SCIENTIFIC REPORTS

OPEN

Received: 18 May 2017
Accepted: 29 September 2017
Published online: 19 October 2017

Controlling for Confounding Effects in Single Cell RNA Sequencing Studies Using both Control and Target Genes

Mengjie Chen[1,2] & Xiang Zhou[3,4]

Single cell RNA sequencing (scRNAseq) technique is becoming increasingly popular for unbiased and high-resolutional transcriptome analysis of heterogeneous cell populations. Despite its many advantages, scRNAseq, like any other genomic sequencing technique, is susceptible to the influence of confounding effects. Controlling for confounding effects in scRNAseq data is a crucial step for accurate downstream analysis. Here, we present a novel statistical method, which we refer to as scPLS (single cell partial least squares), for robust and accurate inference of confounding effects. scPLS takes advantage of the fact that genes in a scRNAseq study often can be naturally classified into two sets: a control set of genes that are free of effects of the predictor variables and a target set of genes that are of primary interest. By modeling the two sets of genes jointly using the partial least squares regression, scPLS is capable of making full use of the data to improve the inference of confounding effects. With extensive simulations and comparisons with other methods, we demonstrate the effectiveness of scPLS. Finally, we apply scPLS to analyze two scRNAseq data sets to illustrate its benefits in removing technical confounding effects as well as for removing cell cycle effects.

Single-cell RNA sequencing (scRNAseq) has emerged as a powerful tool in genomics. While the traditional RNA sequencing, known as the bulk RNAseq, measures gene expression levels averaged across many different cells in a sample of potentially heterogeneous cell population, scRNAseq can measure gene expression levels directly at the single cell resolution. As a result, scRNAseq is less influenced by the variation of cell type and cell composition across different samples–a major confounding in the analyses of bulk RNAseq studies. Because of this benefit and its high resolution, scRNAseq provides unprecedented insights into many basic biological questions that are previously difficult to address. For example, scRNAseq has been applied to classify novel cell subtypes[1,2] and cellular states[3,4], reconstruct cell lineage and quantify progressive gene expression during development[5–8], perform spatial mapping and re-localization[9,10], identify differentially expressed genes and gene expression modulars[11–13], and investigate the genetic basis of gene expression variation by detecting heterogenic allelic specific expressions[14,15].

Like any other genomic sequencing experiment, scRNAseq studies are influenced by many factors that can introduce unwanted variation in the sequencing data and confound the down-stream analysis[16]. However, such unwanted variation are often exacerbated in scRNAseq experiments due to a range of scRNAseq specific conditions that include amplification bias, low amount of input material and low transcript capture efficiency[17]; dropout events that are driven by both biological and technical factors[18,19]; global changes in expression due to transcriptional bursts[20]; as well as changes in cell cycle and cell size[21]. Indeed, adjusting for confounding factors in scRNAseq data has been shown to be crucial for accurate estimation of gene expression levels and successful down-stream analysis[16–18,22,23]. However, depending on the source, adjusting for confounding factors in scRNAseq can be non-trivial. Some confounding effects, such as read sampling noise and drop-out events, are direct consequences of low sequencing-depth, which are random in nature and can be readily addressed by probabilistic modeling using existing statistical methods[18,22–25]. Other confounding effects are inherent to a particular

[1]Department of Medicine, University of Chicago, Chicago, IL 60637, USA. [2]Department of Human Genetics, University of Chicago, Chicago, IL 60637, USA. [3]Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA. [4]Center for Statistical Genetics, University of Michigan, Ann Arbor, MI 48109, USA. Correspondence and requests for materials should be addressed to M.C. (email: mengjiechen@uchicago.edu) or X.Z. (email: xzhousph@umich.edu)

experimental protocol and can cause amplification bias, but can be easily mitigated by using new protocols[26]. Yet other confounding effects are due to observable batches and can be adjusted for by including batch labels and technician ids as covariates or dealt with other statistical methods[27,28]. However, many confounding factors are hidden and are difficult or even impossible to measure. Common hidden confounding factors include various technical artifacts during library preparation and sequencing, and unwanted biological confounders such as cell cycle status. These hidden confounding factors can cause systematic bias, are notoriously difficult to control for, and are the focus of the present study.

To effectively infer and control for hidden confounding factors in scRNAseq studies, we develop a novel statistical method, which we refer to as scPLS (single cell partial least squares). scPLS takes advantage of the fact that genes in a scRNAseq study can often be naturally classified into two sets: a control set of genes that are free of effects of the predictor variables and a target set of genes that are of primary interest. By modeling the two sets of genes jointly using the partial least squares regression, scPLS is capable of making full use of the data to improve the inference of confounding factors. scPLS is closely related to and bridges between two existing subcategories of methods for transcriptome analysis: a subcategory of methods that treat control and target genes in the same fashion (e.g. PCA[29–32] and LMM[33–35]), and another subcategory of methods that use control genes alone for inferring confounding factors (e.g. RUV[29,36] and scLVM[37]). By bridging between the two subcategories of methods, scPLS enjoys robust performance across a range of application scenarios. scPLS is also computationally efficient: with a new block-wise expectation maximization (EM) algorithm, it is scalable to thousands of cells and tens of thousands of genes. Using simulations and two real data applications, we show how scPLS can be used to remove confounding effects and enable accurate down-stream analysis in scRNAseq studies. Our method is implemented as a part of the Citrus project and is freely available at: http://chenmengjie.github.io/Citrus/.

The paper is organized as follows. In the Review of Previous Methods section, we provide a brief review of existing statistical methods for removing confounding effects in transcriptome analysis and describe how scPLS is related to and motivated from these methods. In the Method Overview Section, we provide a methodological description of the scPLS model, with inference details provided in the Methods Section. In the Simulations section we present comparisons between scPLS and several existing methods using simulations. In Real Data Applications section, we apply scPLS to two real scRNAseq data sets to remove technical confounding effects or cell cycle effects. Finally, we conclude the paper with a summary and discussion.

Review of Previous Methods

Many statistical methods have been developed in sequencing- and array-based genomic studies to infer hidden confounding factors and control for hidden confounding effects. Based on their targeted application, these statistical methods can be generally classified into two categories.

The first category of methods are supervised and application-specific: these methods are designed to infer the confounding factors in the presence of a *known* predictor variable, and to correct for the confounding effects without removing the effects of the predictor variable. For example, scientists are often interested in identifying genes that are differentially expressed between two pre-determined treatment conditions or that are associated with a measured predictor variable of interest. To remove the confounding effects in these applications, methods, include SVA[30], sparse regression models[38,39], and, more recently, RUV[40,41], are developed. Although these application-specific methods are widely applied in many genomics studies, their usage is naturally restricted to cases where the primary variable of interest is known. The application-specific methods become inconvenient in cases where there are multiple variables of interest (e.g. in eQTL mapping problems). They also become inapplicable when the primary variable of interest is not observed (e.g. in clustering problems).

The second category of methods are unsupervised, and are designed to infer the confounding factors without knowing or using the predictor variable of interest. Our scPLS belongs to this category. Notable applications of unsupervised methods in scRNAseq studies include cell type clustering and classification[1–8]. Existing unsupervised statistical methods can be further classified into two subcategories. The first subcategory of methods treat all genes in the same fashion and use all of them to infer the confounding factors. For example, the principal component analysis (PCA) or the factor model extracts the principal components or factors from all genes (or all highly variable genes) as surrogates for the confounding factors[29–32]. The inferred factors are treated as covariates whose effects are further removed from gene expression levels before downstream analyses. Similarly, the linear mixed models (LMMs) construct a sample relatedness matrix based on all genes to capture the influence of the confounding factors[33–35]. The relatedness matrix are then included in the downstream analyses to control for the confounding effects. In contrast, the second subcategory of unsupervised methods are recently developed to take advantage of a set of control genes for inferring the confounding factors[29,37]. These methods divide genes into two sets: a control set of genes that are known to be free of effects of interest *a priori* and a target set of genes that are of primary interest. Unlike the first subcategory, the second subcategory of methods treat the two gene sets differently in inferring the confounding factors: the confounding factors are only inferred from the control set, and are then used to remove the confounding effects in the target genes for subsequent downstream analysis. For example, scRNAseq studies often add ERCC spike-in controls prior to the PCR amplification and sequencing steps. The spike-in controls can be used to capture the hidden confounding technical factors associated with the experimental procedures, which are further used to remove technical confounding effects (e.g. reverse transcription or PCR amplification confounding effects) from the target genes[33]. Similarly, most scRNAseq studies include a set of control genes that are known to have varying expression levels across cell cycles. These cell cycle genes can be used to capture the unmeasured cell cycle status of each cell, which are further used to remove cell cycle effects in the target genes[37]. Prominent methods in the second subcategory include the unsupervised version of RUV[29,36] and scLVM[37].

The two subcategories of unsupervised methods use different strategies to infer the confounding factors. Therefore, these two sets of methods are expected to perform well in different settings. Specifically, the first subcategory of methods have the advantage of using information contained in all genes to accurately infer the confounding effects. However, when the predictor variable of interest influences a large number of genes, then this subcategory of methods may incorrectly remove the primary effects of interest. On the other hand, the second subcategory of methods infer confounding factors only from the control genes and are thus not prone to mistakenly removing the primary effects of interest. However, these methods overlook one important fact–that the hidden confounding factors not only influence the control genes but also the target genes, i.e. the exact reason that we need to remove such confounding effects in the first place. Because the confounding factors influence both control and target genes, using control genes alone to infer the confounding factors can be suboptimal as it fails to use the information from target genes.

To more effectively infer and control for hidden confounding factors in scRNAseq studies, we develop a novel statistical method, which we refer to as scPLS (single cell partial least squares). scPLS bridges between the two subcategories of unsupervised methods and effectively includes each as a special case. Like the first subcategory of methods, scPLS models both control and target genes jointly to infer the confounding factors. Like the second subcategory of methods, scPLS is capable of taking advantage of a control set to guild the inference of confounding factors. scPLS builds upon the partial least squares regression model and relies on a key modeling assumption that only target genes contain the primary effects of interest or other systematic biological variations. By incorporating such systematic variations in the target genes only, we can jointly model both control and target genes to infer the confounding effects while avoiding mis-removing the primary effects of interest. Therefore, scPLS has the potential to make full use of the data to improve the inference of confounding factors and the removal of confounding effects.

