*. 2018;9:2002.
21. Pollen, A. A. *et al.* Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nat. Biotechnol.* 32, 1053–1058 (2014).
22. Heimberg G, Bhatnagar R, El-Samad H, Thomson M. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *cels*. 2016;2:239–50.
23. Zhang, M. J., Ntranos, V. & Tse, D. One read per cell per gene is optimal for single-cell RNA-Seq. *bioRxiv* 389296 (2018). doi:10.1101/389296
24. Kim JK, et al. Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nat Commun*. 2015;6:8687.
25. Brennecke P, et al. Accounting for technical noise in single-cell RNA-seq experiments. *Nat. Methods*. 2013;10:1093–5.
26. Finak G, et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol*. 2015;16.
27. Svensson V, et al. Power analysis of single-cell RNA-sequencing experiments. *Nat. Methods*. 2017;14:381–7.
28. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*. 2014;15:550.
29. Zheng GXY, et al. Massively parallel digital transcriptional profiling of single cells. *Nat Commun*. 2017;8:14049.
30. Islam S, et al. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nat. Methods*. 2014;11:163–6.
31. Cannoodt R, Saelens W, Saeys Y. Computational methods for trajectory inference from single-cell transcriptomics. *Eur. J. Immunol.* 2016;46:2496–506.
32. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* 2019;37:547–54.
33. Street K, et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*. 2018;19.
34. Baker SM, Rogerson C, Hayes A, Sharrocks AD, Rattray M. Classifying cells with Scasat, a single-cell ATAC-seq analysis tool. *Nucleic Acids Res*. 2019;47:e10.
35. Urrutia E, Chen L, Zhou H, Jiang Y. Destin: toolkit for single-cell analysis of chromatin accessibility. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btz141.
36. Zamanighomi M, et al. Unsupervised clustering and epigenetic classification of single cells. *Nature Communications*. 2018;9:2410.
37. Schep AN, Wu B. Buenrostro, J. D. & Greenleaf, W. J. chromVAR: inferring transcription-factor-associated accessibility from single-cell epigenomic data. *Nat. Methods*. 2017;14:975–8.
38. Ji Z, Zhou W, Ji H. Single-cell regulome data analysis by SCRAT. *Bioinformatics*. 2017;33:2930–2.
39. Arzalluz-Luque Á, Devailly G, Mantsoki A, Joshi A. Delineating biological and technical variance in single cell expression data. *Int. J. Biochem. Cell Biol.* 2017;90:161–6.
40. Stegle O, Teichmann SA, Marioni JC. Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet*. 2015;16:133–45.
41. Kharchenko PV, Silberstein L, Scadden DT. Bayesian approach to single-cell differential expression analysis. *Nat. Methods*. 2014;11:740–2.
42. Stuart T, et al. Comprehensive integration of single-cell data. *Cell*. 2019. https://doi.org/10.1016/j.cell.2019.05.031.
43. Soneson C, Robinson MD. Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods*. 2018;15:255–61.
44. Clough E, Barrett T. The Gene Expression Omnibus database. *Methods Mol Biol*. 2016;1418:93–110.
45. Buenrostro, J. D. *et al.* Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. *Cell* 173, 1535-1548.e16 (2018).
46. Corces MR, et al. Lineage-specific and single-cell chromatin accessibility charts human hematopoiesis and leukemia evolution. *Nat. Genet.* 2016;48:1193–203.
47. Satpathy AT, et al. Transcript-indexed ATAC-seq for precision immune profiling. *Nat. Med.* 2018;24:580–90.
48. Stoeckius M, et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods*. 2017;14:865–8.
49. van der Maaten L, Hinton G. Visualizing data using t-SNE. *Journal of Machine Learning Research*. 2008;9:2579–605.
50. Bhaduri A, Nowakowski TJ, Pollen AA, Kriegstein AR. Identification of cell types in a mouse brain single-cell atlas using low sampling coverage. *BMC Biol*. 2018;16.
51. Li R. Quon G.scBFA R code. Zenodo. https://doi.org/10.5281/zenodo.3372766.
52. Shalek AK, et al. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature*. 2014;510:363–9.
53. Mayer C, et al. Developmental diversification of cortical inhibitory interneurons. *Nature*. 2018;555:457–62.
54. Mann, M. *et al.* Heterogeneous responses of hematopoietic stem cells to inflammatory stimuli are altered with age. *Cell Rep* 25, 2992-3005.e5 (2018).
55. Grün D, et al. Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*. 2015;525:251–5.
56. Patil VS, et al. Precursors of human CD4+ cytotoxic T lymphocytes identified by single-cell transcriptome analysis. *Sci Immunol*. 2018;3.
57. Grün D, et al. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell*. 2016;19:266–77.
58. Breton G, et al. Human dendritic cells (DCs) are derived from distinct circulating precursors that are precommitted to become CD1c+ or CD141+ DCs. *J. Exp. Med.* 2016;213:2861–70.
