Model-based dimensionality reduction for single-cell RNA-seq using generalized bilinear models

Phillip B. Nicol[1,*] and Jeffrey W. Miller[1]

[1]Department of Biostatistics, Harvard University

April 24, 2023

Abstract

Dimensionality reduction is a critical step in the analysis of single-cell RNA-seq data. The standard approach is to apply a transformation to the count matrix, followed by principal components analysis. However, this approach can spuriously indicate heterogeneity where it does not exist and mask true heterogeneity where it does exist. An alternative approach is to directly model the counts, but existing model-based methods tend to be computationally intractable on large datasets and do not quantify uncertainty in the low-dimensional representation. To address these problems, we develop scGBM, a novel method for model-based dimensionality reduction of single-cell RNA-seq data. scGBM employs a scalable algorithm to fit a Poisson bilinear model to datasets with millions of cells and quantifies the uncertainty in each cell's latent position. Furthermore, scGBM leverages these uncertainties to assess the confidence associated with a given cell clustering. On real and simulated single-cell data, we find that scGBM produces low-dimensional embeddings that better capture relevant biological information while removing unwanted variation. scGBM is publicly available as an R package.

Introduction

Single-cell RNA sequencing (scRNA-seq) is a revolutionary technology that allows gene expression to be profiled at the level of individual cells (Saliba et al., 2014). This allows for the identification of novel cell types that play critical roles in biological processes. However, the increased resolution provided by scRNA-seq comes at the cost of introducing several statistical and computational challenges. One major challenge is the large size of scRNA-seq datasets, which often contain millions of cells (Cao et al., 2019); thus, new methods must be computationally scalable (Lähnemann et al., 2020). Another challenge is that the extreme sparsity and discreteness of scRNA-seq count data make traditional statistical models based on normal distributions inappropriate (Vallejos et al., 2017). A third major challenge is that cell-level measurements are noisy and contain relatively little information (compared to bulk sequencing), making it important to quantify the uncertainty in these measurements and propagate it to downstream analyses (Lähnemann et al., 2020).

Due to the large size of single-cell datasets, it is standard practice to use a dimensionality reduction technique such as principal components analysis (PCA) before clustering and other downstream analyses (Luecken and Theis, 2019). However, applying PCA directly to the count matrix can introduce undesirable technical artifacts, since PCA implicitly assumes that the entries of the count matrix are normally distributed (Miller and Carter, 2020). Thus, researchers typically apply a transformation to the count matrix prior to running PCA, in order to make the model assumptions more reasonable. However, it has been shown that commonly used transformations such as $\log(1+x)$ fail to adequately address this problem, and can still lead to substantial biases in the subsequent PCA results (Townes et al., 2019). Indeed, it seems unlikely that there exists any single transformation that satisfies the necessary requirements simultaneously (Van Eeuwijk, 1995).

*Corresponding author: phillipnicol@g.harvard.edu

Alternatively, some methods perform dimensionality reduction using a probabilistic model of the count data matrix. These methods can avoid the artifactual biases of simple transformations and, further, can provide principled uncertainty quantification for downstream analyses and visualization. The GLM-PCA method of Townes et al. (2019) models the entries of the count matrix using a Poisson or negative-binomial distribution and estimates latent factors in the log space, however, GLM-PCA suffers from slow runtime and convergence issues on single-cell datasets with millions of cells (Lause et al., 2021). Similarly, ZINB-WAVE (Risso et al., 2018) employs a similar model that assumes a zero-inflated negative-binomial distribution for the counts, but it has been noted that ZINB-WAVE can take days to run on datasets with large numbers of cells (Agostinis et al., 2022). Furthermore, several studies have shown that the distribution of UMI counts is not zero inflated (Svensson, 2020; Sarkar and Stephens, 2021; Townes et al., 2019).

In this paper, we introduce scGBM, a novel approach to dimensionality reduction for scRNA-seq data. Starting from the same underlying model as in GLM-PCA (Townes et al., 2019), we provide three key innovations. First, we develop a new estimation algorithm that is faster than existing approaches and scales up to datasets with millions of cells. Second, we quantify uncertainty in the low-dimensional embedding, enabling calibrated inference in downstream analyses. Third, we use these uncertainties to define a *cluster confidence index* (CCI) that measures the stability of each cluster and the relationships between clusters. On real and simulated data, we demonstrate examples where the current leading approaches are unable to capture true biological variability, while scGBM is successful in doing so.

Results

Limitations of current leading methods.

It is known that fundamental issues can result from using PCA on $\log(1+x)$ transformed scRNA-seq count data to perform dimensionality reduction (Townes et al., 2019; Booeshaghi and Pachter, 2021). Methods using count models have been developed to address these issues, however, we find that the current leading approaches still have significant limitations. One of the most popular methods is scTransform (Hafemeister and Satija, 2019), implemented in the widely used Seurat package (Satija et al., 2015). Let $Y \in \mathbb{R}^{I \times J}$ be the matrix of UMI counts, where I is the number of genes, J is the number of cells, and UMI stands for unique molecular identifier. The idea behind scTransform is to apply PCA to the matrix of Pearson residuals obtained by fitting negative-binomial GLMs to the count matrix Y. Specifically, for each row i, scTransform fits the following model to data $Y_{i1}, \ldots, Y_{iJ}$:

$$\begin{aligned} Y_{ij} &\sim \text{NegBinom}\left(\mu_{ij}, \alpha_i\right) \\ \log(\mu_{ij}) &= \beta_{0i} + \beta_{1i} \log(S_j) \end{aligned} \tag{1}$$

where $S_j = \sum_{i=1}^{I} Y_{ij}$ and $\text{NegBinom}(\mu, \alpha)$ is the negative-binomial distribution with mean μ and variance $\mu + \mu^2/\alpha$. Then, PCA is applied to the matrix $Z = [Z_{ij}] \in \mathbb{R}^{I \times J}$ where

$$Z_{ij} := \frac{Y_{ij} - \hat{\mu}_{ij}}{\sqrt{\hat{\mu}_{ij} + \hat{\mu}_{ij}^2/\hat{\alpha}_i}} \tag{2}$$

are the Pearson residuals and $\hat{\mu}_{ij}$ and $\hat{\alpha}_i$ are regularized versions of the estimated parameters for the model in Equation (1). We demonstrate potential shortcomings of scTransform using (i) a simulation with three cell types differentiated by single marker genes, and (ii) a simulation with two cell types differentiated by log-fold changes of equal magnitude across all genes.

Single marker genes simulation. In simulation (i), we generated data for $J = 1000$ cells from three cell types, where cell types A and B are each distinguished by overexpressing a single marker gene. Specifically, relative to cell type C, cell type A overexpresses gene 1 and cell type B overexpresses gene 2, while the remaining 998 genes have identical mean expression across the three cell types; see Figure 1a and Supplement A for full details of the simulation. We applied scTransform to this dataset and found that, in terms of PCs 1 and 2, scTransform does not reveal the distinction between these three cell types (Figure 1b,c). The reason why this occurs is that scTransform implicitly standardizes the scale of the genes by dividing by the standard deviation, so all genes end up getting similar weight in the PCA. Additionally,

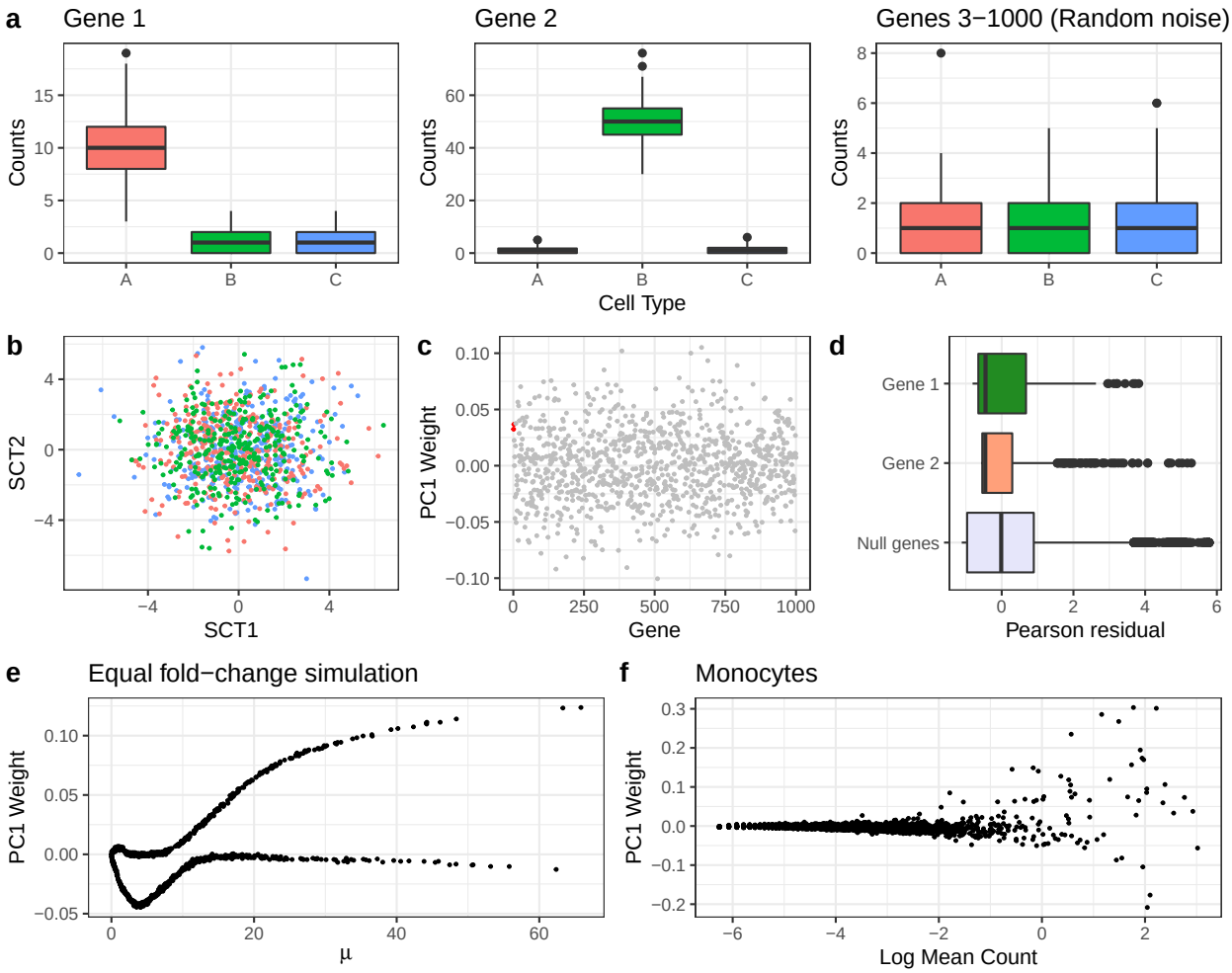

Figure 1: **Limitations of scTransform. a.** Single marker genes simulation: Boxplots of simulated counts for 1000 cells from 3 cell types. Gene 1 is overexpressed in cell type A and gene 2 is overexpressed in cell type B. The remaining genes are Poisson(1) random noise. **b.** The scores obtained from applying PCA to the Pearson residuals computed by scTransform. **c.** The corresponding weight of each gene in the first principal component. **d.** Boxplots of Pearson residuals for gene 1, gene 2, and the remaining "null" genes. **e.** Equal log-fold change simulation: Two cell types such that all genes have an absolute $\log_2$ fold change of 1 between the two cell types. The weight given to each gene in the first principal component depends on the baseline expression μ. **f.** Purified monocytes data: Applying scTransform to monocytes from 10X genomics, the genes with the largest baseline mean (in terms of log mean expression) tend to have the largest PC weights.

the overdispersion parameter α_i captures some of the latent variability in the first two genes, further reducing the signal. This is illustrated by Figure 1d, which shows that the variance of the Pearson residuals from the 998 noise genes is comparable to (and, in fact, larger than) that of the two marker genes. Thus, the normalization procedure used by scTransform upweights the noise genes and downweights the signal genes.

Equal log-fold change simulation. In simulation (ii), we generated $J = 1000$ cells from two equally sized groups. In the first group, we randomly generated the mean of each gene $i = 1, \ldots, I$ by sampling $\mu_i \sim \text{Exponential}(0.1)$ independently. In the second group, the mean of gene i was randomly set to either $2\mu_i$ or $\mu_i/2$, with probability $1/2$, independently. In other words, all genes are differentially expressed by a multiplicative factor (fold change) of 2 between the two groups of cells. Each count was generated as independent Poisson with the corresponding mean. Applying scTransform (Figure 1e), we see that the PC1 weights are strongly dependent on the base mean μ_i. This is undesirable because, in real data, some genes tend to have higher (or lower) measured expression simply because they are more (or less) abundant due to baseline cell activity or due to technical gene-specific effects unrelated to biology. Moreover, these weights are commonly used as a measure of a gene's influence or importance on the corresponding component (Soumillon et al., 2014; Petropoulos et al., 2016; Roden et al., 2006).