Results

scPLS Method Overview. We provide modeling details for scPLS here. While the formulation of scPLS is general, we focus on its application in scRNAseq. The scRNAseq data resembles that of the bulk RNAseq data and consists of a gene expression matrix on n cells and $p + q$ genes. We consider dividing the genes into two sets: a control set that contains q control genes and a target set that contains p genes of primary interest. The control genes are selected based on the purpose of the analysis. For example, the control set would contain ERCC spike-ins if we want to remove technical confounding factors, and would contain cell cycle genes if we want to remove cell cycle effects. We use the following partial least squares regression to jointly model both control and target genes:

$$\mathbf{x_i} = \boldsymbol{\Lambda}_x \mathbf{z_i} + \varepsilon_{xi},\ \varepsilon_{xi} \sim \mathrm{MVN}(0,\ \boldsymbol{\Psi}_{xi}) \tag{1}$$

$$\mathbf{y_i} = \boldsymbol{\Lambda}_y \mathbf{z_i} + \boldsymbol{\Lambda}_u \mathbf{u_i} + \varepsilon_{yi},\ \varepsilon_{yi} \sim \mathrm{MVN}(0,\ \boldsymbol{\Psi}_{yi}) \tag{2}$$

where for i'th individual cell, $\mathbf{x_i}$ is a q-vector of expression levels for q control genes; $\mathbf{y_i}$ is a p-vector of expression levels for p target genes; $\mathbf{z_i}$ is k_z-vector of unknown confounding factors that affect both control and target genes; the coefficients of the confounding factors are represented by the q by k_z loading matrix $\boldsymbol{\Lambda}_x$ for the control genes and the p by k_z loading matrix $\boldsymbol{\Lambda}_y$ for the target genes; $\mathbf{u_i}$ is a k_u-vector of unknown factors in the target genes and potentially represents the predictors of interest or other structured variations (see below); $\boldsymbol{\Lambda}_u$ is a p by k_u loading matrix; ε_{xi} is a q-vector of idiosyncratic error with covariance $\boldsymbol{\Psi}_{xi} = diag(\sigma_{x1}^2, \cdots, \sigma_{xq}^2)$; ε_{yi} is a p-vector of idiosyncratic error with covariance $\boldsymbol{\Psi}_{yi} = diag(\sigma_{y1}^2, \cdots, \sigma_{yp}^2)$; MVN denotes the multivariate normal distribution. We assume that ε_{xi}, ε_{yi}, $\mathbf{z_i}$, and $\mathbf{u_i}$ are all independent from each other. Following standard latent factor models, we further assume that $\mathbf{z_i} \sim \mathrm{MVN}(0,\ \mathbf{I})$ and $\mathbf{u_i} \sim \mathrm{MVN}(0, \mathbf{I})$. We model transformed data instead of the raw read counts. We also assume that the expression levels of each gene have been centered to have mean zero, which allows us to ignore the intercept.

scPLS includes two types of unknown latent factors. The first set of factors, $\mathbf{z_i}$, represents the unknown confounding factors that affect both control and target genes. The effects of $\mathbf{z_i}$ on the control and target genes are captured in the loading matrices $\boldsymbol{\Lambda}_x$ and $\boldsymbol{\Lambda}_y$, respectively. We call $\mathbf{z_i}$ the confounding factors throughout the text, and we aim to remove the confounding effects $\boldsymbol{\Lambda}_y \mathbf{z_i}$ from the target genes. The second set of factors, $\mathbf{u_i}$, aims to capture a low dimensional structure of the expression level of p target genes. The factors $\mathbf{u_i}$ can represent the unknown predictor variables of interest, specific experimental perturbations, cell subpopulations, gene signatures or other intermediate factors that coordinately regulate a set of genes. Therefore, the factors $\mathbf{u_i}$ can be interpreted as cell subtypes, treatment status, transcription factors or regulators of biological pathways in different studies[42–46]. Although $\mathbf{u_i}$ could be of direct biological interest in many data sets, we do not explicitly examine the inferred $\mathbf{u_i}$ here. Rather, we view modeling $\mathbf{u_i}$ in the target genes as a way to better capture the complex variance structure there and to facilitate the precise estimation of confounding factors $\mathbf{z_i}$. For simplicity, we call $\mathbf{u_i}$ the biological factors throughout the text, though we note that $\mathbf{u_i}$ could well represent non-biological processes such as treatment or environmental effects. Thus, the expression levels of the control genes can be described by a linear combination of the confounding factors $\mathbf{z_i}$ and residual errors; the expression levels of the target genes can be described by a linear combination of the confounding factors $\mathbf{z_i}$, the biological factors $\mathbf{u_i}$ and residual errors. For both types of confounding factors, we are interested in inferring the factor effects $\boldsymbol{\Lambda}_y \mathbf{z_i}$ and $\boldsymbol{\Lambda}_u \mathbf{u_i}$ rather than the individual factors $\mathbf{z_i}$ and $\mathbf{u_i}$. Therefore, unlike in standard factor models, we are not concerned with the identifiability of the factors. Figure 1 shows an illustration of scPLS.

scPLS is closely related to the two subcategories of unsupervised methods described in the previous Section. Specifically, without the biological effects term $\boldsymbol{\Lambda}_u \mathbf{u_i}$, scPLS effectively reduces to the first subcategory of methods that treat all genes in the same fashion for inferring the confounding factors. Without the Equation 2 term, scPLS effectively reduces to the second subcategory of methods that use only control genes for inference. (Note that, after inferring the confounding factors $\mathbf{z}_i$ from Equation 1, the second subcategory of methods still use a reduced

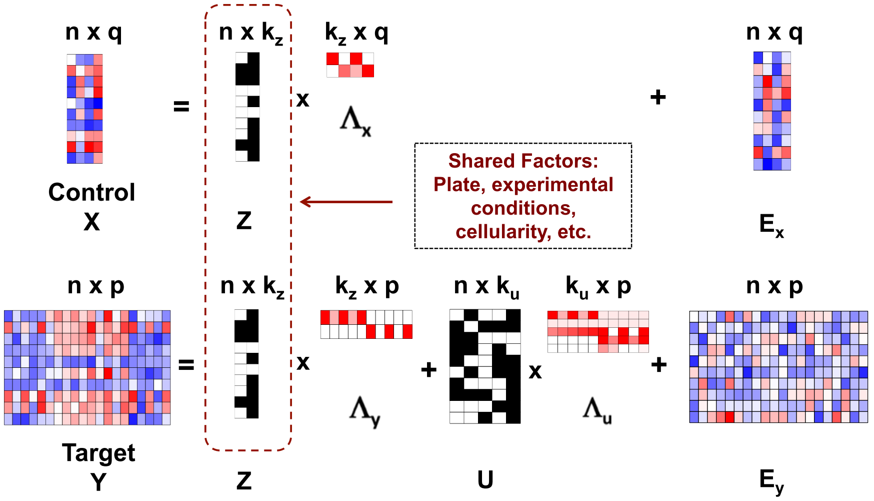

Figure 1. Illustration of scPLS. We model the expression level of genes in the control set (**X**) and genes in the target set (**Y**) jointly. Both control and target genes are affected by the common confounding factors (**Z**) with effects $\boldsymbol{\Lambda}_x$ and $\boldsymbol{\Lambda}_y$ in the two gene sets, respectively. The target genes are also influenced by biological factors (**U**) with effects $\boldsymbol{\Lambda}_u$. The biological factors represent intermediate factors that coordinately regulate a set of genes, and are introduced to better capture the complex variance structure in the target genes. $\mathbf{E_x}$ and $\mathbf{E_y}$ represent residual errors. scPLS aims to remove the confounding effects $\mathbf{Z}\boldsymbol{\Lambda}_y$ in the target genes.

version of Equation 2 without the biological effects term $\boldsymbol{\Lambda}_u\mathbf{u_i}$ to remove the confounding effects.) By including both modeling terms, scPLS can robustly control for confounding effects across a range of scenarios. Therefore, scPLS provides a flexible modeling framework that effectively includes the two subcategories of unsupervised methods as special cases and has the potential to outperform these previous methods.

Simulations. We performed a simulation study to compare scPLS with other methods. Specifically, we simulated gene expression levels for 50 control genes and 1,000 target genes for 200 cells. These 200 cells come from two equal-sized groups, representing two treatment conditions or two cell subpopulations. Among the 1,000 target genes, only 200 of them are differentially expressed (DE) between the two groups and thus represent the signature of the two groups. The effect sizes of the DE genes were simulated from a normal distribution and we scaled the effects further so that the group label explains twenty percent of phenotypic variation (PVE) in expression levels in the DE genes. In addition to the group effects, we set $k_z = 2$, $k_u = 5$ and simulated each element of $\mathbf{z_i}$ and $\mathbf{u_i}$ from a standard normal distribution. We simulated each element of Λ_x from $N(-0.25, \sigma_l^2)$ and each element of $\boldsymbol{\Lambda}_y$ from $N(0.25, \sigma_l^2)$. Note that $\boldsymbol{\Lambda}_x$ and $\boldsymbol{\Lambda}_y$ were simulated differently to capture the fact that the effect sizes of the confounding factors could be different for control and target genes. We simulated each element of $\boldsymbol{\Lambda}_u$ from $N(0, \sigma_b^2)$. We simulated each element of ε_{xi} and ε_{yi} from a standard normal distribution. We set $\sigma_l^2 = 0.4$ and $\sigma_b^2 = 0.6$ to ensure that, in non-DE genes, the confounding factors $\mathbf{z_i}$ explain 20% PVE in either the control or the target genes; the biological factors $\mathbf{u_i}$ explain 30% PVE of the target genes; and the residual errors to explain the rest of PVE. To vary signal strength in the data, we also created a series of sub data sets by varying the number of non DE genes in the data, so that the proportion of variance explained by DE genes in total equal to a fixed percentage (PDE, in the range of 20–100%, with 10% increments). After we simulated gene expression levels, we further converted these continuous values into count data by using a Poisson distribution: the final observation for ith cell and jth gene c_{ij} is from $c_{ij} \sim \mathrm{Poi}(N\,\exp(\mu + w_{ij}))$, with w_{ij} being the continuous gene expression levels simulated above and $N = 500000$, $\mu = \log(10/500000)$, which ensures an average read count of 10. Note that, because of the residual errors, the resulting count data are over-dispersed with respect to a Poisson distribution.