59. Li, Q. *et al.* Developmental heterogeneity of microglia and brain myeloid cells revealed by deep single-cell RNA sequencing. *Neuron* 101, 207-223.e10 (2019).
60. Lane, K. *et al.* Measuring signaling and RNA-Seq in the same cell links gene expression to dynamic patterns of NF-κB activation. *Cell Syst* 4, 458-469.e5 (2017).
61. Herman, J. S., Sagar, null & Grün, D. FateID infers cell fate bias in multipotent progenitors from single-cell RNA-seq data. *Nat. Methods* 15, 379–386 (2018).
62. Nestorowa S, et al. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood*. 2016;128:e20–31.
63. Buettner F, et al. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol.* 2015;33:155–60.
64. Koh PW, et al. An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. *Sci Data*. 2016;3:160109.
65. Kowalczyk MS, et al. Single-cell RNA-seq reveals changes in cell cycle and differentiation programs upon aging of hematopoietic stem cells. *Genome Res*. 2015;25:1860–72.
66. Schlitzer A, et al. Identification of cDC1- and cDC2-committed DC progenitors reveals early lineage priming at the common DC progenitor stage in the bone marrow. *Nat. Immunol.* 2015;16:718–28.
67. Li, L. *et al.* Single-cell RNA-Seq analysis maps development of human germline cells and gonadal niche interactions. *Cell Stem Cell* 20, 858-873.e4 (2017).
68. Olsson A, et al. Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. *Nature*. 2016;537:698–702.

69. Petropoulos S, et al. Single-cell RNA-Seq reveals lineage and X chromosome dynamics in human preimplantation embryos. *Cell*. 2016;165:1012–26.
70. Engel I, et al. Innate-like functions of natural killer T cell subsets result from highly divergent gene programs. *Nat. Immunol.* 2016;17:728–39.
71. Qiu, W.-L. *et al*. Deciphering pancreatic islet β cell and α cell maturation pathways and characteristic features at the single-cell level. *Cell Metab.* 25, 1194-1205.e4 (2017).
72. Sloan, S. A. *et al*. Human astrocyte maturation captured in 3D cerebral cortical spheroids derived from pluripotent stem cells. *Neuron* 95, 779-790. e6 (2017).
73. Loh KM, et al. Mapping the pairwise choices leading from pluripotency to human bone, heart, and other mesoderm cell types. *Cell*. 2016;166:451–67.
74. Hochgerner H, et al. STRT-seq-2i: dual-index 5′ single cell and nucleus RNA-seq on an addressable microwell array. *Sci Rep*. 2017;7:16327.
75. Muñoz-Manchado, A. B. *et al*. Diversity of interneurons in the dorsal striatum revealed by single-cell RNA sequencing and PatchSeq. *Cell Rep* 24, 2179-2190.e7 (2018).
76. Zemmour D, et al. Single-cell gene expression reveals a landscape of regulatory T cell phenotypes shaped by the TCR. *Nat. Immunol.* 2018;19: 291–301.
77. Tian, L. *et al*. SIS-seq, a molecular 'time machine', connects single cell fate with gene programs. *bioRxiv* 403113 (2018). doi:10.1101/403113
78. Dueck H, et al. Deep sequencing reveals cell-type-specific patterns of single-cell transcriptome variation. *Genome Biol.* 2015;16:122.
79. Tirosh I, et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science*. 2016;352:189–96.
80. Long, S. A. *et al*. Partial exhaustion of CD8 T cells and clinical response to teplizumab in new-onset type 1 diabetes. *Science Immunology* 1, eaai7793 (2016).
81. Yao Z, et al. A single-cell roadmap of lineage bifurcation in human ESC models of embryonic brain development. *Cell Stem Cell*. 2017;20:120–34.
82. Pal B, et al. Construction of developmental lineage relationships in the mouse mammary gland by single-cell RNA profiling. *Nat Commun*. 2017;8:1627.
83. Vanlandewijck M, et al. A molecular atlas of cell types and zonation in the brain vasculature. *Nature*. 2018;554:475–80.
84. Hochgerner H, Zeisel A, Lönnerberg P, Linnarsson S. Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing. *Nat. Neurosci.* 2018;21:290–9.
85. La Manno, G. *et al*. Molecular diversity of midbrain development in mouse, human, and stem cells. *Cell* 167, 566-580.e19 (2016).
86. Harris KD, et al. Classes and continua of hippocampal CA1 inhibitory neurons revealed by single-cell transcriptomics. *PLoS Biol.* 2018;16:e2006387.
87. Kumar RM, et al. Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature*. 2014;516:56–61.
88. Guo F, et al. The Transcriptome and DNA methylome landscapes of human primordial germ cells. *Cell*. 2015;161:1437–52.
89. Burns JC, Kelly MC, Hoa M, Morell RJ, Kelley MW. Single-cell RNA-Seq resolves cellular complexity in sensory organs from the neonatal inner ear. *Nat Commun*. 2015;6:8557.
90. Patel AP, et al. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*. 2014;344:1396–401.

## Publisher's Note