The reason why the peculiar dependence in Figure 1e occurs is that scTransform decomposes latent effects additively in the mean rather than in the log of the mean. Although scTransform adjusts for gene-specific and cell-specific intercepts in log space via $\log(\mu_{ij}) = \beta_{0i} + \beta_{1i} \log(S_j)$, the latent effects are decomposed in linear space because applying PCA to the Pearson residuals Z is equivalent to finding an approximate low rank factorization $U\Sigma V^T \approx Z$ where $U \in \mathbb{R}^{I \times M}$, $V \in \mathbb{R}^{J \times M}$, and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_M)$, that is,

$$\sum_{m=1}^{M} \sigma_m u_{im} v_{jm} \approx \frac{Y_{ij} - \hat{\mu}_{ij}}{\sqrt{\hat{\mu}_{ij} + \hat{\mu}_{ij}^2/\hat{\alpha}_i}}. \tag{3}$$

Purified monocytes data. To illustrate this issue on real data, we considered a dataset of $J = 2{,}612$ purified monocytes downloaded from the 10X genomics website (www.10xgenomics.com). Since these data come from a single cell type, we expect minimal true latent variation. Since the true mean is unknown, we computed the log mean count for each gene and plotted it against the PC1 weights obtained from scTransform (Figure 1f). Although the weights are more variable than in the simulated example, we again observe that the genes with the largest baseline mean expression tend to have the largest weights.

scGBM fits a Poisson bilinear model to the count matrix.

Our proposed method, scGBM, addresses these issues by using a Poisson bilinear model. scGBM models the entries of the UMI count matrix $Y \in \mathbb{R}^{I \times J}$ as

$$\begin{aligned} Y_{ij} &\sim \text{Poisson}(\mu_{ij}) \\ \log(\mu_{ij}) &= \alpha_i + \beta_j + \sum_{m=1}^{M} \sigma_m u_{im} v_{jm} \end{aligned} \tag{4}$$

for $i = 1, \ldots, I$ and $j = 1, \ldots, J$, where α_i is a gene-specific intercept, β_j is a cell-specific intercept, $U := [u_{im}] \in \mathbb{R}^{I \times M}$ is a matrix of latent factor weights, $V := [v_{jm}] \in \mathbb{R}^{J \times M}$ is a matrix of latent factor scores, and $\sigma_1, \ldots, \sigma_M > 0$ are scaling factors. For intuition, this model can roughly be thought of as applying PCA inside the link function of a generalized linear model (GLM). As in PCA, we constrain the columns of U to be orthonormal and order the columns such that $\sigma_1 > \cdots > \sigma_M > 0$, so the first factor captures the greatest amount of latent structure. Identifiability constraints are provided in the Methods section. Following Choulakian (1996), we refer to the model in Equation (4) as a *generalized bilinear model* (GBM), but other authors use different names such as GLM-PCA (Townes et al., 2019) or GAMMI (Van Eeuwijk, 1995). We refer to Miller and Carter (2020) for an extensive discussion of the literature on GBMs. By default, we fix the cell-specific offsets to $\beta_j = \log(\sum_{i=1}^{I} Y_{ij})$, but we also provide the option to estimate β_j.

A GBM that only includes intercepts and latent factors, as in Equation (4), is likely to be sufficient for most single-cell studies. However, in some cases, it is beneficial to control for known batches such as groups of samples sequenced in different locations or on different dates. To achieve this, scGBM can estimate a

Dataset	# of cells	scGBM-proj	scGBM-full	GLM-PCA (Fisher)	GLM-PCA (AvaGrad)	GLM-PCA (SGD)
10X immune cells	3,994	0.14 min	1.68 min	55.72 min	1.58 min	1.72 min
COVID-19 Atlas	44,721	1.89 min	22.77 min	468.24 min	19.52 min	20.66 min
10X mouse brain	1,308,421	57.67 min	782.40 min	9878.20 min*	811.09 min	505.82 min

Table 1: Runtime comparison of scGBM and GLM-PCA on three datasets: 10X immune cells (Zheng et al., 2017, downloaded via the *DuoClustering2018* package from Duò et al., 2018), COVID-19 Atlas (Wilk et al., 2020), and 10X mouse brain (Lun and Morgan, 2020). $I = 1000$ variable genes were selected using the Seurat package (Satija et al., 2015). Each method was run for a maximum of 100 iterations. scGBM-full is using IRSVD on the full data, and scGBM-proj is using IRSVD with the projection method, with 64 cores allocated. For scGBM-proj, we used subsamples of sizes 400, 4,000, and 100,000 on the three datasets, respectively. For GLM-PCA (SGD) the minibatch size was set to be equal to the subsample size used by scGBM-proj. For GLM-PCA, the learning rate (for Avagrad, SGD) and the penalty (for Fisher scoring) was set sufficiently small (or large) to prevent divergence. Dimensionality of $M = 20$ was used for all algorithms. *The runtime for GLM-PCA (Fisher) on 10X mouse brain was estimated by extrapolating the time taken for 10 iterations.

batch-specific intercept for each gene; see Methods for details. More generally, the model can be augmented to include arbitrary sample covariates and gene covariates, as well as interactions between these (Miller and Carter, 2020). However, in this paper, we only consider binary sample covariates since other types of covariates are rarely adjusted out during this stage of single-cell analyses.

In contrast to many bulk RNA-seq models, our scGBM model assumes a Poisson outcome and thus does not allow for overdispersion. In fact, there is increasing evidence that the technical sampling distribution of UMI counts is close to Poisson and that any additional dispersion is due to biological variability such as heterogeneous cell types or cell states (Sarkar and Stephens, 2021; Lause et al., 2021). Since the goal of scGBM is to remove technical variability while preserving all biological variability for downstream analyses, it is natural to use a Poisson outcome distribution.

Fast estimation using iteratively reweighted singular value decomposition.

While GBMs solve the fundamental issues exhibited by the log+PCA and scTransform approaches, existing methods for fitting GBMs are not scalable to the large datasets encountered in scRNA-seq. To address this limitation, we propose a new approach to fitting the Poisson GBM that combines iteratively reweighted least squares (IRLS) with singular value decomposition (SVD). This pairing is natural since IRLS is the standard way to fit GLMs and SVD is the standard way to perform PCA. In particular, we show that the latent factors U and V can be estimated by finding a low-rank approximation to a matrix where each entry is weighted according to its variance under the Poisson model; see Methods for details.

There are several advantages of the proposed algorithm, which we call *iteratively reweighted singular value decomposition* (IRSVD). First, it is asymptotically faster than Fisher scoring, the technique used by Miller and Carter (2020) and Townes et al. (2019); see Methods for the computational complexity of each iteration. Further, IRSVD leverages special properties of Poisson GLMs to obtain vectorized updates for the intercepts (that is, the entries of α and β can be updated simultanoeusly), which greatly reduces the runtime in practice. Finally, the identifiability constraints (such as orthogonality) are preserved at every iteration of the algorithm.

We compared our IRSVD algorithm (referred to here as scGBM-full) to three algorithms provided by the GLM-PCA R package as implemented by Townes and Street (2020): Fisher scoring, Avagrad (Savarese et al., 2021), and stochastic gradient descent (SGD). Table 1 compares the runtime (wall clock time) on three real single-cell datasets of varying sizes. We found that scGBM-full was approximately 20 times faster than GLM-PCA (Fisher), demonstrating the computational improvement of our approach. GLM-PCA (Avagrad) took approximately the same time as scGBM-full, while GLM-PCA (SGD) was similar or somewhat faster, however, we found that GLM-PCA (Avagrad) and GLM-PCA (SGD) converge to significantly less accurate solutions than scGBM-full on simulated data. Specifically, in simulations, we evaluated the accuracy of each method's estimates of U, ΣV, and $U\Sigma V^T$, and found that scGBM-full consistently outperformed the others (Figure 2a,b). Another limitation of the gradient-based methods implemented in GLM-PCA is that they are

very sensitive to the learning rate. Table S1 shows an example where changing the learning rate by just 0.01 leads to divergence. This significantly increases the computational burden in practice, since the algorithm must search for a suitable learning rate.

Finally, we compared the convergence rates of the various algorithms by plotting the log-likelihood $\sum_{i,j}\left(Y_{ij}\log(\mu_{ij})-\mu_{ij}\right)$ (additive constants removed) against the wall clock time (that is, the elapsed time since the start of the algorithm). We considered two simulated datasets, one with low latent variability (LLV) and one with high latent variability (HLV) (Figure 2c,d); see Methods for simulation details. We observe that the initialization used by scGBM-full has significantly higher log-likelihood than the one used by GLM-PCA. Figure S5 shows the log-likelihood versus runtime for several of the real datasets in Table 1.

Scaling up by fitting on a subset of samples and projecting.

Even with the improvement due to IRSVD, the computation time scales linearly with the number of samples, making it burdensome on datasets with millions of cells. Thus, to further improve scalability, we propose using a combination of subsampling and projection. Specifically, we begin by randomly selecting a subset of cells and estimating the U matrix using only the data from these cells. Then, holding U fixed to this estimate, column j of the Poisson bilinear model in Equation (4) is simply a GLM with covariate matrix U, which can be fit using standard techniques (see Methods); we call this the "projection method". The observation that fixing U or V yields a standard GLM has been used in previous RNA-seq techniques (Leek and Storey, 2007; Sun et al., 2012; Risso et al., 2014). Importantly, the GLMs can be fit in parallel since each column is processed independently. This also reduces memory requirements since the count matrix Y can remain stored in sparse form and only needs to be loaded into memory one column at a time, making it suitable for on-disk processing tools like *DelayedArray* (Pagès, 2020).

In Table 1, we see that the projection method (scGBM-proj) leads to significantly reduced runtimes compared to scGBM-full. Figure 2c displays the log-likelihood attained by scGBM-proj as a function of the number of iterations used to estimate U, on simulated data. In the case of high latent variation (HLV), scGBM-proj attains a higher log-likelihood in less wall clock time than the scGBM-full algorithm. Similar results were observed on real data (Figure S5).

To assess the accuracy of the projection method on real data, we considered the 10X immune cells and COVID-19 Atlas dataset from Table 1. We first applied scGBM-full to the whole dataset, obtaining $\hat{V}_{\text{full}} \in \mathbb{R}^{J\times M}$. We then used the projection method (scGBM-proj) with various subset sizes to obtain $\hat{V}_{\text{proj}} \in \mathbb{R}^{J\times M}$. We quantified the performance of the projection method by computing the absolute value of the correlation between columns of $\hat{V}_{\text{full}}$ and $\hat{V}_{\text{proj}}$ (Figure 3). We found that the first column of V (that is, the one corresponding to the largest scaling factor, σ_1) can be reliably estimated even using a small fraction of the cells to estimate U. These results suggest that using the projection method with $\approx$ 10-15% of the data is sufficient for capturing the most dominant sources of biological variation. We obtained similar results when testing the projection method on simulations with known ground truth (see Figure S1).

scGBM overcomes limitations of current leading methods.

Next, we assess the quality of scGBM results compared to two commonly used approaches, finding that it overcomes the limitations observed earlier. The first approach is to apply PCA to $\log(1+\text{CPM})$, where $\text{CPM} \in \mathbb{R}^{I\times J}$ is the matrix of counts per million and log is applied entry-wise; we refer to this as "log+PCA." The second is scTransform (Hafemeister and Satija, 2019), which applies PCA to a matrix of Pearson residuals, as described above. Both methods are implemented in the *Seurat* package (Satija et al., 2015).

One might think that scTransform and scGBM would yield very similar results since, as stated by Townes et al. (2019), applying PCA to the Pearson residuals can be viewed as an approximation to fitting the Poisson bilinear model used by scGBM. This approximation is justifiable when the scale of the latent variation is small (see Proposition 1), however, when the scale of the latent variation is large, scTransform and scGBM can produce very different results.