We considered three different simulation scenarios. In scenario I, the confounding factors $\mathbf{z_i}$ are independent of group labels. In scenario II, the confounding factors are correlated with group labels. To simulate correlated data, we simulated each element of $\mathbf{z_i}$ from $N(0, 1)$ if the corresponding sample belongs to the first group, but from $N(-0.25, 1)$ if the corresponding sample belongs to the second group. Finally, we also considered a scenario III where there is no biological factor (i.e. data were simulated effectively under the PCA modeling assumption and all genes could be used to infer the confounding factors). We performed 10 simulation replicates for each scenario. For scenario I and II, we further introduced dropout events that are commonly observed in scRNAseq data. This was done by going through one gene at a time and setting the expression level for j th gene c_{ij} to zero with probability π_{ij} that depends on the expression level through $log\frac{\pi_{ij}}{1-\pi_{ij}} = c_{ij}$.

We compared scPLS to four different methods: (1) PCA and (2) LMM (implemented in GEMMA[47,48]) use all genes to infer the confounding effects; while (3) RUVseq (version 1.2.0); which we simply refer to as RUV in the following text) and (4) scLVM (version 0.99.1) use only control genes to infer the confounding effects. We note that while some of these methods are developed not specifically for scRNAseq, these methods represent a range of strategies to deal with confounding factors. We used default settings in all the above methods. We used the count data directly for RUV and used log transformed data (i.e. $\log(c_{ij}+1)$) for all other methods. For PCA and RUV, we set the number of latent factors to be the true number (i.e. 2). Such number is determined automatically by the software itself for scLVM, and is not needed for LMM. We compared different methods based on clustering performance. In particular, for each of these methods, we obtained corrected data and applied k-means method to cluster cells into two subpopulations. We the compared the clusters inferred from the corrected data with the truth and used adjusted rand index (ARI) to measure clustering performance. ARI is computed across a range of signal strength that is measured as PDE explained above. Intuitively, if a method performs well in removing confounding factors, then the corrected data from this method can be used to better infer the two cell subpopulations and thus yields a higher ARI score.

Overall, scPLS performs the best in both scenarios I and II, with or without dropout events (Fig. 2a). The addition of dropout events in either of the two scenarios reduces the performance of all methods but does not change their relative rank of performance. The superior performance of scPLS also suggests that properly using both control and target genes can lead to effective removal of confounding effects. Among the rest of the methods, PCA and LMM performs better than RUV and scLVM, suggesting that target genes contain a substantial amount of information for removing confounding effects. Beside the comparison of clustering performance, for each gene in turn, we also used different methods to estimate the proportion of gene expression variance contributed by confounding factors. Consistent with the clustering performance comparison, we found that scPLS also yielded more accurate proportion of variance estimates (Fig. 2b).

To examine the robustness of scPLS, we applied scPLS to the same data but with a reduced number of control genes (Fig. 3a). Because scPLS does not completely rely on the information contained in the control genes, it achieves robust performance even if we only use a much smaller subset of control genes. We also examined the performance of scPLS in Scenario III where there is no biological effects (Fig. 3b) and found that scPLS performs well there. As it is often unknown whether a low-rank structural variation exists in a real data set, our simulation suggests that we can always include the biological factors $\mathbf{u_i}$ in the model even in the absence of such factors. In addition, scPLS is not sensitive with respect to the number of biological factors used in fitting the model, and achieves similar power for a range of reasonable k_u values (Fig. 3c).

Real Data Applications. Next, we applied scPLS to two real data sets. The first dataset is used to demonstrate the effectiveness of scPLS in removing the technical confounding effects by using ERCC spike-ins. Removing technical confounding effects is a common and important task in transcriptome analysis. The second dataset is used to demonstrate the effectiveness of scPLS in removing cell cycle effects by using a known set of cell cycle genes. Removing cell cycle effects can reveal gene expression heterogeneity that is otherwise obscured.

Removing Technical Confounding Factors. The first dataset consists of 251 samples from[22]. Among these, 119 are mouse embryonic stem cells (mESCs), including 74 mESCs cultured in a two-inhibitor (2i) medium and 45 mESCs cultured in a serum medium. The remaining 132 cells are control "cells" that are obtained by mixing single cells cultured in each condition (i.e. these control "cells" are similar to bulk seq data in terms of consisting a mixture of cell types, but are prepared and sequenced using single cell protocol). The control cells include 76 cells cultured in 2i and 56 cells cultured in serum. Because the control cells are homogeneous within each culture condition, when we cluster these cell, we would expect the only true cluster detectable among these cells is the culture condition. Therefore, we decide to focus on these control cells to compare the performance of different methods for removing technical effects.

We obtained the raw UMI counts data directly from the authors. The data contains measurements for 92 ERCC spike-ins and 23,459 genes. Due to the low coverage of this dataset (median coverage equals one), we filtered out lowly expressed genes and selected only genes that have at least five counts and spike-ins that have at least one count in more than a third of the cells. This filtering step resulted in a total of 32 ERCC spike-ins that were used as the controls and 2,795 genes that were used as the targets.

As in the simulations, we log transformed the count data and centered the transformed values for scPLS, PCA, LMM and scLVM. We used the count data for RUV. In this data, scPLS infers $k_z=1$ confounding factors and $k_u=1$ biological factors. In the target genes, the confounding factors and structured biological factors explain a median of 18% and 30% of gene expression variance, respectively. The PVE by the confounding and biological factors can be as high as 73.7% and 77.9%, respectively, in the target genes.

We applied scPLS and the other four methods to remove confounding effects in the data. Since control cells are homogeneous within each culture condition, we reasoned that if the method is effective in removing confounding effects, then the corrected data from the corresponding method could be used to better reveal two clusters that correspond to the two known culture conditions. For the clustering analysis, we applied the four different clustering approaches on the uncorrected or corrected data from different methods. The four clustering approaches include: (1) kmeans, where we applied the k-means method directly on the uncorrected or corrected data; (2) PCA, where we extracted the top five PCs from either the uncorrected or corrected data and then applied the k-means method using the top PCs; (3) tSNE, where we used tSNE to either the uncorrected or corrected data and then applied the k-means method on the extracted tSNE factors; (4) SC3, where we used a recently developed state-of-the-art single cell clustering method single cell census clustering (SC3)[49]. For all these clustering approaches, we set the number of clusters to two and measured clustering performance by the adjusted Rand Index (ARI). The results are shown in Table 1 and are overall consistent with the simulations. Specifically, scPLS outperforms the other methods in three out of the four clustering approaches. scPLS performs slightly worse than RUV when tSNE was used to cluster data–but tSNE works extremely poorly in this data presumably because tSNE's non-linearity assumption does not fit the data well.

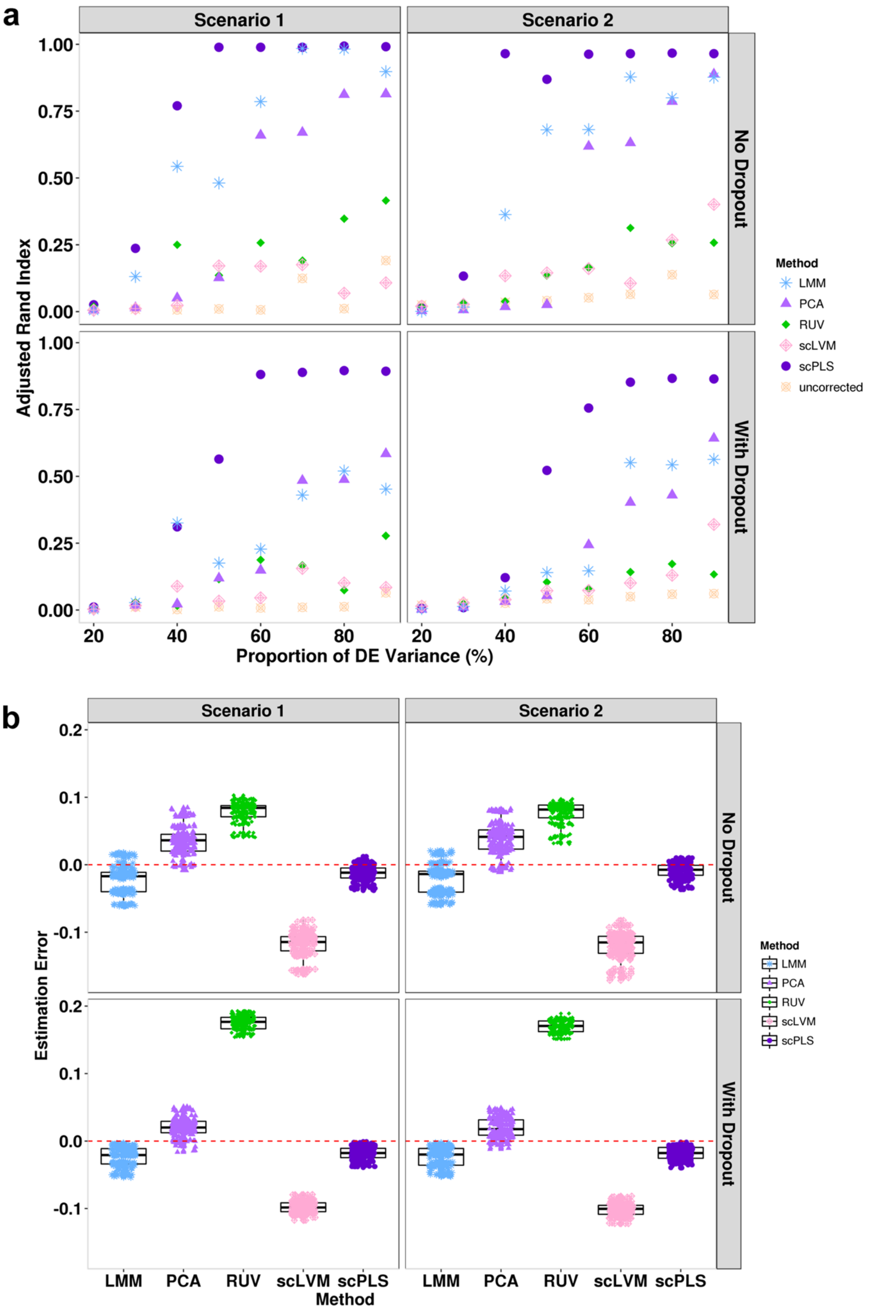

Figure 2. Method comparison in simulations. Clustering analysis using scPLS-corrected data achieves higher Adjusted Rand Index (ARI) than using LMM-, PCA-, RUV- and scLVM-corrected data or uncorrected data in both scenario I with (**a**) or without drop-out (**c**) and scenario II with (**b**) or without drop-out (**d**) across a range of signal strength. ARI is averaged across ten simulation replicates. x-axis shows the signal strength, which are measured as the percentage of DE genes variance out of all genes. (**c**) Sensitivity analysis shows that, scPLS maintains a high ARI (y-axis) when a smaller subset of control genes are used ($q = 10$, 20, 30 or 40 instead of 50).

Removing Cell Cycle Effects. Our method can also be used to remove cell cycle effects. To demonstrate its effectiveness there, we applied scPLS and several other methods to a second dataset that was used for demonstrating cell cycle influence[37]. This dataset contains gene expression measurements on 9,570 genes from 182 embryonic stem cells (ESCs) with pre-determined cell-cycle phases (G1, S and G2M). The uncorrected data we obtained are already pre-processed by the original study to remove the technical effects and are thus continuous. Therefore, we did not apply RUV here. To remove cell cycle effects, we used 629 annotated cell-cycle genes as controls and the other genes as targets. scPLS infers $k_z = 1$ cell cycle confounding factors, and $k_u = 1$ biological factors. These factors explain a median of 0.4% and 0.1% of gene expression variance, respectively. The PVE by cell cycle factors and biological factors can be as high as 7% and 2%, respectively. We visualized the uncorrected data and scPLS corrected data on a PCA plot (Fig. 4). In the uncorrected data, there is a clear separation of cells according to

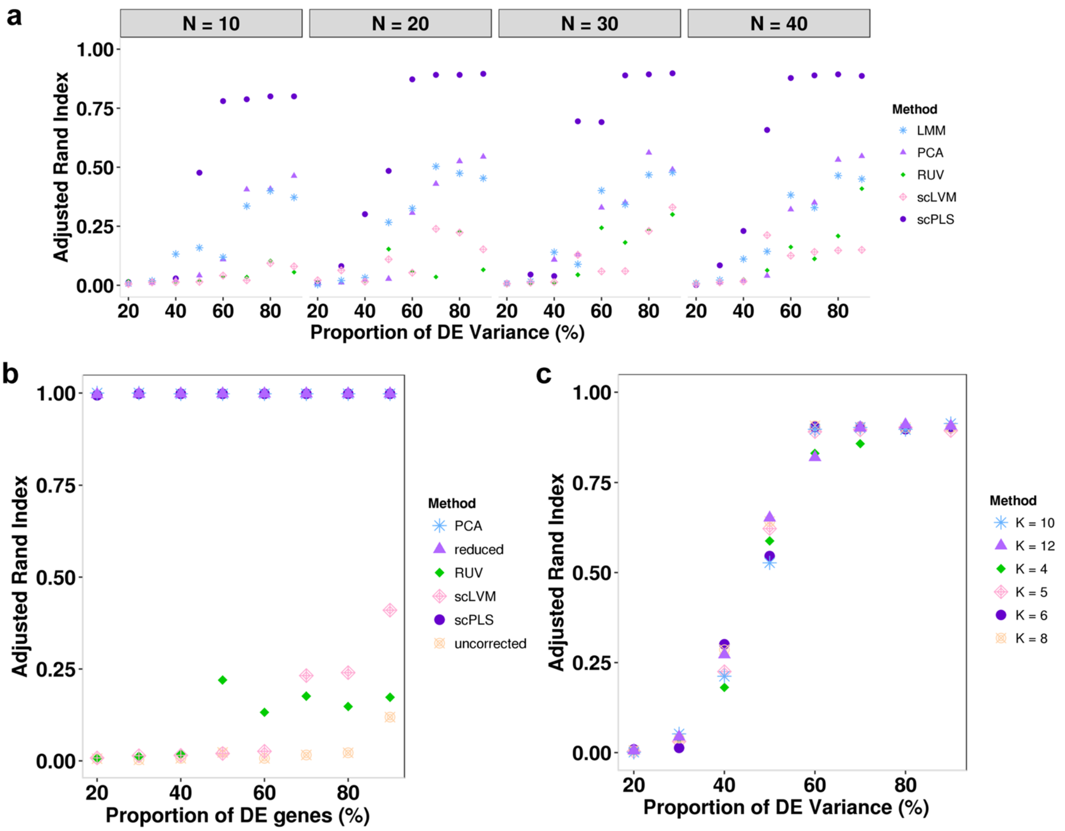

Figure 3. Method comparison in simulations (continued). (**a**) Error in estimating the proportion of variance contributed by confounding factors across genes using data corrected by different methods. Error is computed as the difference between the estimated proportion and the true proportion. (**b**) scPLS performs well in Scenario III when the model is misspecified (with true $k_u = 0$). (**c**) scPLS is robust with respect to k_u, as the ARI remains similar when a different number of biological factors is used ($k_u = 2, 4, 6, 8$) in Scenario I with dropout. x-axis shows the signal strength, which are measured as the percentage of DE genes variance out of all genes.

	uncorrected	scPLS	RUV	LMM	PCA	scLVM
kmeans	59	100	31	67	42	4
PCA	59	100	31	67	67	0
tSNE	42	44	46	44	36	0
SC3	100	100	97	91	80	2

Table 1. Clustering performance on the uncorrected data or data corrected by different methods (columns). Different clustering approaches (rows) are applied in order to examine the robustness of the comparison results. Clustering performance is measured by the adjusted Rand Index. All performance measurements are averaged across 10 runs and are multiplied by a factor of 100. The top performer is colored blue.

cell-cycle stage. Such separation of cells is not observed in the corrected data, indicating that the cell cycle related expression signature is effectively removed.

We compared scPLS and the other three methods in their effectiveness in removing cell cycle effects. Following the original study[37], we evaluated method performance based on the following criteria. Specifically, we computed for each gene the proportion of expression variance explained by the cell cycle factor. We denote this quantity as PVEi, which stands for inferred PVE. Because the cell-cycle stage of each cell had been experimentally determined in this data set, we further computed the variance explained by the true cell cycle labels. We denote this quantity as PVEt, which stands for true PVE. For scPLS, PVEi and PVEt are highly correlated ($r^2 = 0.94$), demonstrating the efficacy of scPLS. The correlation remains the same whether we use the full control set or with a subset of 300 controls. The correlation between PVEi and PVEt in scPLS is slightly higher, with statistical significance, than scLVM ($r^2 = 0.92$; p-value $< 10^{-16}$ comparing scPLS vs scLVM), LMM ($r^2 = 0.92$; p-value $< 10^{-16}$ comparing scPLS vs LMM), and PCA ($r^2 = 0.92$; p-value $< 10^{-16}$ comparing scPLS vs PCA). In addition, as an alternative measurement, the median of the absolute difference between PVEi and PVEt across genes from scPLS, scLVM, LMM and PCA are 0.018, 0.023, 0.019 and 0.019, respectively, again supporting a small advantage of scPLS. However, we do want to acknowledge that all methods work reasonably well in this data (which is

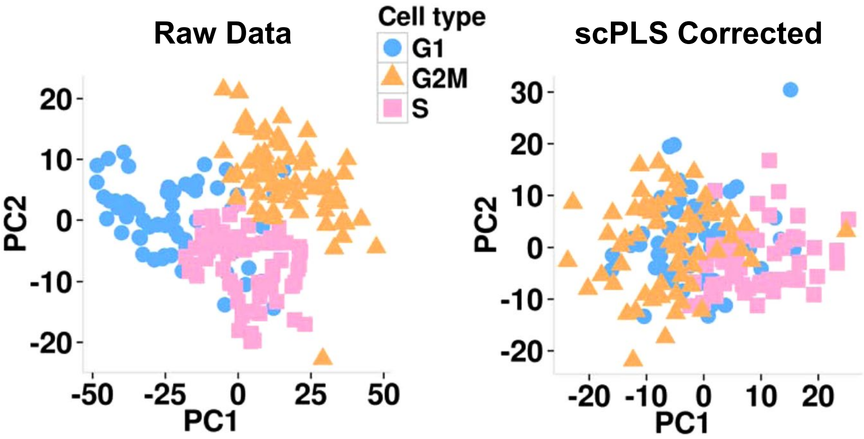

Figure 4. PCA plots for the uncorrected data and scPLS corrected data in the second dataset. In the uncorrected data, there is a clear separation of cells by cell-cycle stage. Such separation of cells is no longer observed in the scPLS corrected data.

consistent with the low variance explained by the confounding factors), suggesting that removing cell cycle effects is a relatively trivial task at least in this data set.

Discussion

We have presented scPLS for removing hidden confounding effects in scRNAseq studies. scPLS models both control and target genes jointly to infer the confounding factors and shows robust performance across a range of application scenarios. With simulations and applications to two real data sets, we have demonstrated its effectiveness for removing technical confounding effects or cell cycle effects in scRNAseq studies.