Single marker genes simulation. To illustrate, first consider the single marker genes simulation from Figure 1. Figure 4a plots the factor scores from the three methods, showing that scGBM is the only method able to distinguish the three simulated cell types. The factor 1 weights (Figure 4b) show that log+PCA and scTransform are unable to separate the two signal genes (1 and 2) from the random noise. The log+PCA

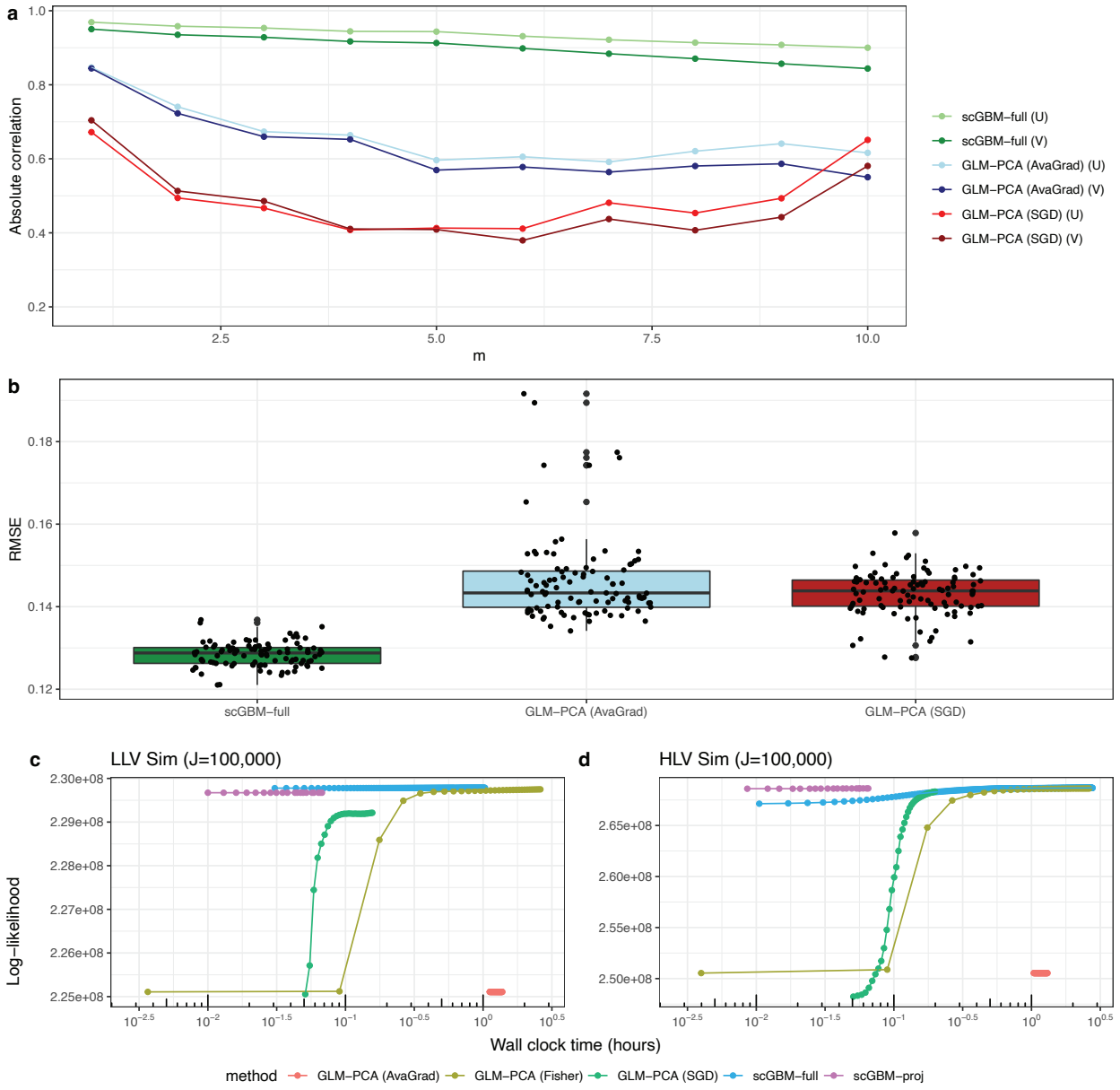

Figure 2: **Comparison of scGBM-full (IRSVD) and GLM-PCA on simulated data. a.** Absolute value of the correlation between estimated columns of U and V and the ground truth, across 100 simulated datasets with $I = 1000$, $J = 2000$, and $M = 10$. The points are the median across the datasets. **b.** Root mean squared error (RMSE) between the ground truth latent effects $X = U\Sigma V^T$ and the estimate $\hat{X}$ from each method, for 100 simulated datasets. The Fisher scoring algorithm implemented in GLM-PCA was not compared since it had a significantly longer runtime. **c.** Log-likelihood versus wall clock time on a low latent variability (LLV) simulated dataset with $I = 1000$ and $J = 10^5$. **d.** The same plot for a high latent variability (HLV) simulated dataset. (See Technical details of simulations.)

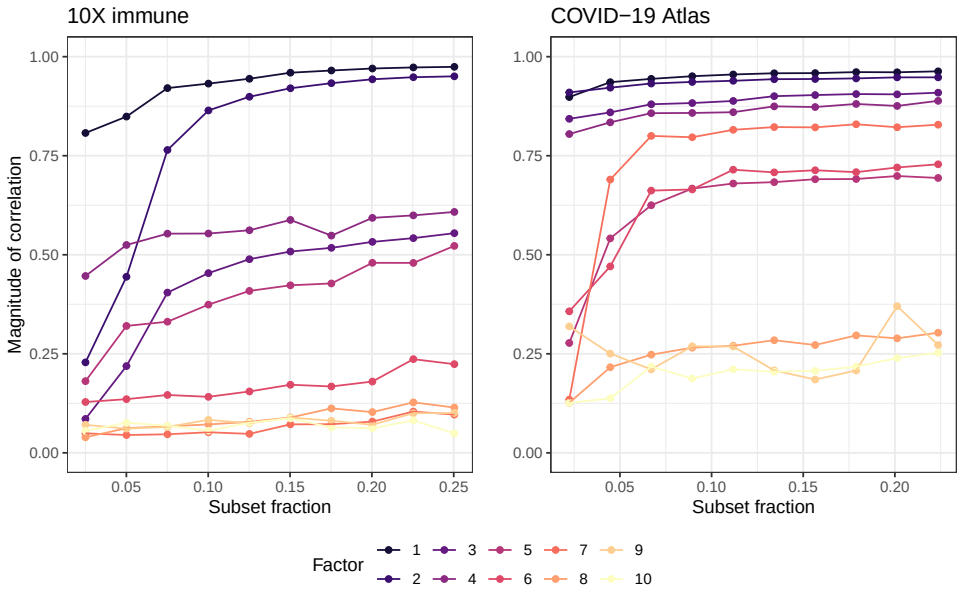

Figure 3: Testing the accuracy of the projection method. We applied scGBM-full to the 10X immune cells dataset (J = 3,994) from the *DuoClustering2018* package (Duò et al., 2018) to estimate V. Then, using subsets of various sizes, we used scGBM-proj to estimate V and computed the correlation between the columns of the two estimates of V (full and subset). The points represent the median absolute correlation across 100 runs of scGBM-proj. As the subset fraction increases, scGBM-proj agrees more closely with scGBM-full, with 10-15% being sufficient to capture the dominant sources of variation. We repeated this analysis for the COVID-19 Atlas dataset (Wilk et al., 2020).

approach fails to distinguish these cell types for the same reason that scTransform fails, namely, that it standardizes all genes in such a way that the signal genes are drowned out by the noise genes.

Equal log-fold change simulation. Next, consider the equal log-fold change simulation (Figure 1e) where there are two cell types and all genes are differentially expressed with a $\log_2$ fold change of ± 1. Unlike scTransform, scGBM does not assign higher weights to genes simply because they have a higher baseline expression (Figure 4c). Indeed, the scGBM factor weights are not correlated with the base mean μ.

Purified monocytes data. As a real data example, we revisit the purified monocytes data (Figure 1f). Here, we observed that the scGBM weights have a higher variance for genes with low mean. This is due to the inherent large uncertainty in $\log(\mu_{ij})$ when μ_{ij} is close to zero, leading to unstable estimates of u_{im} (Figure 4d, raw). To address this, we take advantage of the fact that scGBM is model-based, which enables us to quantify uncertainty in the parameter estimates. Using the classical method of inverting the Fisher information matrix (Methods), we approximate the standard error corresponding to each entry of U and use the Adaptive Shrinkage (*ash*) method of Stephens (2017) to perform *post hoc* shrinkage towards 0. Since *ash* shrinks genes with larger uncertainties more strongly (Figure 4d, stabilized), this effectively stabilizes the scGBM weights such that low mean genes are not given overly large weights simply due to noise.

Semi-simulated hybrid T cells. As a final example, we used real data to construct a semi-simulated dataset with cells that lie on a "gradient" between two different cell types. Beginning with naive T cells and memory T cells from the *DuoClustering2018* package (Duò et al., 2018), we simulated hybrid cells by combining a fraction of the genes from each cell type. Specifically, we used the following procedure to generate a dataset with $J = 5000$ hybrid cells, each of which is generated as follows:

1. Randomly select one memory T cell and one naive T cell.
2. Sample $I' \sim \text{Unif}\{1, \ldots, I\}$ and randomly choose a subset of I' genes.
3. Construct a simulated cell such that the counts for the I' randomly chosen genes are equal to those of the memory T cell, and the remaining $I - I'$ genes counts are equal to those of the naive T cell.

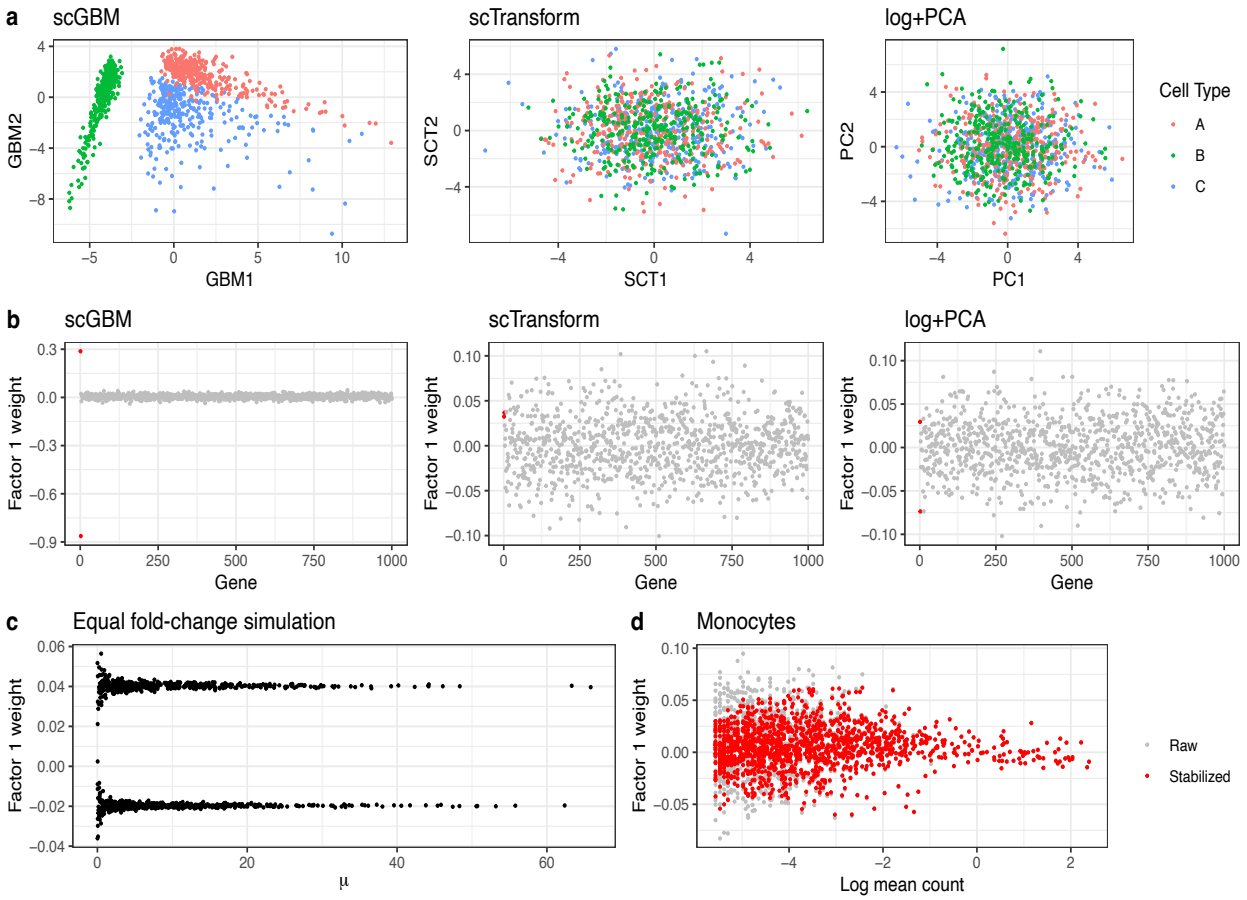

Figure 4: Results for scGBM, scTransform, and log+PCA on the simulations from Figure 1. **a.** Single marker genes simulation: The first two columns of V for scGBM compared to the first two PC scores from the other two methods. **b.** The weight of each gene in the first factor. The first two genes (colored red) are the only ones that are differentially expressed in the simulation. **c.** Equal log-fold change simulation: Factor 1 weights for scGBM using the simulated data from Figure 1c. **d.** Purified monocytes data: Factor 1 weights for scGBM using the purified monocytes from Figure 1f. The red dots are stabilized weight estimates using adaptive shrinkage based on sampling variability (Methods).