Although we have focused on its applications to scRNAseq studies, scPLS can be readily applied to other genomic sequencing studies. For instance, our method can be used to remove confounding effects from gene expression levels in bulk RNAseq studies[50] or from methylation levels in bisulfite sequencing studies[51]. The main requirement of our method is a set of pre-specified control genes that are measured together with the target genes in the sequencing studies. It is often straightforward to obtain such control genes. For example, many scRNAseq studies include a set of ERCC spike-in controls that could be used to model and remove technical confounding effects[33]. Even when such ERCC spike-in controls are not present or when they are unreliable[29], we can select a known set of house-keeping genes as controls to remove technical confounding[29]. Similarly, we can use a set of known cell cycle genes to remove cell cycle effects. Importantly, the performance of scPLS is robust to the number of genes included in the control set and yields comparable results even when a much smaller number of control genes is used. This is because scPLS not only uses information from control genes but also relies on information from target genes. Insensitivity to the control set makes scPLS especially suited to removing confounding factors in studies where a control set is not clearly defined. Because of its effectiveness and robustness, we expect scPLS to be useful in removing confounding effects in a wide variety of sequencing studies.

One important feature of scPLS is that it includes a low-rank component to model the structured biological variation often observed in real data. By decomposing the (residual) gene expression variation into a low-rank structured component that is likely to be contributed by a sparse set of biological factors, and an unstructured component that reflects the remaining variation, scPLS can better model the residual error structure for accurate inference of confounding effects. Although here we have focused on using the biological factors to better infer the confounding effects, we note that the low-rank biology factors themselves could be of direct interest. In fact, low-rank factors inferred from many data sets using standard factor models have been linked to important biological pathways or transcription factors[42–46]. Inferring the biological factors using scPLS is not feasible at the moment, however: because of model identifiability, scPLS can only be used to infer the biological effects (i.e. $\mathbf{\Lambda}_u \mathbf{u}_\mathbf{i}$) but not the biological factors (i.e. $\mathbf{u}_\mathbf{i}$). That said, additional assumptions can be made on the structure of the factors or the factor loading matrices to make factor inference possible[52]. For example, we could impose sparsity assumptions on the low-rank factors to facilitate the inference of a parsimonious set of biological factors. Exploring the use of biological factors in scPLS is an interesting avenue for future research.

We have been mainly focused on comparing the performance of different confounding effects removing methods by evaluating the clustering performance as the target downstream analysis. It has been well recognized that the choice of data normalization in scRNA-Seq is highly dependent on the specific biological question and the target downstream analysis[53]. Indeed, different downstream analysis (e.g. differential expression, lineage reconstruction, detecting allele-specific expression, spatial reconstruction etc.) can be affected differently by different choices of normalization. While evaluating the performance of various confounding effects removing methods for other downstream analysis is beyond the scope of the present study, we acknowledge that the "best" confounding effects removing method may vary depending on the question of interest. Therefore, it would be important to evaluate the performance of scPLS in other analysis settings in future studies. Nevertheless, we believe scPLS represent an important addition to the existing tools for removing confounding effects. Finally, in simulations

		Naive EM		EM-in-chunks (s = 1,000)		EM-in-chunks (s = 500)	
n	***p***	**Accuracy**	**CPU time**	**Accuracy**	**CPU time**	**Accuracy**	**CPU time**
200	2000	67.29 (5.33)	244.9 (0.35)	73.32 (6.09)	103.8 (0.06)	75.6 (6.52)	57.78 (0.03)
200	4000	135.07 (10.48)	964.39 (1.95)	144.00 (13.38)	216.25 (0.71)	148.57 (14.11)	123.06 (0.20)
400	2000	72.96 (5.58)	467.6 (0.97)	66.98 (5.15)	203.09 (1.09)	53.48 (4.61)	110.43 (0.10)
400	4000	95.5 (7.41)	1834.86 (3.5)	101.8 (9.46)	422.74 (4.84)	105.05 (9.97)	236.23 (0.48)

Table 2. Comparison of the naive EM algorithm and the EM-in-chunks algorithm in terms of accuracy and speed. The EM-in-chunks algorithm uses either a chunk size of 500 genes or a chunk size of 1,000 genes. Accuracy is measured by the estimation error of the loading matrix in terms of the normalized Frobenius norm (i.e. $\sqrt{||\Lambda_x - \hat{\Lambda}_x||_F/n}$). Because of the dimensionality of the loading matrix, the estimation error is not guaranteed to decrease with increasing sample size n. Speed is measured by CPU time in seconds for 100 iterations on an Intel Xeon E5-2670 2.6 GHz CPU. Standard deviations across 10 replicates are listed inside parenthesis. s: number of genes per chunk. n: the number of cells. p: the number of genes in the target set. The number of genes in the control set is $q = 50$ in all simulations.

we have also mainly focused on using the k-means clustering method to evaluate the clustering performance. Many other clustering methods are being developed recently, some of which are specifically targeted to single cell RNAseq studies. Those methods include RaceID[54], SCUBA[55], SNN-Cliq[56], ZIFA[57], t-SNE[4], SC3[49]; just to name a few. Because scPLS does not rely on a particular clustering method, scPLS can be paired with any clustering methods to take advantage of their benefits. Indeed, we have applied different clustering approaches to measure the performance of scPLS and other methods for removing confounding effects in the real data and obtained consistent results.

Like many other methods for scRNAseq[21] or bulk[58,59] RNAseq studies, scPLS requires a data transformation step that converts the count data into quantitative expression data. Different transformation methods can affect the interpretation of the data and are advantageous in different situations[16]. Because scPLS does not rely on a particular transformation procedure, scPLS can also be paired with any transformation methods to take advantage of their benefits. One potential disadvantage of scPLS is that it does not model raw count data directly. In bulk RNAseq studies, despite the count nature of sequencing data, it has been show that there is often a limited advantage of modeling the raw read counts directly, at least for RNAseq studies[60,61]. Statistical methods that convert and model the quantitative expression data have been shown to be robust[58,59] and most large scale bulk RNAseq studies in recent years have used transformed data instead of count data[31,62–64]. However, we note that, unlike bulk RNAseq studies, single cell RNAseq data often come with low read depth. In low read depth cases, modeling count data while accounting for over-dispersion or dropout events in single cell RNAseq studies may have added benefits[17,18]. Therefore, extending our framework to modeling count data[65,66] is another promising avenue for future research.

Methods

EM Algorithms for scPLS.

We develop an expectation-maximization (EM) algorithm for inference in scPLS. Specifically, we first initialize the factor loading matrices ($\mathbf{\Lambda}_x$, $\mathbf{\Lambda}_y$, $\mathbf{\Lambda}_u$) based on sequential single value decompositions on the gene expression matrices ($\mathbf{X} = (\mathbf{x_1}, \cdots, \mathbf{x_q})$, $\mathbf{Y} = (\mathbf{y_1}, \cdots, \mathbf{y_p})$) (Algorithm 1). Afterwards, we treat the latent factors ($\mathbf{w_i} = (\mathbf{z}_\mathbf{i}^T, \mathbf{u}_\mathbf{i}^T)^T$) as missing data, use an iterative procedure to compute the expectation of the factors conditional on each individual cell data ($\mathbf{v_i} = (\mathbf{x}_\mathbf{i}^T, \mathbf{y}_\mathbf{i}^T)^T$) in turn in the E-step, and then update the factor loading matrices $\left(\mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Lambda}_x & \mathbf{0} \\ \mathbf{\Lambda}_y & \mathbf{\Lambda}_u \end{pmatrix}^T, \ \mathbf{v_i} = \begin{pmatrix} \mathbf{z_i} \\ \mathbf{u_i} \end{pmatrix}\right)$ by merging information across all individuals in the M-step (Algorithm 2). We list the EM algorithm below, with detailed derivation provided later.

Algorithm 1: Initializer of EM algorithms for scPLS.

```
Input: Data matrices $\mathbf{X}$, $\mathbf{Y}$, and the number of latent factors $k_z$ and $k_u$.
Output: $\Lambda^{(0)}$, the initial value for $\Lambda$.
Apply SVD on $\mathbf{X}$, obtain $\mathbf{U}, \mathbf{D}, \mathbf{V}$;
Calculate $\mathbf{Z} = \mathbf{U}_{(k_u)}\mathbf{D}^{1/2}_{(k_z)}$ and standardize the elements in $\mathbf{Z}$ to have mean 0 and variance 1;
Use least squares to estimate $\Lambda_x^{(0)} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X}$ and $\Lambda_y^{(0)} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{Y}$;
Obtain the residuals of $\mathbf{X}$ after removing the confounding effects, or $\mathbf{R} = \mathbf{X} - \mathbf{Z}\Lambda_x^{(0)}$;
Similarly, apply SVD on $\mathbf{R}$, obtain $\mathbf{U}', \mathbf{D}', \mathbf{V}'$;
Calculate $\mathbf{S} = \mathbf{U}'_{(k_u)}\mathbf{D}'^{1/2}_{(k_u)}$ and standardize elements in $\mathbf{S}$ so that all elements have mean 0 and variance 1;
Use least squares to estimate $\Lambda_u^{(0)} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{R}$;
```

Algorithm 2: Naive EM algorithm for scPLS.

```
Input: Data $\mathbf{w}$.
Output: $\hat{\mathbf{v}}, \hat{\Lambda}$.
Initialize $\Lambda^{(0)}$ using Algorithm 1 ;
Initialize $\psi^{(0)} = \mathbf{I}$ ;
E step: Compute $E(\mathbf{v_i}|\mathbf{w_i})^{(t)}$ and $E(\mathbf{v_i}\mathbf{v_i}^T|\mathbf{w_i})^{(t)}$, given $\Lambda^{(t)}, \psi^{(t)}$;
M step: $(\Lambda_x^T)^{(t+1)} = \left(\sum_{i=1}^n \mathbf{x_i}(E(\mathbf{z_i}|\mathbf{w_i})^T)^{(t)}\right)\left(\sum_{i=1}^n E(\mathbf{z_i}\mathbf{z_i}^T|\mathbf{w_i})^{(t)}\right)^{-1}$;
$(\Lambda_y^T)^{(t+1)} = \left(\sum_{i=1}^n \mathbf{y_i}(E(\mathbf{z_i}|\mathbf{w_i})^T)^{(t)} - \sum_{i=1}^n (\Lambda_u^T)^{(t)} E(\mathbf{u_i}\mathbf{z_i}^T|\mathbf{w_i})^{(t)}\right)\left(\sum_{i=1}^n E(\mathbf{z_i}\mathbf{z_i}^T|\mathbf{w_i})^{(t)}\right)^{-1}$ ;
$(\Lambda_u^T)^{(t+1)} = \left(\sum_{i=1}^n \mathbf{y_i}(E(\mathbf{u_i}|\mathbf{w_i})^T)^{(t)} - \sum_{i=1}^n (\Lambda_y^T)^{(t+1)} E(\mathbf{z_i}\mathbf{u_i}^T|\mathbf{w_i})^{(t)}\right)\left(\sum_{i=1}^n E(\mathbf{u_i}\mathbf{u_i}^T|\mathbf{w_i})^{(t)}\right)^{-1}$ ;
$\Lambda^{(t+1)} = \begin{pmatrix} \Lambda_x^{(t+1)} & \mathbf{0} \\ \Lambda_y^{(t+1)} & \Lambda_u^{(t+1)} \end{pmatrix}^T$;
$\psi^{(t+1)} = \frac{1}{n}\text{diag}\{\sum_{i=1}^n (\mathbf{w_i}\mathbf{w_i}^T - (\Lambda_x^T)^{(t+1)} E(\mathbf{v_i}|\mathbf{w_i})^{(t)}\mathbf{w_i}^T)\}$ ;
Stop when $||(\Lambda^T)^{(t+1)}\Lambda^{(t+1)} - (\Lambda^T)^{(t)}\Lambda^{(t)}||_F^2$ is below a threshold;
```

We refer to the above algorithm (Algorithm 2) as the naive EM algorithm. The naive EM algorithm is computationally expensive: it scales quadratically with the number of genes and linearly with the number of cells/samples. To improve the computational speed, we develop a new EM-in-chunks algorithm (Algorithm 3). Our algorithm is based on the observation that the expression levels of the target genes are determined by the same set of underlying factors and that these factors can be estimated accurately even with a small subset set of target genes. This allows us to randomly divide target genes into dozens of chunks, compute the expectation of the factors in each chunk separately in the E-step, and then average these expectations across chunks. With the averaged expectations, we then update the factor loading matrices in the M-step. Thus, our new algorithm modifies the E-step in the naive algorithm and becomes K times faster than the naive one, where K is the number of chunks. This same idea has also been applied in the ZIFA algorithm[57]. Simulations (detailed in the simulations Section) show that our EM-in-chunks algorithm yields almost comparable results to the naive EM algorithm with respect to estimation errors, but can be close to an order of magnitude faster (Table 2). With the EM-in-chunks algorithm, our method is easily scalable to handle tens of thousands of cells (Fig. 5). For example, on a single Xeon desktop CPU, we can analyze 10,000 cells and 1,000 genes using our method in approximately 40 min. Therefore, we apply the EM-in-chunks algorithm with chunk size 500 throughout the rest of the paper.

Algorithm 3: EM-in-chunks algorithm for scPLS.

```
Input: Data $W$.
Output: $\hat{V}, \hat{\Lambda}$.
Initialize $\Lambda^{(0)}$ using Algorithm 2 ;
Initialize $\psi^{(0)} = \mathbf{I}$ ;
Initialize $E(\mathbf{v_i}|\mathbf{w_i})^{(0)}$ and $E(\mathbf{v_i}\mathbf{v_i}^T|\mathbf{w_i})^{(0)}$ using E step in Algorithm 1 ;
M step: $((\Lambda_x^k)^T)^{(t+1)} = \left(\sum_{i=1}^n \mathbf{x_i^k}(E(\mathbf{z_i}|\mathbf{w_i})^T)^{(t)}\right)\left(\sum_{i=1}^n E(\mathbf{z_i}\mathbf{z_i}^T|\mathbf{w_i})^{(t)}\right)^{-1}$;
$((\Lambda_y^k)^T)^{(t+1)} = \left(\sum_{i=1}^n \mathbf{y_i}^k(E(\mathbf{z_i}|\mathbf{w_i})^T)^{(t)} - \sum_{i=1}^n ((\Lambda_u^k)^T)^{(t)} E(\mathbf{u_i}\mathbf{z_i}^T|\mathbf{w_i})^{(t)}\right)\left(\sum_{i=1}^n E(\mathbf{z_i}\mathbf{z_i}^T|\mathbf{w_i})^{(t)}\right)^{-1}$ ;
$((\Lambda_u^k)^T)^{(t+1)} = \left(\sum_{i=1}^n \mathbf{y_i}^k(E(\mathbf{u_i}|\mathbf{w_i})^T)^{(t)} - \sum_{i=1}^n ((\Lambda_y^k)^T)^{(t+1)} E(\mathbf{z_i}\mathbf{u_i}^T|\mathbf{w_i})^{(t)}\right)\left(\sum_{i=1}^n E(\mathbf{u_i}\mathbf{u_i}^T|\mathbf{w_i})^{(t)}\right)^{-1}$ ;
$(\Lambda^k)^{(t+1)} = \begin{pmatrix} (\Lambda_x^k)^{(t+1)} & \mathbf{0} \\ (\Lambda_y^k)^{(t+1)} & (\Lambda_u^k)^{(t+1)} \end{pmatrix}^T$;
$(\psi_x^k)^{(n+1)} = \frac{1}{n}\text{diag}\{\sum_{i=1}^n (\mathbf{w_i}\mathbf{w_i}^T - ((\Lambda_x^k)^T)^{(t+1)} E(\mathbf{v_i}|\mathbf{w_i})^{(t)}\mathbf{w_i}^T)\}$ ;
E step;
for $k = 1$ to $K$ do
    Compute $E(\mathbf{z_i}^k|\mathbf{z_i}^k)$ and $E(\mathbf{z_i}^k(\mathbf{z_i}^k)^T|\mathbf{w_i}^k)$, given $\Lambda^k$, $\psi^k$;
end
Average among $K$ chunks and obtain $E(\mathbf{z_i}|\mathbf{w_i}) = \frac{1}{K}\sum_{k=1}^K E(\mathbf{z_i}^k|\mathbf{w_i}^k)$, $E(\mathbf{z_i}\mathbf{z_i}^T|\mathbf{w_i}) = \frac{1}{K}\sum_{k=1}^K E(\mathbf{z_i}^k(\mathbf{z_i}^k)^T|\mathbf{w_i}^k)$;
Iterate between M and E step until last cycle;
Given $E(\mathbf{z_i}|\mathbf{w_i})$ and $E(\mathbf{z_i}\mathbf{z_i}^T|\mathbf{w_i})$ from the last cycle, the final estimate of $\Lambda$ and $\psi$ are calculated using one M step in
  Algorithm 1 ;
```

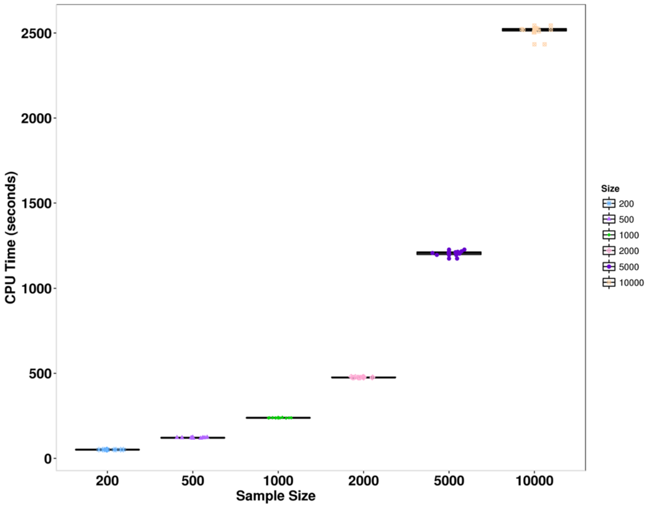

Figure 5. Computing time of the EM-in-chunks algorithm for analyzing single cell data sets of varying sample sizes. Computational time, in seconds (y-axis), were measured on data sets with a fixed number of genes (=1000) but varying number of single cells (x-axis). Ten replicates were performed for each setting on an Intel Xeon E5-2670 2.6 GHz CPU.

Finally, we use the Bayesian information criterion (BIC) to determine the number of confounding factors k_z and the number of biological factors k_u. Specifically, we evaluate the likelihood on a grid of k_z (1 to 3) and k_u values (1 to 10) and choose the optimal combination that minimizes the BIC. After estimating the model parameters on the optimal set of k_z and k_u, we use the residuals $\hat{\mathbf{y}}_{\mathbf{i}} = \mathbf{y}_{\mathbf{i}} - \hat{\mathbf{\Lambda}}_{\mathbf{y}}\hat{\mathbf{z}}_{\mathbf{i}}$ as the de-noised values for subsequent analysis. Note that the residuals are only free of the confounding effects $\mathbf{\Lambda}_y\mathbf{z}_{\mathbf{i}}$ but still contain the biological effects $\mathbf{\Lambda}_u\mathbf{u}_{\mathbf{i}}$.

EM Algorithm Derivation.