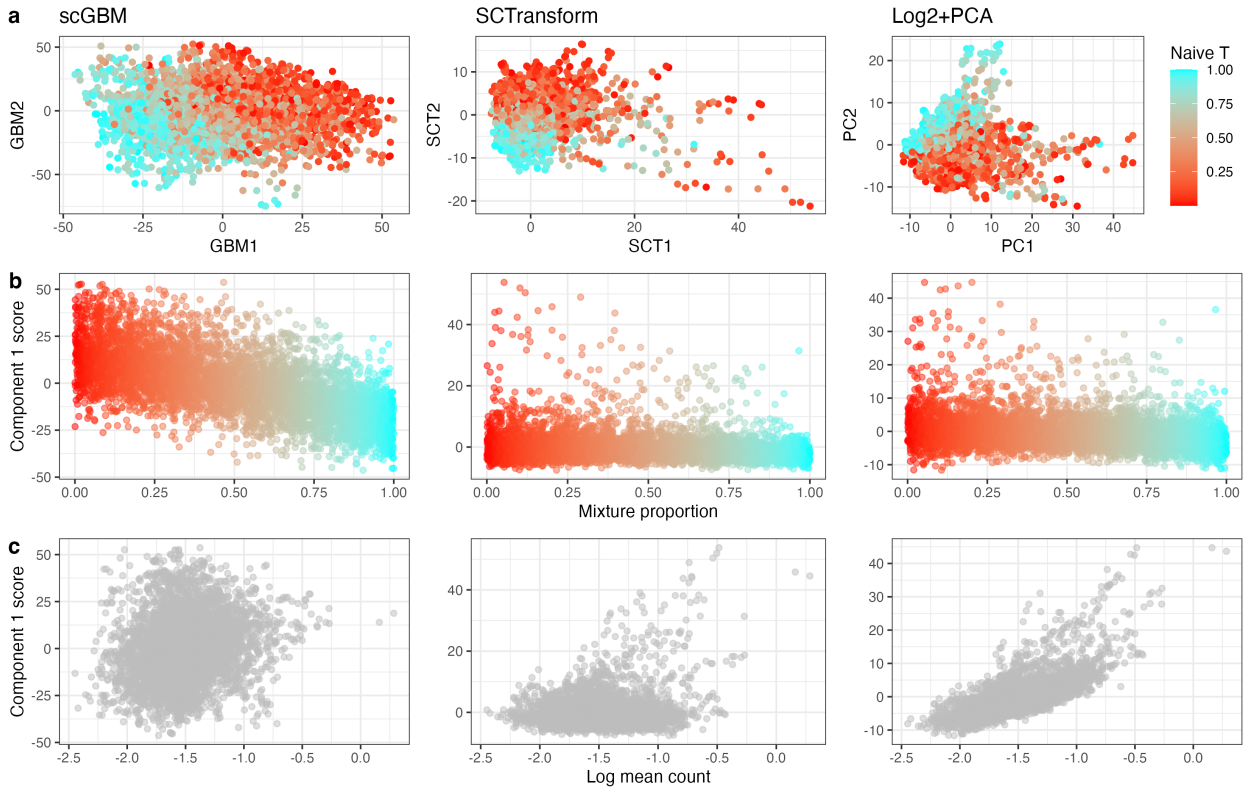

Figure 5: Comparison of scGBM, scTransform, and log+PCA on semi-simulated data in which cells are on a biological gradient between naive T cells and memory T cells. **a.** Scatterplots of the top 2 factor scores for the three methods. **b.** Factor 1 score versus the mixture proportion, that is, the fraction of genes from the naive T cell ($1 - I'/I$). **c.** Factor 1 score versus the log of the mean UMI count per cell.

In this dataset $I = 6088$ is the number of genes that has non-zero expression in at least 10 cells (in the original data). Figure 5(a,b) shows that scGBM is better able to recover the true biological gradient compared to scTransform and log+PCA, for which the true gradient is obscured by over-sensitivity to outliers. In these competing methods, the first factor score for each cell is strongly correlated with the mean count for that cell ($J^{-1}S_j$), suggesting that these methods are not adequately adjusting for sequencing depth (Figure 5c).

Using scGBM uncertainty quantification to assess cluster stability.

Clustering typically occurs downstream of dimensionality reduction, as a *post hoc* analysis step (Kiselev et al., 2019). Uncertainty in the estimated low-dimensional representation is likely to influence clustering results and other downstream analyses, but the effect of this uncertainty has not been thoroughly investigated in previous work. For example, one might wonder whether different clusters would be identified if the same set of cells was re-sequenced and the same analysis was performed on the resulting new count matrix.

Since scGBM performs dimensionality reduction using a statistical model, uncertainty in the low-dimensional scores V can be quantified using the classical method of inverting the Fisher information matrix (see Methods for details). The results in this section are obtained using scGBM-full, but standard errors can be obtained using scGBM-proj as well. Visually, uncertainty in the low-dimensional representation can be displayed by drawing an ellipse around each point, with the dimensions of the ellipse equal to the estimated standard errors. To demonstrate, we simulated a count matrix consisting of random Poisson noise, $Y_{ij} \sim \text{Poisson}(1)$ i.i.d. with $I = 1000$ and $J = 5000$, and used scGBM to obtain estimated scores $\hat{V}$. Then we applied the Leiden algorithm (Traag et al., 2019) to $\hat{V}$ to identify clusters of cells, since this is the default clustering algorithm used in Seurat (Satija et al., 2015) (`FindAllClusters()` with default resolution equal to 0.8). The algorithm identified 9 clusters even though the cells were simulated to be homogeneous

(Figure 6a). The fact that standard single-cell clustering algorithms produce too many clusters on null data has also been noted in other studies (Morelli et al., 2021; Grabski et al., 2022). We then used scGBM to estimate standard errors for $\hat{V}$ and visualized them by drawing an ellipse around each point (Figure 6b). This visualization indicates that there is relatively low confidence in the low-dimensional representation of the cells.

To provide a quantitative measure of the uncertainty in each cluster, we introduce the *cluster confidence index* (CCI). Given estimates $\hat{v}_{jm}$ and corresponding standard errors $\text{se}(\hat{v}_{jm})$, we randomly generate perturbed values $\tilde{v}_{jm} \sim \mathcal{N}(\hat{v}_{jm}, \text{se}(\hat{v}_{jm})^2)$, perform clustering on the rows of $\tilde{V} = [\tilde{v}_{jm}]$, and compute the fraction of pairs of cells that are still in the same cluster. Then, for each original cluster, the CCI is defined as the median of this fraction over many repetitions (see Methods for details).

To illustrate, we computed the CCIs for the random Poisson noise dataset, using the estimates $\hat{v}_{jm}$ and standard errors $\text{se}(\hat{v}_{jm})$ provided by scGBM, and using the Leiden clustering algorithm with the same resolution parameter as before. The CCIs are around 10-20%, correctly indicating that one should have little confidence in this clustering (Figure 6c). Meanwhile, the 10X immune cell data (Table 1) provides an example with real biological groups consisting of four purified cell types (Figure S3a). Treating the cell types as the clusters to assess, the CCIs (using scGBM and k-means with $k = 4$ for resampling) are typically above 80%, correctly indicating that these cell type annotations are stable under sampling variability (Figure S3b).

To quantify the overlap between clusters in a way that accounts for model-based uncertainty, we define the inter-cluster confidence index (inter-CCI) as follows: out of all pairs of points j and j' that were originally in clusters k and k', respectively, the inter-CCI is the median of the fraction that are in the same cluster after resampling (see Methods for details). Note that the inter-CCI between a cluster and itself coincides with the CCI of that cluster. To demonstrate the inter-CCI metric, we apply it to a dataset of 100,064 cells from the tumor microenvironment of 26 breast cancer patients (Wu et al., 2021); see Figure 7a. Since Wu et al. (2021) used the standard Seurat pipeline (Satija et al., 2015) to cluster cells and used Garnett (Pliner et al., 2019) for annotation, we computed the inter-CCIs for this Seurat-based clustering; see heatmap in Figure 7b. The blocks along the diagonal indicate groups of cell types with substantial overlap. These blocks appear to correspond to biologically relevant classes of cells such as lymphoid (T cells, B cells, NK cells), myeloid (monocytes and DCs), and endothelial cells. Similarly, Figure S3c shows inter-CCIs for the 10X immune cells. The inter-CCIs indicate a slight overlap between the regulatory T cell and naive cytotoxic cell clusters, but very little to no overlap between any other pair of clusters.

Discussion

The Poisson bilinear model has significant advantages compared to the commonly used approach of applying PCA to transformed counts. Foremost, it more accurately recovers true latent variation — revealing heterogeneity when it exists, and correctly showing no heterogeneity when there is none. Additionally, it enables model-based uncertainty quantification of both the low-dimensional representation and downstream inferences.

It is important to emphasize that scGBM is not designed to be a direct competitor of popular visualization methods such as t-SNE or UMAP (Van der Maaten and Hinton, 2008; McInnes et al., 2018), which are typically applied to PCA scores and are thus downstream of the initial dimensionality reduction. Rather, we are advocating for the use of scGBM instead of PCA for dimensionality reduction in single-cell RNA-seq analysis. One could then apply UMAP to the matrix of GBM scores V, but typically it appears that the first two columns of V provide adequate visualizations of biological variability without additional processing.

In future work, there are several interesting directions for improving upon our current approach. Although our scGBM algorithm is faster than GLM-PCA, it is still much slower than PCA — especially because commonly used transformations preserve sparsity, which allows for the use of more efficient SVD algorithms (Baglama and Reichel, 2005). This limitation could potentially be addressed by modifying our IRSVD algorithm to avoid applying the SVD to a dense matrix. A second limitation of scGBM is that, unlike PCA, the latent factors depend on the choice of M. Although there are heuristics for choosing the number of latent factors M (Methods), it remains an open problem to find a principled choice for M. A final area for future research would be to induce sparsity in the weights U, to improve biological interpretability. Sparsity in U would mean that only a few genes have non-zero weights, making it easier to connect the factors to known

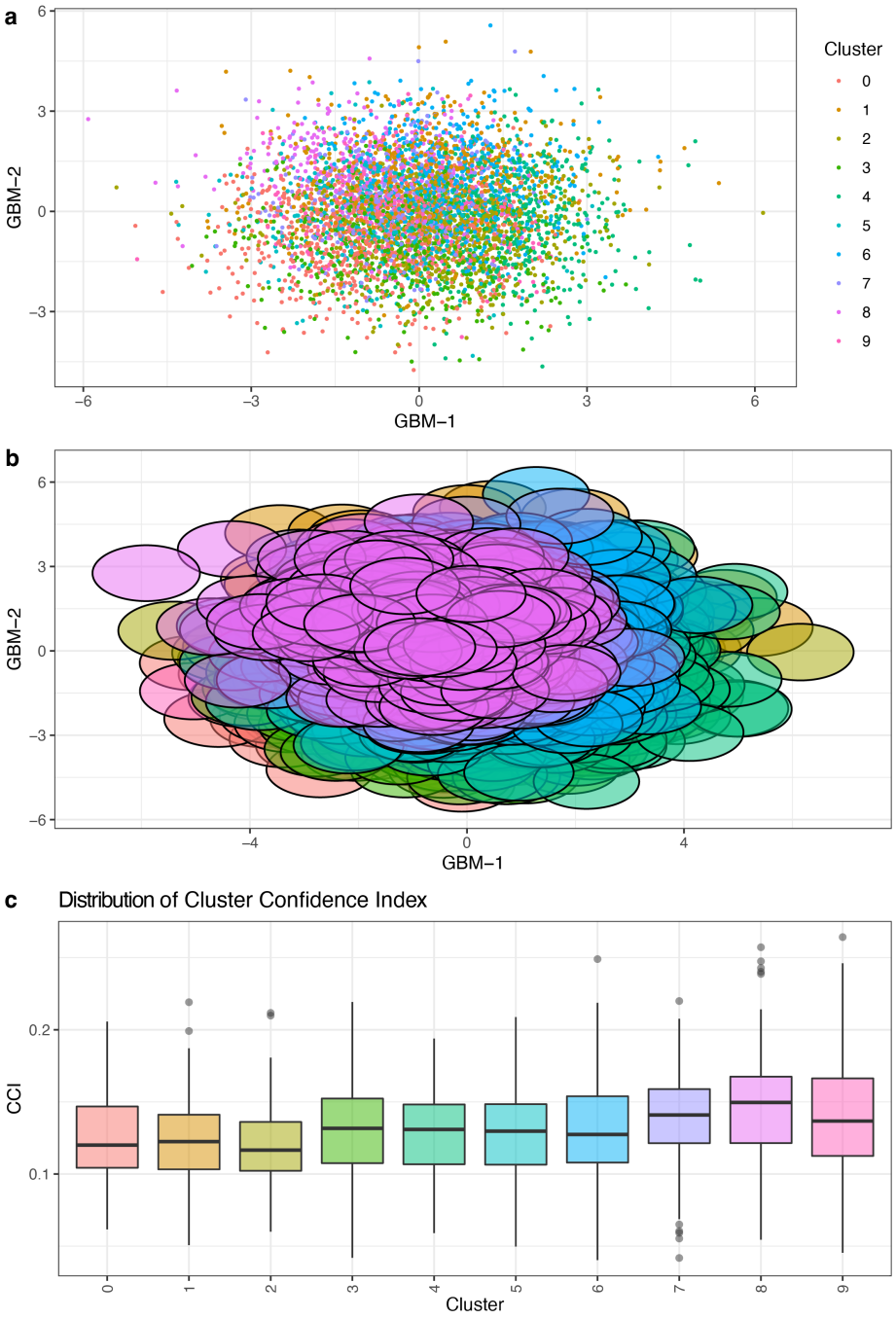

Figure 6: **a.** The first two columns of $\hat{V}$ estimated from random Poisson noise ($Y_{ij} \sim \text{Poisson}(1)$). The Leiden algorithm (Traag et al., 2019) with resolution 0.8 detects 9 spurious clusters. **b.** Uncertainty in the low-dimensional representation can be visualized by drawing an ellipsoid around each point, with axis lengths equal to the estimated standard errors. **c.** The cluster confidence indices (CCIs) for all 9 clusters are low, meaning that these clusters are unstable under resampling, as expected.

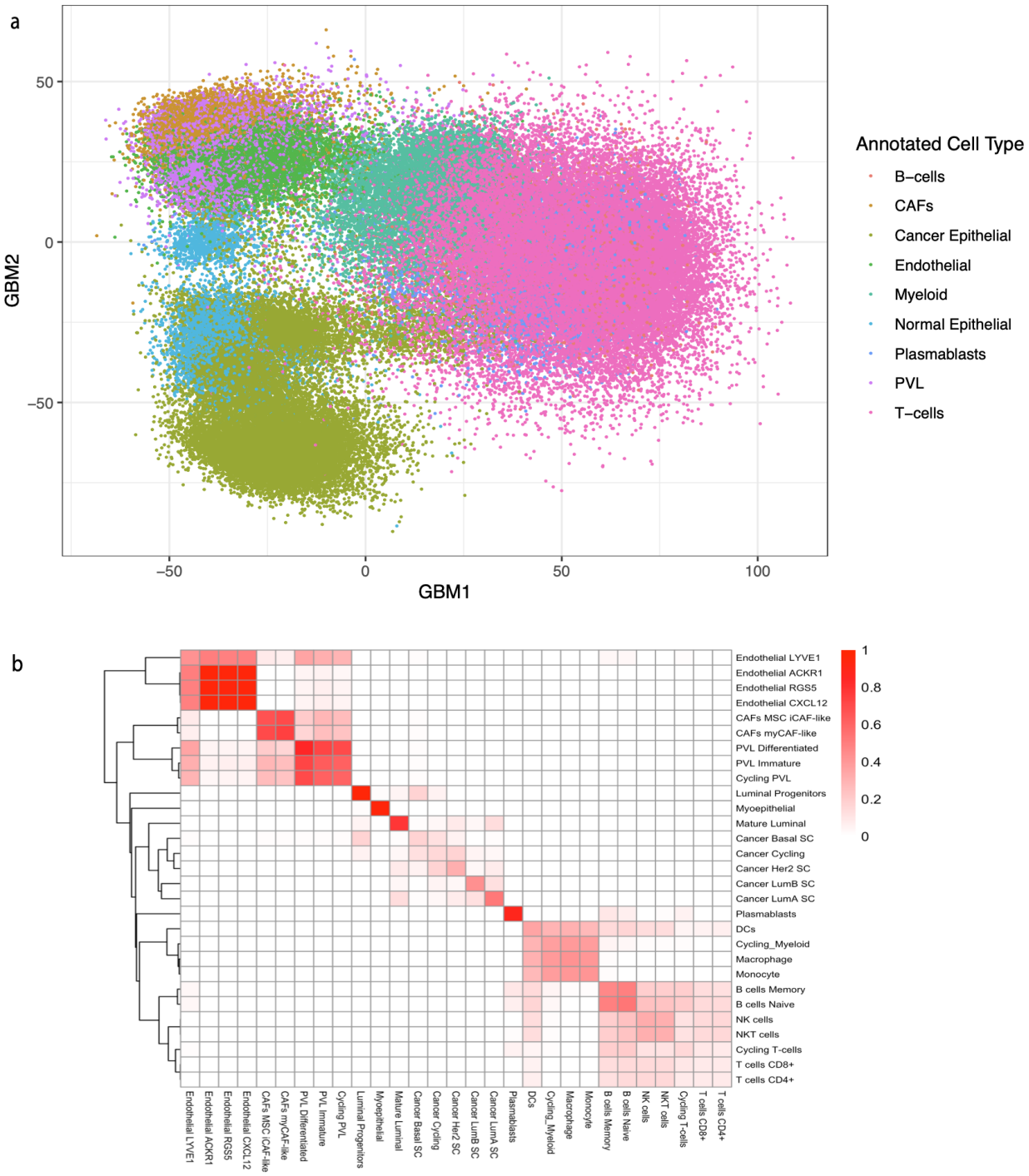

Figure 7: **a.** We applied scGBM to the Wu et al. (2021) dataset consisting of 100,064 breast cancer cells and plotted the first two scGBM scores (columns of $\hat{V}$) colored by major cell type. **b.** The heatmap of inter-cluster confidence indices for the minor cell types. The inter-CCIs clearly reveal relationships between the clusters that are not visible in the scatterplot, and even enable one to analyze the numerous minor cell types, which would be very difficult to visualize in a 2-d scatterplot.

biology, for instance, in terms of gene sets and pathways. To this end, it would be interesting to see if ideas from sparse PCA (Zou et al., 2006) can be extended to bilinear models.

Methods

Generalized bilinear model.

The scGBM method employs the Poisson bilinear model in Equation (4) for the matrix $Y \in \mathbb{R}^{I \times J}$ of UMI counts (I genes and J cells). If we define $\mu := [\mu_{ij}] \in \mathbb{R}^{I \times J}$ then Equation (4) can be rewritten in matrix form as

$$\begin{aligned} Y &\sim \text{Poisson}(\mu) \\ \log(\mu) &= \alpha \mathbf{1}_J^T + \mathbf{1}_I \beta^T + U \Sigma V^T \end{aligned}$$

where Poisson and log are applied entry-wise, $\mathbf{1}_K = (1, \ldots, 1)^T \in \mathbb{R}^K$ is a vector of ones, and $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_M) \in \mathbb{R}^{M \times M}$. Theorem 5.1 of Miller and Carter (2020) shows that this model is identifiable under the following constraints:

- $U^T U = V^T V = I_M$ (orthonormality),
- $\sigma_1 > \cdots > \sigma_M > 0$,
- the first non-zero entry in every column of U is positive,
- $\sum_{i=1}^I \alpha_i = 0$, $\sum_{i=1}^I u_{im} = 0$, and $\sum_{j=1}^J v_{jm} = 0$.

Many of the results presented in this section can easily be extended to cases with more complicated experimental designs (for example, adding row and column covariates), different link functions, and different outcome distributions. See Miller and Carter (2020) for more general GBM formulations.

Estimation via iteratively reweighted singular value decomposition (IRSVD).

Given a matrix $Z \in \mathbb{R}^{I \times J}$, the weighted low rank problem is to find a rank k matrix $X \in \mathbb{R}^{I \times J}$ that minimizes the weighted Frobenius norm:

$$\sum_{i,j} W_{ij}(Z_{ij} - X_{ij})^2 \tag{5}$$

where $W \in \mathbb{R}^{I \times J}$ is a matrix of known non-negative weights. When $W_{ij} = 1$ for all i and j, a solution can be found via the truncated singular value decomposition (SVD) of Z (Eckart and Young, 1936). Unfortunately, the general case cannot be reduced to an eigenvector problem unless $\text{rank}(W) = 1$. When the weights have been scaled to be in $[0, 1]$, Srebro and Jaakkola (2003) present the following iterative algorithm to find X:

$$X^{(t+1)} = \text{SVD}_k\big(W \circ Z + (1 - W) \circ X^{(t)}\big) \tag{6}$$

where $\circ$ is the Hadamard product and $\text{SVD}_k(X)$ denotes the rank k truncated SVD of X. Tuzhilina and Hastie (2021) note that Equation (6) can be seen as a projected gradient descent step: since the gradient is

$$\nabla_X \frac{1}{2} \big\| \sqrt{W} \circ (Z - X) \big\|_F^2 = -W \circ (Z - X), \tag{7}$$

a gradient step would be

$$X^{(t)} + \gamma\big(W \circ (Z - X^{(t)})\big) \tag{8}$$

which is the same as Equation (6) when the step size $\gamma = 1$. The projection back onto the set of rank k matrices is given by $\text{SVD}_k(\cdot)$. Tuzhilina and Hastie (2021) show that the convergence rate can be improved by using the acceleration method of Nesterov (1983):

$$Q^{(t)} = X^{(t)} + \frac{t-1}{t+2}(X^{(t)} - X^{(t-1)}) \tag{9}$$

$$X^{(t+1)} = \text{SVD}_k\big(W \circ Z + (1 - W) \circ Q^{(t)}\big). \tag{10}$$

We now show that maximum likelihood estimation in the Poisson bilinear model can be approximated by solving a weighted low rank problem. Denoting $X_{ij} = (U\Sigma V^T)_{ij}$, the log-likelihood is

$$\ell = \sum_{i=1}^{I}\sum_{j=1}^{J}\Big(Y_{ij}(\alpha_i + \beta_j + X_{ij}) - \exp(\alpha_i + \beta_j + X_{ij})\Big) + \text{const.} \tag{11}$$

Differentiating the log-likelihood with respect to X_{ij}, we have

$$\frac{\partial \ell}{\partial X_{ij}} = Y_{ij} - \exp(\alpha_i + \beta_j + X_{ij}) = Y_{ij} - \mu_{ij}. \tag{12}$$

Viewing μ_{ij} as a function of X_{ij}, that is, $\mu_{ij} = f(X_{ij})$ where $f(x) = \exp(\alpha_i + \beta_j + x)$, a first-order Taylor approximation at $\hat{X}_{ij}$ yields

$$\mu_{ij} = f(X_{ij}) \approx f(\hat{X}_{ij}) + f'(\hat{X}_{ij})(X_{ij} - \hat{X}_{ij}) = \hat{\mu}_{ij} + \hat{\mu}_{ij}(X_{ij} - \hat{X}_{ij}), \tag{13}$$

letting $\hat{\mu}_{ij} = \exp(\alpha_i + \beta_j + \hat{X}_{ij})$. Plugging this approximation into Equation (12), we have

$$\frac{\partial \ell}{\partial X_{ij}} \approx Y_{ij} - \hat{\mu}_{ij} - \hat{\mu}_{ij}(X_{ij} - \hat{X}_{ij}) = \hat{\mu}_{ij}\Big(\hat{X}_{ij} + \frac{Y_{ij} - \hat{\mu}_{ij}}{\hat{\mu}_{ij}} - X_{ij}\Big). \tag{14}$$

Thus, the gradient of the negative log-likelihood is approximately equal to the gradient of the weighted low rank problem (Equation (7)) with

$$\begin{aligned} W_{ij} &= \hat{\mu}_{ij} \\ Z_{ij} &= \hat{X}_{ij} + \frac{Y_{ij} - \hat{\mu}_{ij}}{\hat{\mu}_{ij}}. \end{aligned} \tag{15}$$

Note that, unlike the standard weighted low rank problem, the weights depend on the current parameter estimates rather than being fixed. This suggests the following iterative algorithm for estimating the parameters of the Poisson bilinear model, starting from initial estimates $\hat{\alpha}$, $\hat{\beta}$, and $\hat{X} = \widehat{U\Sigma V^T}$.

1. **Update intercepts.** Holding $\hat{\beta}$ and $\hat{X}$ fixed, for each i, the model reduces to a standard GLM with intercept α_i. In the Poisson case, there is a closed form solution for maximizing with respect to α_i:

$$\hat{\alpha}_i = \log\Big(\sum_{j=1}^{J} Y_{ij}\Big) - \log\Big(\sum_{j=1}^{J}\exp(\hat{\beta}_j + \hat{X}_{ij})\Big). \tag{16}$$

   By default, we fix $\hat{\beta}_j = \log(\sum_{i=1}^{I} Y_{ij})$, but if one wishes to estimate β, this can be done by updating

$$\hat{\beta}_j = \log\Big(\sum_{i=1}^{I} Y_{ij}\Big) - \log\Big(\sum_{i=1}^{I}\exp(\hat{\alpha}_i + \hat{X}_{ij})\Big). \tag{17}$$

2. **Update latent factors.** Given estimates $\hat{\alpha}$ and $\hat{\beta}$ along with our current estimate $\hat{X}$, define Z_{ij} and W_{ij} by Equation (15). Then standardize the weights to be in $[0, 1]$ by dividing each entry $\max_{ij} W_{ij}$. Finally, update $\hat{X}$ by setting

$$\hat{X}^{(\text{new})} = \text{SVD}_M\big(\hat{X} + \gamma W \circ (Z - \hat{X})\big). \tag{18}$$

   This can also be modified to use Nesterov acceleration as in Equation (9).