To derive the EM algorithm, we first integrate out the latent variables $\mathbf{z}_{\mathbf{i}}$ and $\mathbf{u}_{\mathbf{i}}$ and obtain

$$P(\mathbf{x}_{\mathbf{i}}|\mathbf{\Lambda}_x, \psi_x) = MVN(0, \psi_x + \mathbf{\Lambda}_x^T\mathbf{\Lambda}_x), \tag{3}$$

$$P(\mathbf{y}_{\mathbf{i}}|\mathbf{\Lambda}_y, \mathbf{\Lambda}_u, \psi_y) = MVN(0,\psi_y + \mathbf{\Lambda}_y^T\mathbf{\Lambda}_y + \mathbf{\Lambda}_u^T\mathbf{\Lambda}_u). \tag{4}$$

The latent variable $\mathbf{x}_{\mathbf{i}}$ and $\mathbf{z}_{\mathbf{i}}$ follow a joint normal distribution

$$\begin{pmatrix} \mathbf{x}_{\mathbf{i}} \\ \mathbf{z}_{\mathbf{i}} \end{pmatrix} \sim MVN\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{matrix} \psi_x + \mathbf{\Lambda}_x^T\mathbf{\Lambda}_x & \mathbf{\Lambda}_x \\ \mathbf{\Lambda}_x^T & \mathbf{I} \end{matrix}\right). \tag{5}$$

Denoting $\mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Lambda}_x & \mathbf{0} \\ \mathbf{\Lambda}_y & \mathbf{\Lambda}_u \end{pmatrix}^T$, $\mathbf{v}_{\mathbf{i}} = \begin{pmatrix} \mathbf{z}_{\mathbf{i}} \\ \mathbf{u}_{\mathbf{i}} \end{pmatrix}$, and $\psi = \begin{pmatrix} \psi_x & \mathbf{0} \\ \mathbf{0} & \psi_y \end{pmatrix}$, we can re-write $\mathbf{w}_{\mathbf{i}} = \begin{pmatrix} \mathbf{x}_{\mathbf{i}} \\ \mathbf{y}_{\mathbf{i}} \end{pmatrix}$ as $\mathbf{w}_{\mathbf{i}} = \mathbf{\Lambda}^T\mathbf{v}_{\mathbf{i}} + \psi$. The variables $\mathbf{w}_{\mathbf{i}}$ and $\mathbf{v}_{\mathbf{i}}$ then follow a joint normal distribution

$$\begin{pmatrix} \mathbf{w}_{\mathbf{i}} \\ \mathbf{v}_{\mathbf{i}} \end{pmatrix} \sim MVN\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{matrix} \begin{pmatrix} \psi_y & \mathbf{0} \\ \mathbf{0} & \psi_x \end{pmatrix} + \mathbf{\Lambda}^T\mathbf{\Lambda} & \mathbf{\Lambda} \\ \mathbf{\Lambda}^T & \mathbf{I} \end{matrix}\right). \tag{6}$$

We view the latent factors $\mathbf{v}_{\mathbf{i}}$ as the missing data. In the E step, we calculate the expectation of the log likelihood function for complete data. The expectation is taken with respect to the conditional distribution of $\mathbf{v}_{\mathbf{i}}$ given $\mathbf{w}_{\mathbf{i}}$

$$\begin{aligned} E(\log l(\mathbf{v}, \mathbf{w})|\mathbf{w}) &= -\frac{1}{2}\sum_{i=1}^{n} E[\mathbf{v}_{\mathbf{i}}^T\mathbf{\Lambda}\psi^{-1}\mathbf{\Lambda}^T\mathbf{v}_{\mathbf{i}} - 2\mathbf{v}_{\mathbf{i}}^T\mathbf{\Lambda}\psi^{-1}\mathbf{w}_{\mathbf{i}}|\mathbf{w}_{\mathbf{i}}] - \frac{n}{2}\log|\psi| - \frac{1}{2}\sum_{i=1}^{n}\mathbf{w}_{\mathbf{i}}^T\psi^{-1}\mathbf{w}_{\mathbf{i}} \\ &= -\frac{1}{2}\sum_{i=1}^{n} E[\mathrm{tr}(\mathbf{\Lambda}\psi^{-1}\mathbf{\Lambda}^T\mathbf{v}_{\mathbf{i}}\mathbf{v}_{\mathbf{i}}^T)|\mathbf{w}_{\mathbf{i}}] + \sum_{i=1}^{n} E(\mathbf{v}_{\mathbf{i}}|\mathbf{w}_{\mathbf{i}})^T \Lambda\psi^{-1}\mathbf{w}_{\mathbf{i}} - \frac{n}{2}\log|\psi| - \frac{1}{2}\sum_{i=1}^{n}\mathbf{w}_{\mathbf{i}}^T\psi^{-1}\mathbf{w}_{\mathbf{i}}. \end{aligned} \tag{7}$$

In the M step, we maximize the above expectation. To do so, we take derivatives of the log-likelihood function with respect to $\mathbf{\Lambda}_x$, $\mathbf{\Lambda}_y$ and $\mathbf{\Lambda}_u$, and obtain

$$\frac{\partial E\ \log l}{\partial \mathbf{\Lambda}_x} = \sum_{i=1}^{n}\psi_x^{-1}\mathbf{\Lambda}_x^T E(\mathbf{z}_{\mathbf{i}}\mathbf{z}_{\mathbf{i}}^T|\mathbf{w}_{\mathbf{i}}) - \sum_{i=1}^{n}\psi_x^{-1}\mathbf{x}_{\mathbf{i}}E(\mathbf{z}_{\mathbf{i}}|\mathbf{w}_{\mathbf{i}})^T, \tag{8}$$

$$\frac{\partial E\ \log l}{\partial \boldsymbol{\Lambda}_y} = \sum_{i=1}^{n} \psi_y^{-1} \boldsymbol{\Lambda}_y^T E(\mathbf{z_i}\mathbf{z_i}^T|\mathbf{w_i}) + \sum_{i=1}^{n} \psi_y^{-1} \boldsymbol{\Lambda}_u^T E(\mathbf{u_i}\mathbf{z_i}^T|\mathbf{w_i}) - \sum_{i=1}^{n} \psi_y^{-1} \mathbf{y_i} E(\mathbf{z_i}|\mathbf{w_i})^T, \tag{9}$$

$$\frac{\partial E\ \log l}{\partial \boldsymbol{\Lambda}_u} = \sum_{i=1}^{n} \psi_y^{-1} \boldsymbol{\Lambda}_u^T E(\mathbf{u_i}\mathbf{u_i}^T|\mathbf{w_i}) + \sum_{i=1}^{n} \psi_y^{-1} \boldsymbol{\Lambda}_y^T E(\mathbf{z_i}\mathbf{u_i}^T|\mathbf{w_i}) - \sum_{i=1}^{n} \psi_y^{-1} \mathbf{y_i} E(\mathbf{u_i}|\mathbf{w_i})^T, \tag{10}$$

where the conditional expectations are

$$E(\mathbf{v_i}|\mathbf{w_i}) = \boldsymbol{\Lambda}(\psi + \boldsymbol{\Lambda}^T\boldsymbol{\Lambda})^{-1}\mathbf{w_i}, \tag{11}$$

$$\mathrm{Var}(\mathbf{v_i}|\mathbf{w_i}) = \mathbf{I} - \boldsymbol{\Lambda}(\psi + \boldsymbol{\Lambda}^T\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^T \tag{12}$$

$$E(\mathbf{v_i}\mathbf{v_i}^T|\mathbf{w_i}) = \mathrm{Var}(\mathbf{v_i}|\mathbf{w_i}) + E(\mathbf{v_i}|\mathbf{w_i})E(\mathbf{v_i}|\mathbf{w_i})^T. \tag{13}$$

The above equations form the basis of our EM algorithms.

References

1. Usoskin, D. *et al.* Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing. *Nat Neurosci* **18**, 145–53, https://doi.org/10.1038/nn.3881 (2015).
2. Zeisel, A. *et al.* Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science* **347**, 1138–42, https://doi.org/10.1126/science.aaa1934 (2015).
3. Jaitin, D. A. *et al.* Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. *Science* **343**, 776–9, https://doi.org/10.1126/science.1247651 (2014).
4. Macosko, E. Z. *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161**, 1202–14, https://doi.org/10.1016/j.cell.2015.05.002 (2015).
5. Treutlein, B. *et al.* Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq. *Nature* **509**, 371–5, https://doi.org/10.1038/nature13173 (2014).
6. Tang, F. *et al.* Tracing the derivation of embryonic stem cells from the inner cell mass by single-cell rna-seq analysis. *Cell Stem Cell* **6**, 468–78, https://doi.org/10.1016/j.stem.2010.03.015 (2010).
7. Durruthy-Durruthy, R. *et al.* Reconstruction of the mouse otocyst and early neuroblast lineage at single-cell resolution. *Cell* **157**, 964–78, https://doi.org/10.1016/j.cell.2014.03.036 (2014).
8. Xue, Z. *et al.* Genetic programs in human and mouse early embryos revealed by single-cell rna sequencing. *Nature* **500**, 593–7, https://doi.org/10.1038/nature12364 (2013).
9. Achim, K. *et al.* High-throughput spatial mapping of single-cell rna-seq data to tissue of origin. *Nat Biotechnol* **33**, 503–9, https://doi.org/10.1038/nbt.3209 (2015).
10. Satija, R., Farrell, J. A., Gennert, D., Schier, A. F. & Regev, A. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* **33**, 495–502, https://doi.org/10.1038/nbt.3192 (2015).
11. Shalek, A. K. *et al.* Single-cell rna-seq reveals dynamic paracrine control of cellular variation. *Nature* **510**, 363−+; https://doi.org/10.1038/nature13437 (2014).
12. Kim, K. T. *et al.* Single-cell mrna sequencing identifies subclonal heterogeneity in anti-cancer drug responses of lung adenocarcinoma cells. *Genome Biol* **16**, 127, https://doi.org/10.1186/s13059-015-0692-3 (2015).
13. Lee, M. C. *et al.* Single-cell analyses of transcriptional heterogeneity during drug tolerance transition in cancer cells by rna sequencing. *Proc Natl Acad Sci USA* **111**, E4726–35, https://doi.org/10.1073/pnas.1404656111 (2014).
14. Borel, C. *et al.* Biased allelic expression in human primary fibroblast single cells. *Am J Hum Genet* **96**, 70–80, https://doi.org/10.1016/j.ajhg.2014.12.001 (2015).
15. Deng, Q., Ramskold, D., Reinius, B. & Sandberg, R. Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science* **343**, 193–6, https://doi.org/10.1126/science.1245316 (2014).
16. Stegle, O., Teichmann, S. A. & Marioni, J. C. Computational and analytical challenges in single-cell transcriptomics. *Nat Rev Genet* **16**, 133–45, https://doi.org/10.1038/nrg3833 (2015).
17. Vallejos, C. A., Marioni, J. C. & Richardson, S. Basics: Bayesian analysis of single-cell sequencing data. *PLoS Comput Biol* **11**, e1004333, https://doi.org/10.1371/journal.pcbi.1004333 (2015).
18. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nature Methods* **11**, 740–U184, https://doi.org/10.1038/Nmeth.2967 (2014).
19. Lun, A. T. L., Bach, K. & Marioni, J. C. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. *Genome Biology* **17**, 75 (2016).
20. Kumar, N., Singh, A. & Kulkarni, R. V. Transcriptional bursting in gene expression: Analytical results for general stochastic models. *PLoS Computational Biology* **11**, e1004292 (2015).
21. Brennecke, P. *et al.* Accounting for technical noise in single-cell rna-seq experiments. *Nature Methods* **10**, 1093–1095, https://doi.org/10.1038/Nmeth.2645 (2013).
22. Grun, D., Kester, L. & van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nat Methods* **11**, 637–40, https://doi.org/10.1038/nmeth.2930 (2014).
23. Kim, J. K., Kolodziejczyk, A. A., Illicic, T., Teichmann, S. A. & Marioni, J. C. Characterizing noise structure in single-cell rna-seq distinguishes genuine from technical stochastic allelic expression. *Nat Commun* **6**, 8687, https://doi.org/10.1038/ncomms9687 (2015).
24. Finak, G. *et al.* Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. *Genome Biol* **16**, 278, https://doi.org/10.1186/s13059-015-0844-5 (2015).
25. Reinius, B. & Sandberg, R. Random monoallelic expression of autosomal genes: stochastic transcription and allele-level regulation. *Nat Rev Genet* **16**, 653–64, https://doi.org/10.1038/nrg3888 (2015).
26. Islam, S. *et al.* Quantitative single-cell rna-seq with unique molecular identifiers. *Nat Methods* **11**, 163–6, https://doi.org/10.1038/nmeth.2772 (2014).
27. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics* **8**, 118–127 (2007).