We find that $\gamma = 1$ works well for most single-cell datasets. However, the rate of convergence can be slightly improved by using an adaptive scheme where γ is increased by a multiplicative factor of 1.05 when the log-likelihood increases and decreases by a multiplicative factor of 2 when the log-likelihood decreases.

Using a good choice of initialization is important for convergence. If we were to initialize $\hat{X} = 0$, then after the first update to the intercepts, the weights would be

$$W_{ij} = \exp(\hat{\alpha}_i + \hat{\beta}_j), \tag{19}$$

which makes W a rank 1 matrix. When $\text{rank}(W) = 1$, it turns out that the weighted low-rank problem can be solved in one step by reducing to a standard SVD, as follows.

Proposition 1. *If $W = d_1 d_2^T$ where $d_1 \in \mathbb{R}^I$, $d_2 \in \mathbb{R}^J$, and $W_{ij} > 0$ for all i, j, then*

$$X = \sqrt{D_1^{-1}}\,\mathrm{SVD}_k\big(\sqrt{D_1} Z \sqrt{D_2}\big)\sqrt{D_2^{-1}} \tag{20}$$

minimizes Equation (5) subject to $\mathrm{rank}(X) \le k$, where $D_1 = \mathrm{diag}(d_1)$ and $D_2 = \mathrm{diag}(d_2)$.

Proof. The proof is due to Razenshteyn et al. (2016). Since $W_{ij} = d_{1i} d_{2j}$, we have

$$\sum_{i,j} W_{ij}(Z_{ij} - X_{ij})^2 = \big\|\sqrt{D_1}(Z - X)\sqrt{D_2}\big\|_F^2 = \big\|\sqrt{D_1} Z \sqrt{D_2} - A\big\|_F^2. \tag{21}$$

where $A = \sqrt{D_1} X \sqrt{D_2}$. Let $\mathcal{R}_k = \{A \in \mathbb{R}^{I \times J} : \mathrm{rank}(A) \le k\}$. By the Eckart–Young theorem, $A^* = \mathrm{SVD}_k(\sqrt{D_1} Z \sqrt{D_2})$ minimizes Equation (21) over $A \in \mathcal{R}_k$. Since $W_{ij} > 0$ implies $d_{1i} > 0$ and $d_{2j} > 0$, it follows that $X \mapsto \sqrt{D_1} X \sqrt{D_2}$ is a bijection from $\mathcal{R}_k$ to itself. Thus, $X^* = \sqrt{D_1^{-1}} A^* \sqrt{D_2^{-1}}$ minimizes Equation (21) over $X \in \mathcal{R}_k$. □

Applying Proposition 1 suggests using an initial estimate of

$$\hat{X} = \sqrt{D_1^{-1}}\mathrm{SVD}_M(\sqrt{D_1} Z \sqrt{D_2})\sqrt{D_2^{-1}} = \mathrm{SVD}_M\big((Y - W)/\mathrm{sqrt}(W)\big)/\mathrm{sqrt}(W) \tag{22}$$

where sqrt and / denote entry-wise square root and entry-wise division, respectively. Note that since $W = \hat{\mu}$, the entries of $(Y - W)/\mathrm{sqrt}(W)$ coincide with the Pearson residuals under a Poisson model. This derivation provides some theoretical justification for using PCA on the Pearson residuals to approximate the parameters of a GBM. However, the assumption that W is rank 1 (or close to it) is unlikely in cases where there is a large amount of latent structure. Thus, in many cases of interest, this will only be a rough approximation.

In practice, it is necessary to clip extreme values of $\hat{X}$ before proceeding, since extreme values can lead to instability in the SVD. That is, we hard threshold using $x \mapsto \max(\min(x, c), -c)$ to ensure all values of $\hat{X}$ are in $[-c, c]$. By default, we set $c = 8$. This clipping procedure is similar to the one used by scTransform (Hafemeister and Satija, 2019).

Combining with Equation (22), we arrive at our proposed initialization procedure:

Initialization. Set $\hat{\beta}_j = \log\left(\sum_{i=1}^I Y_{ij}\right)$, $\hat{\alpha}_i = \log\left(\sum_{j=1}^J Y_{ij}\right) - \log\left(\sum_{j=1}^J \exp(\hat{\beta}_j)\right)$, $W_{ij} = \exp(\hat{\alpha}_i + \hat{\beta}_j)$, and

$$\hat{X} = \mathrm{clip}\left[\mathrm{SVD}_M\Big(\big((Y - W)/\mathrm{sqrt}(W)\big)\Big)/\mathrm{sqrt}(W)\right] \tag{23}$$

where $\mathrm{clip}(x) = \max(\min(x, c), -c)$ is applied to each entry, with $c = 8$.

Computational complexity analysis.

0. **Initialization.** The rank M truncated SVD takes $O(IJM)$ time (Halko et al., 2011). The other operations only take $O(IJ)$ time, so initialization is $O(IJM)$ altogether.

1. **Update intercepts.** Each $\hat{\alpha}_i$ update takes $O(J)$ time, and each $\hat{\beta}_j$ update takes $O(I)$ time. Thus, the entire step requires $O(IJ)$ time.

2. **Update latent factors.** The entry-wise operations are $O(IJ)$ and the projection (rank M truncated SVD) takes $O(IJM)$ time (Halko et al., 2011).

Thus, the overall runtime for one iteration of IRSVD is $O(IJM)$. This is faster than the Fisher scoring algorithm of Miller and Carter (2020), which is $O(IJM^2)$ per iteration. The quadratic complexity in M can be practically significant since M is often chosen to be around 20-50. Further, IRSVD is simpler than the Miller and Carter (2020) algorithm, making it easier to implement and yielding a smaller constant.

Projection method.

For datasets with an extremely large number of cells J, it is computationally infeasible to perform the SVD since the Z matrix in Equation (18) is dense. If the relevant cell subpopulations are sufficiently large, it is reasonable to first estimate U using a random subset of cells and then estimate $V\Sigma$ while holding U fixed. In PCA, estimating V given U is accomplished by projecting onto the subspace spanned by the columns of U. In a GBM, the corresponding "projection" is accomplished by fitting a GLM to each column of Y. Specifically, holding α and U fixed at their current estimates, for each j we estimate $\beta_j, v_{j1}, \ldots, v_{jM}$ by fitting the model in Equation (4) to data $Y_{1j}, \ldots, Y_{Ij}$. The initial α and U are estimated using a small subsample of cells. Thus, estimating V given U can be accomplished by fitting J GLMs, each with M coefficients. An advantage of this approach is that the GLMs can be fit in parallel, making it computationally tractable even when J is very large.

Uncertainty quantification.

Standard errors for the entries of U and V can be estimated using the classical method of inverting the Fisher information matrix. Miller and Carter (2020) use this approach, accounting for joint uncertainty in U and V as well as their identifiability constraints, by inverting the constraint-augmented Fisher information. However, this is computationally expensive on single-cell datasets. Thus, we use a rough approximation by inverting the diagonal blocks of the Fisher information, that is, the $M \times M$ submatrices

$$I_{v_j} := \left[-E\left(\frac{\partial^2 \ell}{\partial v_{jm} \partial v_{jm'}} \right) \right]_{m,m'} \in \mathbb{R}^{M \times M} \tag{24}$$

for $j = 1, \ldots, J$, where

$$\frac{\partial^2 \ell}{\partial v_{jm} \partial v_{jm'}} = -\sum_{i=1}^{I} \sigma_m u_{im} \mu_{ij} \sigma_{m'} u_{im'} \tag{25}$$

by differentiating the log-likelihood ℓ. In matrix form,

$$I_{v_j} = (U\Sigma)^T \tilde{W}_j (U\Sigma) \tag{26}$$

where $\tilde{W}_j = \mathrm{diag}(\mu_{1j}, \ldots, \mu_{Ij}) \in \mathbb{R}^{I \times I}$. Taking the square root of the diagonal entries of $I_{v_j}^{-1}$, we obtain approximate standard errors for $v_{j1}, \ldots, v_{jM}$. Uncertainty in U can be approximated in the same way. In practice, we absorb Σ into V (that is, the scores are $V\Sigma$), so in this case the standard errors can be computed by $U^T \tilde{W}_j U$.

This approach will tend to underestimate the uncertainty in V since it treats the other parameters U, Σ, α, β as fixed. Nonetheless, on simulated data, we find that this yields confidence intervals with coverage only slightly below the target coverage, for large I and J (Figure S4). Improving the calibration of scGBM uncertainty quantification is outside the scope of this paper but is an area for future work.

Adaptive shrinkage for stabilization.

Miller and Carter (2020) and Townes (2019) use ℓ_2 penalties on the factors U and V to stabilize the estimates via shrinkage towards 0. However, this requires a tuning parameter to be chosen in advance. Instead, scGBM uses a *post hoc* adaptive shrinkage method for the weights U. Let $\hat{U} \in \mathbb{R}^{I \times M}$ be the estimate of the U matrix, and let $s_{im} := \mathrm{se}(\hat{U}_{im})$ be the estimated standard error of entry $\hat{U}_{im}$. Consider the following spike-and-slab model for U, introduced by Stephens (2017):

$$\hat{U}_{im} | U_{im} \sim \mathcal{N}(U_{im}, s_{im}^2) \tag{27}$$

$$U_{im} \sim \pi_0 \delta_0 + \sum_{k=1}^{K} \pi_k \mathcal{N}(0, \tau_k^2) \tag{28}$$

for each i and m independently, where δ_0 denotes the point mass at zero. The component variances $\tau_1^2, \ldots, \tau_K^2$ are pre-specified and the mixture weights $\pi_0, \ldots, \pi_K$ are estimated using an empirical Bayes procedure. The

idea is that the true parameter value U_{im} is drawn from a spike-and-slab distribution and the empirical estimate $\hat{U}_{im}$ is drawn from a normal distribution with mean U_{im} and standard deviation equal to the estimated standard error of $\hat{U}_{im}$. The posterior on U_{im} given $\hat{U}_{im}$ is then used for uncertainty quantification. The method is called *adaptive shrinkage* and is implemented in the R package *ashr* (Stephens et al., 2022). In Figure 4d, the stabilized estimates are defined as the posterior mean of U_{im} given $\hat{U}_{im}$ under the *ash* model.

Cluster confidence index.

Given estimates $\hat{v}_{jm}$ and corresponding standard errors $\text{se}(\hat{v}_{jm})$, we aim to quantitatively analyze the stability of a given clustering of cells. Let $c_1, \ldots, c_J \in \{1, \ldots, K\}$ be assignments of the J cells to K clusters. To compute the *cluster confidence indices* (CCIs), we repeat the following steps n times (by default, $n = 100$):

1. Draw $\tilde{v}_{jm} \sim \mathcal{N}(\hat{v}_{jm}, \text{se}(\hat{v}_{jm})^2)$ for $j = 1, \ldots, J$, $m = 1, \ldots, M$.
2. Apply a clustering algorithm to the rows of $\tilde{V} = [\tilde{v}_{jm}]$ to obtain new cluster assignments $\tilde{c}_1, \ldots, \tilde{c}_J$.
3. For each pair of clusters $k, k' \in \{1, \ldots, K\}$, compute the fraction

$$f_{k,k'} = \frac{\sum_{(j,j') \in S} \text{I}(\tilde{c}_j = \tilde{c}_{j'})}{\sum_{(j,j') \in S} \text{I}(c_j = k, c_{j'} = k')} \tag{29}$$

   where S denotes the set of pairs $j, j' \in \{1, \ldots, J\}$ such that $j \neq j'$. Here, $\text{I}(\cdot)$ is the indicator function.

Then, for each cluster k, the CCI is defined as the median (across the n repetitions) of the fraction $f_{k,k}$. Likewise, the inter-cluster confidence index (inter-CCI) for clusters k and k' is defined as the median of $f_{k,k'}$. That is, the inter-CCI is the median of the fraction of all pairs of points j and j' that are in the same cluster after resampling, out of all pairs j and j' that were originally in clusters k and k', respectively.

The interpretation is that a low CCI indicates that this cluster could be an artifact of sampling variability. Likewise, a high inter-CCI for clusters k and k' indicates that the separation of these two clusters may be an artifact of sampling variability. The clustering algorithm in step 2 can be specified by the user, and does not have to be the same as the algorithm that was used to create the original assignments $c_1, \ldots, c_J$.

Controlling for known batches.

When there are known batches, the model can be adjusted to have a gene-level intercept for each batch:

$$\log(\mu_{ij}) = \alpha_{ib_j} + \beta_j + \sum_{m=1}^{M} \sigma_m u_{im} v_{jm} \tag{30}$$

where $b_j \in \{1, \ldots, B\}$ is the index of the batch that cell j belongs to, and α_{ib} is the intercept for gene i and batch b. Estimation in this model is the same as before, except Equation (16) is replaced with

$$\hat{\alpha}_{ib} = \log\Big(\sum_{j\,:\,b_j = b} Y_{ij}\Big) - \log\Big(\sum_{j\,:\,b_j = b} \exp(\hat{\beta}_j + \hat{X}_{ij})\Big). \tag{31}$$

Choosing the number of latent factors.

The latent dimension M should be set large enough to ensure that all relevant biological variability is captured. In practice, we recommend setting M between 20 and 50. A reasonable cutoff point can sometimes be selected by plotting σ_m as a function of m. Since the estimated V and U can depend on the choice of M, we recommend removing all latent factors past the elbow point (as opposed to refitting the model with the new choice of M). For example, on the 10X immune cell data (Zheng et al., 2017, see Table 1), we find that there are possible elbows at $m = 3$ and $m = 14$ (Figure S2). Identifying the best cutoff point is subjective, but one useful approach is to consider the second order differences, as suggested by Evanno et al. (2005) for selecting the number of populations in an admixture model (Pritchard et al., 2000).

Details of single-cell datasets.

Unless otherwise specified, all single-cell data was processed using Seurat Satija et al. (2015). In particular, $I = 2000$ variable genes were identified using the function `FindVariableFeatures()`. We then used these genes as the input to scGBM.

- The purified monocyte data was downloaded from www.10xgenomics.com.
- The 10X immune cell data (which includes the Naive T cells and memory T cells used in Figure 5) were downloaded from the *DuoClustering2018* R package (Duò et al., 2018).
- The COVID-19 data (Wilk et al., 2020) was downloaded as a Seurat object from www.covid19cellatlas.org.
- The 10X mouse brain data was downloaded from the Bioconductor package *TENxBrain* (Lun and Morgan, 2020).
- The Wu et al. (2021) data was downloaded from the Gene Expression Omnibus (GEO) database (GSE176078).

Software and code availability.

scGBM is available for download as an R package at github.com/phillipnicol/scGBM. The repository also includes scripts for replicating the figures based on simulated data.

Acknowledgements

PBN is supported by the National Institutes of Health grant T32CA009337. JWM is supported by the National Institutes of Health grant 5R01CA240299-02.

References

F. Agostinis, C. Romualdi, G. Sales, and D. Risso. NewWave: a scalable R/Bioconductor package for the dimensionality reduction and batch effect removal of single-cell RNA-seq data. *Bioinformatics*, 38(9): 2648–2650, 2022.

J. Baglama and L. Reichel. Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM Journal on Scientific Computing*, 27(1):19–42, 2005.

A. S. Booeshaghi and L. Pachter. Normalization of single-cell RNA-seq counts by log(x+1) or log(1+x). *Bioinformatics*, 37(15):2223–2224, 2021.

J. Cao, M. Spielmann, X. Qiu, X. Huang, D. M. Ibrahim, A. J. Hill, F. Zhang, S. Mundlos, L. Christiansen, F. J. Steemers, et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566 (7745):496–502, 2019.

V. Choulakian. Generalized bilinear models. *Psychometrika*, 61(2):271–283, 1996.

A. Duò, M. D. Robinson, and C. Soneson. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, 7, 2018.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3): 211–218, 1936.

G. Evanno, S. Regnaut, and J. Goudet. Detecting the number of clusters of individuals using the software STRUCTURE: a simulation study. *Molecular Ecology*, 14(8):2611–2620, 2005.

I. N. Grabski, K. Street, and R. A. Irizarry. Significance analysis for clustering with single-cell RNA-sequencing data. *bioRxiv*, 2022.

C. Hafemeister and R. Satija. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biology*, 20(1):1–15, 2019.

N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

V. Y. Kiselev, T. S. Andrews, and M. Hemberg. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome Biology*, 21(1):1–35, 2020.

J. Lause, P. Berens, and D. Kobak. Analytic Pearson residuals for normalization of single-cell RNA-seq UMI data. *Genome Biology*, 22(1):1–20, 2021.

J. T. Leek and J. D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):e161, 2007.

M. D. Luecken and F. J. Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019.

A. Lun and M. Morgan. *TENxBrainData: Data from the 10X 1.3 Million Brain Cell Study*, 2020. R package version 1.8.0.

L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

J. W. Miller and S. L. Carter. Inference in generalized bilinear models. *arXiv preprint arXiv:2010.04896*, 2020.

L. Morelli, V. Giansanti, and D. Cittaro. Nested stochastic block models applied to the analysis of single cell data. *BMC Bioinformatics*, 22(1):1–19, 2021.

Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. In *Dokl. Akad. Nauk SSSR,*, volume 269, pages 543–547, 1983.

H. Pagès. *DelayedArray: A unified framework for working transparently with on-disk and in-memory array-like datasets*, 2020. R package version 0.14.1.

S. Petropoulos, D. Edsgärd, B. Reinius, Q. Deng, S. P. Panula, S. Codeluppi, A. P. Reyes, S. Linnarsson, R. Sandberg, and F. Lanner. Single-cell RNA-seq reveals lineage and X chromosome dynamics in human preimplantation embryos. *Cell*, 165(4):1012–1026, 2016.

H. A. Pliner, J. Shendure, and C. Trapnell. Supervised classification enables rapid annotation of cell atlases. *Nature Methods*, 16(10):983–986, 2019.

J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

I. Razenshteyn, Z. Song, and D. P. Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 250–263, 2016.

D. Risso, J. Ngai, T. P. Speed, and S. Dudoit. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature Biotechnology*, 32(9):896–902, 2014.

D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications*, 9(1):1–17, 2018.

J. C. Roden, B. W. King, D. Trout, A. Mortazavi, B. J. Wold, and C. E. Hart. Mining gene expression data by interpreting principal components. *BMC Bioinformatics*, 7(1):1–22, 2006.

A.-E. Saliba, A. J. Westermann, S. A. Gorski, and J. Vogel. Single-cell RNA-seq: advances and future challenges. *Nucleic Acids Research*, 42(14):8845–8860, 2014.

A. Sarkar and M. Stephens. Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis. *Nature Genetics*, 53(6):770–777, 2021.

R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, 2015.

P. Savarese, D. McAllester, S. Babu, and M. Maire. Domain-independent dominance of adaptive methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16286–16295, 2021.

M. Soumillon, D. Cacchiarelli, S. Semrau, A. van Oudenaarden, and T. S. Mikkelsen. Characterization of directed differentiation by high-throughput single-cell RNA-seq. *bioRxiv*, page 003236, 2014.

N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 720–727, 2003.

M. Stephens. False discovery rates: a new deal. *Biostatistics*, 18(2):275–294, 2017.

M. Stephens, P. Carbonetto, D. Gerard, M. Lu, L. Sun, J. Willwerscheid, and N. Xiao. *ashr: Methods for Adaptive Shrinkage, using Empirical Bayes*, 2022. URL https://CRAN.R-project.org/package=ashr. R package version 2.2-54.

Y. Sun, N. R. Zhang, and A. B. Owen. Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *The Annals of Applied Statistics*, 6(4):1664–1688, 2012.

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

F. W. Townes. Generalized principal component analysis. *arXiv preprint arXiv:1907.02647*, 2019.

F. W. Townes and K. Street. *glmpca: Dimension Reduction of Non-Normally Distributed Data*, 2020. URL https://CRAN.R-project.org/package=glmpca. R package version 0.2.0.

F. W. Townes, S. C. Hicks, M. J. Aryee, and R. A. Irizarry. Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biology*, 20(1):1–16, 2019.

V. A. Traag, L. Waltman, and N. J. Van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019.

E. Tuzhilina and T. Hastie. Weighted low rank matrix approximation and acceleration. *arXiv preprint arXiv:2109.11057*, 2021.

C. A. Vallejos, D. Risso, A. Scialdone, S. Dudoit, and J. C. Marioni. Normalizing single-cell RNA sequencing data: challenges and opportunities. *Nature Methods*, 14(6):565–571, 2017.

L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (11), 2008.

F. A. Van Eeuwijk. Multiplicative interaction in generalized linear models. *Biometrics*, pages 1017–1032, 1995.

A. J. Wilk, A. Rustagi, N. Q. Zhao, J. Roque, G. J. Martínez-Colón, J. L. McKechnie, G. T. Ivison, T. Ranganath, R. Vergara, T. Hollis, et al. A single-cell atlas of the peripheral immune response in patients with severe COVID-19. *Nature Medicine*, 26(7):1070–1076, 2020.

S. Z. Wu, G. Al-Eryani, D. L. Roden, S. Junankar, K. Harvey, A. Andersson, A. Thennavan, C. Wang, J. R. Torpy, N. Bartonicek, et al. A single-cell and spatially resolved atlas of human breast cancers. *Nature Genetics*, 53(9):1334–1347, 2021.

G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):1–12, 2017.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

Supplementary Materials

A Technical details of simulations

Single marker gene simulation. We generated $J = 1000$ cells such that 333 are from cell type A, 333 are from cell type B, and 334 are from cell type C. For gene 1, counts are independently generated as Poisson(10) for cell type A and Poisson(1) for all other cells. For gene 2, counts are independently generated as Poisson(50) for cell type B and Poisson(1) for all other cells. For genes $3, 4, \ldots, 1000$ (the null genes), all counts are independently generated as Poisson(1). Figure 1a shows boxplots of the simulated counts.

Projection method simulation. To test the accuracy of the projection method on a simulation with known ground truth, we simulated latent factors and data as follows with $I = 1000$, $J = 10000$, and $M = 10$. Following the procedure described in Section S2 of Miller and Carter (2020), the ground truth matrices U and V were drawn uniformly from the rank M Stiefel manifolds over $\mathbb{R}^I$ and $\mathbb{R}^J$, respectively. The intercepts α_i and β_j were drawn independently from $\mathcal{N}(0, 1)$. The diagonal elements of Σ were uniformly spaced between $\sqrt{I} + \sqrt{J}$ and $\kappa(\sqrt{I} + \sqrt{J})$, where κ is a parameter that controls the amount of latent variability. Miller and Carter (2020) use $\kappa = 2$, which we call low latent variability (LLV). We also use $\kappa = 5$ and call this high latent variability (HLV).

We then compared scGBM-full to scGBM-proj (projection method) using various subsample sizes in the projection method, using $M = 10$ for both methods. Specifically, we considered subsample sizes ranging between 100 and 5000 (1%-50%). For each subsample size, we ran both methods on 100 trial experiments, where each trial was performed by simulating a full dataset with $I = 1000$ and $J = 10000$ and choosing a random subsample of the desired size. We assessed the results using two metrics. First, we considered the relative mean squared error between the estimated and ground truth V:

$$\frac{\sum_{j=1}^{J}\sum_{m=1}^{M}(\hat{V}_{jm} - V_{jm})^2}{\sum_{j=1}^{J}\sum_{m=1}^{M} V_{jm}^2} \tag{32}$$

as defined on page S12 of Miller and Carter (2020). We also considered the absolute value of the correlation between columns of true and estimated V, as used above in Figure 3. The results are shown in Figure S1.

B Supplementary figures and tables

Learning rate	Deviance (after 100 iterations)
0.1 (Default)	∞
0.03	∞
0.02	9.61×10^5
0.01	9.51×10^5
0.001	1.18×10^6

Table S1: **Sensitivity of GLM-PCA (Avagrad) to learning rate.** Using GLM-PCA (Townes and Street, 2020) with `optimizer="Avagrad"` on simulated data with $I = J = 1000$ and $d = 10$, the GLM-PCA objective function (Deviance) after 100 iterations is reported for several choices of learning rate (step size for gradient descent). If the learning rate is too large, then the algorithm diverges, indicated by a value of ∞. If the learning rate is too low, then the rate of convergence is very slow. In practice, it may be difficult to find a learning rate that works well for a particular dataset, and thus, the runtime of GLM-PCA can be significantly longer than what is reported in Table 1.