28. Walker, W. L., Liao, I. H., Donald L. Gilbert, K. S. P. C. E. M. L. L., Brenda, W. & Sharp, F. R. Empirical bayes accomodation of batch-effects in microarray data using identical replicate reference samples: application to rna expression profiling of blood from duchenne muscular dystrophy patients. *BMC Genomics* **9**, 494 (2008).
29. Risso, D., Ngai, J., Speed, T. P. & Dudoit, S. Normalization of rna-seq data using factor analysis of control genes or samples. *Nat Biotechnol* **32**, 896–902, https://doi.org/10.1038/nbt.2931 (2014).
30. Leek, J. T. & Storey, J. D. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet* **3**, 1724–35, https://doi.org/10.1371/journal.pgen.0030161 (2007).
31. Pickrell, J. K. *et al*. Understanding mechanisms underlying human gene expression variation with rna sequencing. *Nature* **464**, 768–72, https://doi.org/10.1038/nature08872 (2010).
32. Stegle, O., Parts, L., Durbin, R. & Winn, J. A bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eqtl studies. *PLoS Comput Biol* **6**, e1000770, https://doi.org/10.1371/journal.pcbi.1000770 (2010).
33. Jiang, L. *et al*. Synthetic spike-in standards for rna-seq experiments. *Genome Res* **21**, 1543–51, https://doi.org/10.1101/gr.121095.111 (2011).
34. Kang, H. M., Ye, C. & Eskin, E. Accurate discovery of expression quantitative trait loci under confounding from spurious and genuine regulatory hotspots. *Genetics* **180**, 1909–1925 (2008).
35. Listgarten, J., Kadie, C., Schadt, E. E. & Heckerman, D. Correction for hidden confounders in the genetic analysis of gene expression. *Proc Natl Acad Sci USA* **107**, 16465–16470 (2010).
36. Jacob, L., Gagnon-Bartsch, J. A. & Speed, T. P. Correcting gene expression data when neither the unwanted variation nor the factor of interest are observed. *Biostatistics* **17**, 16–28 (2015).
37. Buettner, F. *et al*. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nat Biotechnol* **33**, 155–60, https://doi.org/10.1038/nbt.3102 (2015).
38. Sun, Y., Zhang, N. R. & Owen, A. B. Multiple hypothesis testing adjusted for latent variables, with an application to the agemap gene expression data. *Annals of Applied Statistics* **6**, 1664–1688 (2012).
39. Yang, C., Wang, L., Zhang, S. & Zhao, H. Accounting for non-genetic factors by low-rank representation and sparse regression for eqtl mapping. *Bioinformatics* **29**, 1026–1034 (2013).
40. Gagnon-Bartsch, J. A. & Speed, T. P. Using control genes to correct for unwanted variation in microarray data. *Biostatistics* **13**, 539–552 (2012).
41. Gagnon-Bartsch, J. A., Jacob, L. & Speed, T. P. Removing unwanted variation from high dimensional data with negative controls. Tech. Rep. (2013).
42. Carvalho, C. M. *et al*. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association* **103**, 1438–1456, https://doi.org/10.1198/016214508000000869 (2008).
43. Pournara, I. & Wernisch, L. Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC Bioinformatics* **8**, 61, https://doi.org/10.1186/1471-2105-8-61 (2007).
44. Lucas, J. E., Kung, H. N. & Chi, J. T. Latent factor analysis to discover pathway-associated putative segmental aneuploidies in human cancers. *PLoS Comput Biol* **6**, e1000920, https://doi.org/10.1371/journal.pcbi.1000920 (2010).
45. Blum, Y., Le Mignon, G., Lagarrigue, S. & Causeur, D. A factor model to analyze heterogeneity in gene expression. *BMC Bioinformatics* **11**, 368, https://doi.org/10.1186/1471-2105-11-368 (2010).
46. Parts, L., Stegle, O., Winn, J. & Durbin, R. Joint genetic analysis of gene expression data with inferred cellular phenotypes. *PLoS Genet* **7**, e1001276, https://doi.org/10.1371/journal.pgen.1001276 (2011).
47. Zhou, X. & Stephens, M. Genome-wide efficient mixed-model analysis for association studies. *Nat Genet* **44**, 821–4, https://doi.org/10.1038/ng.2310 (2012).
48. Zhou, X. & Stephens, M. Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nat Methods* **11**, 407–9, https://doi.org/10.1038/nmeth.2848 (2014).
49. Kiselev, V. Y. *et al*. Sc3: consensus clustering of single-cell rna-seq data. *Nature Methods* in press; (2017).
50. Tung, J., Zhou, X., Alberts, S. C., Stephens, M. & Gilad, Y. The genetic architecture of gene expression levels in wild baboons. *Elife* **4**; https://doi.org/10.7554/eLife.04729 (2015).
51. Lea, A. J., Tung, J. & Zhou, X. A flexible, efficient binomial mixed model for identifying differential dna methylation in bisulfite sequencing data. *PLoS Genet* **11**, e1005650, https://doi.org/10.1371/journal.pgen.1005650 (2015).
52. West, M. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics* **7**, 733–742 (2003).
53. McDavid, A., Finak, G. & Gottardo, R. The contribution of cell cycle to heterogeneity in single-cell rna-seq data. *Nature Biotechnology* **34**, 591–593 (2016).
54. Marco, E. *et al*. Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proc Natl Acad Sci USA* **111**, E5643–5650 (2014).
55. Grün, D. *et al*. Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature* **525**, 251–255 (2015).
56. Xu, C. & Su, Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **31**, 1974–1980 (2015).
57. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology* **16**, 241 (2015).
58. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: Precision weights unlock linear model analysis tools for rna-seq read counts. *Genome Biol* **15**, R29, https://doi.org/10.1186/gb-2014-15-2-r29 (2014).
59. Ritchie, M. E. *et al*. limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic Acids Res* **43**, e47, https://doi.org/10.1093/nar/gkv007 (2015).
60. Soneson, C. & Delorenzi, M. A comparison of methods for differential expression analysis of rna-seq data. *BMC Bioinformatics* **14**, 91, https://doi.org/10.1186/1471-2105-14-91 (2013).
61. Seyednasrollah, F., Laiho, A. & Elo, L. L. Comparison of software packages for detecting differential expression in rna-seq studies. *Brief Bioinform* **16**, 59–70, https://doi.org/10.1093/bib/bbt086 (2015).
62. Lappalainen, T. *et al*. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* **501**, 506–11, https://doi.org/10.1038/nature12531 (2013).
63. Battle, A. *et al*. Characterizing the genetic basis of transcriptome diversity through rna-sequencing of 922 individuals. *Genome Res* **24**, 14–24, https://doi.org/10.1101/gr.155192.113 (2014).
64. Montgomery, S. B. *et al*. Transcriptome genetics using second generation sequencing in a caucasian population. *Nature* **464**, 773–7, https://doi.org/10.1038/nature08903 (2010).
65. Lee, S., Chugh, P. E., Shen, H., Eberle, R. & Dittmer, D. P. Poisson factor models with applications to non-normalized microrna profiling. *Bioinformatics* **29**, 1105–11, https://doi.org/10.1093/bioinformatics/btt091 (2013).
66. Zhou, M., Hannah, L., Dunson, D. & Carin, L. Beta-negative binomial process and poisson factor analysis. *Artificial Intelligence and Statistics* **22**, 1462–1471 (2012).

Acknowledgements

This work was supported by the National Institutes of Health (NIH) grant R01GM126553 and funding from the Michigan Center for Single-Cell Genomic Data Analytics–part of the Michigan Institute for Data Science (MIDAS). We thank Dr. Dominic Grun for providing the raw read counts of the first dataset.

Author Contributions

M.C. and X.Z. conceived the idea. M.C. and X.Z. developed the method. M.C. conducted the analyses. M.C. and X.Z. wrote the paper.

Additional Information

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

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.