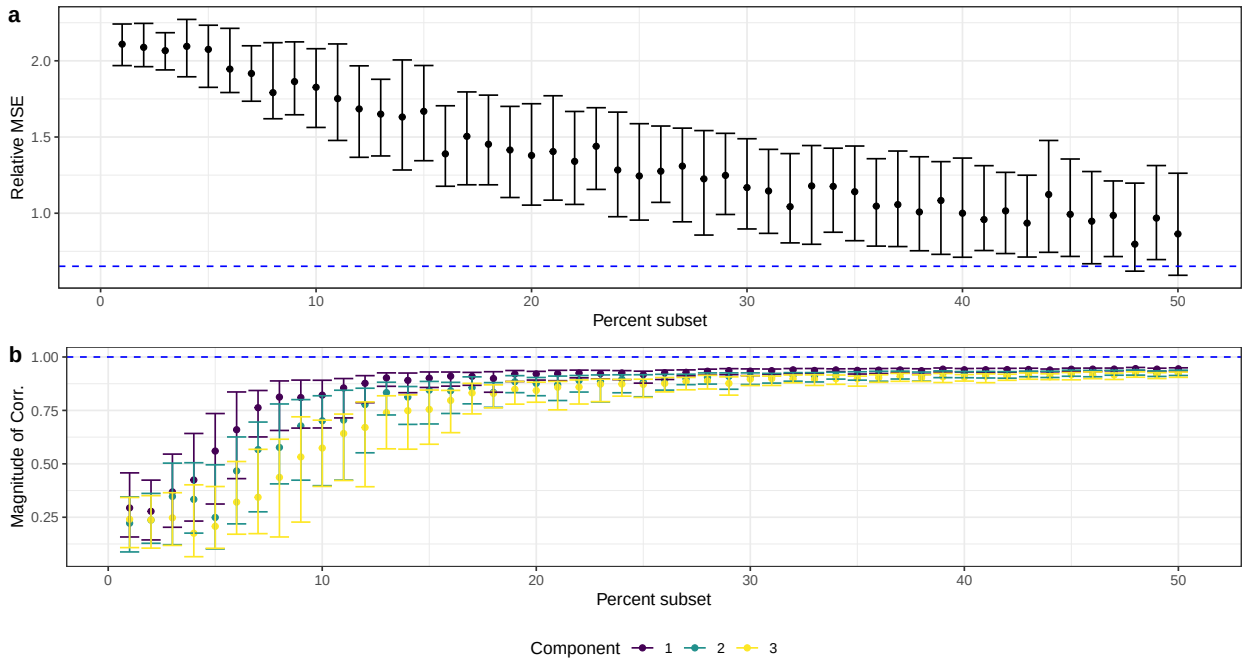

Figure S1: Testing the accuracy of the projection method using simulated data ($I = 1000$, $J = 10000$, $M = 10$). **a.** Relative MSE between ground truth and scGBM-proj estimate of V as a function of subsample size used. The points are the median across 100 trials and the error bars represent the interquartile range. The dashed blue line is the median of the relative MSE for scGBM-full. **b.** Absolute value of the correlation between ground truth and scGBM-proj estimate of V as a function of subsample size. This is a simulation-based replication of the real data analysis in Figure 3.

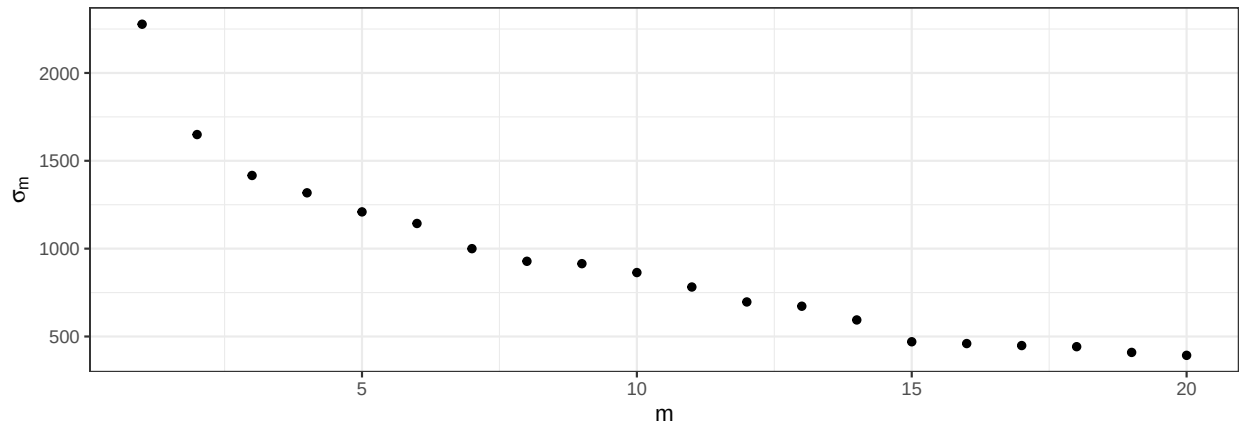

Figure S2: Scree plot of σ_m versus m for the 10X immune cell data (Zheng et al., 2017). Elbows (changes in slope) appear at roughly $m = 3$ and $m = 14$, making these candidate cutoff points for selecting the number of latent factors.

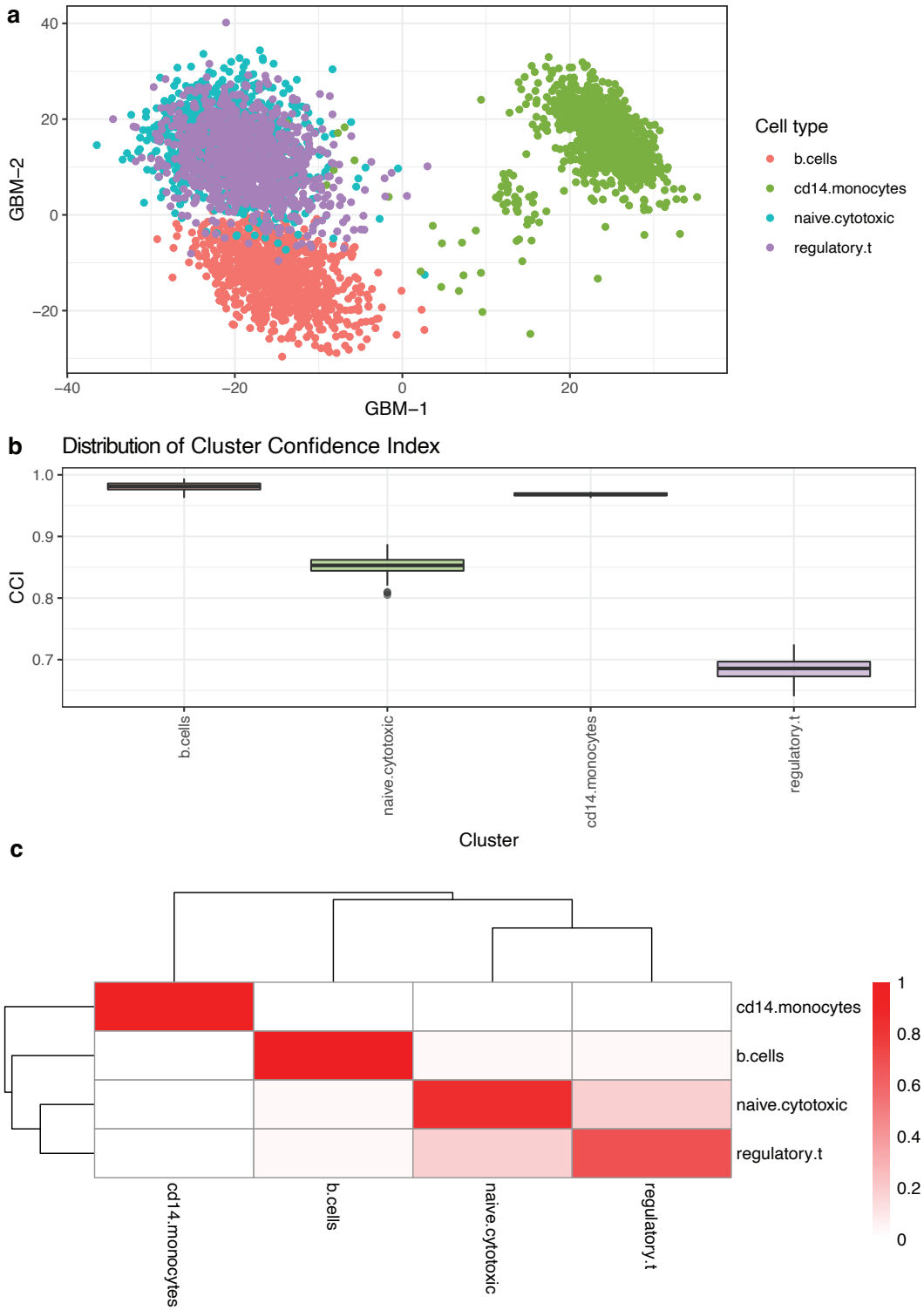

Figure S3: Cluster confidence indices for the Zhengmix4eq dataset from Duò et al. (2018) (originally generated by Zheng et al., 2017). **a.** Scatterplot of V scores for scGBM latent factors 1 and 2, labelled by known cell type. **b.** The CCI for all four cell types in the dataset is relatively high, indicating high stability. Boxplots are over 100 resampled values of $\tilde{V}$. Clustering algorithm used was k-means with $k = 4$. **c.** The cluster confidence heatmap indicates the amount of overlap between different clusters.

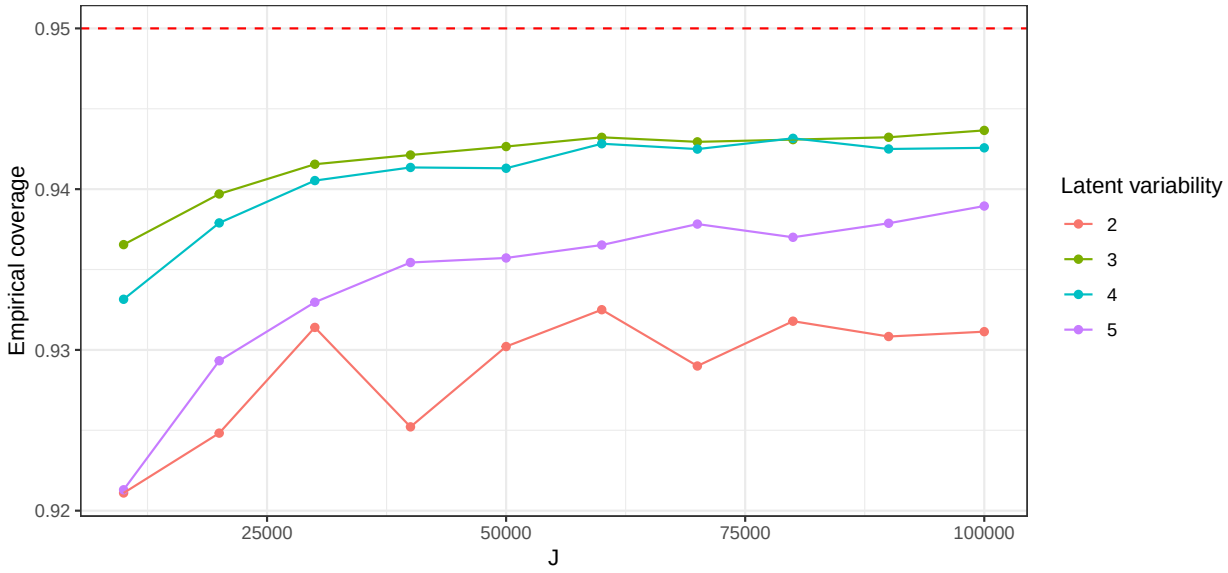

Figure S4: Empirical coverage for different choices of J and latent variability $\kappa = \sigma_1/\sigma_M$. 95% confidence intervals were formed assuming normality: $\hat{v}_{jm} \pm 1.96\text{se}(\hat{v}_{jm})$. The empirical coverage is the fraction of times this interval contains the true v_{jm}. Points are the median across 20 replicates.

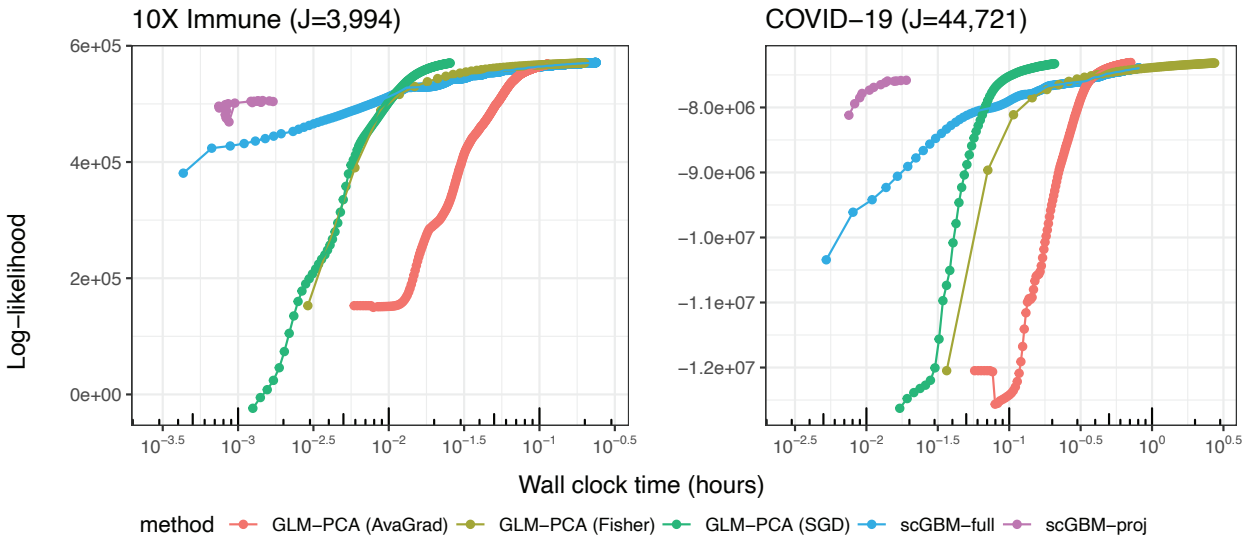

Figure S5: The same analysis as Figure 2c applied to two real datasets (described in the main text Table 1).