*Published online 28 September 2020*

*NAR Genomics and Bioinformatics, 2020, Vol. 2, No. 4* **1**
*doi: 10.1093/nargab/lqaa077*

# SIMPLEs: a single-cell RNA sequencing imputation strategy preserving gene modules and cell clusters variation

**Zhirui Hu †, Songpeng Zu † and Jun S. Liu***

Department of Statistics, Harvard University, 1 Oxford Street, Cambridge, MA 02138, USA

Received March 16, 2020; Revised August 30, 2020; Editorial Decision August 31, 2020; Accepted September 03, 2020



## ABSTRACT

**A main challenge in analyzing single-cell RNA sequencing (scRNA-seq) data is to reduce technical variations yet retain cell heterogeneity. Due to low mRNAs content per cell and molecule losses during the experiment (called 'dropout'), the gene expression matrix has a substantial amount of zero read counts. Existing imputation methods treat either each cell or each gene as independently and identically distributed, which oversimplifies the gene correlation and cell type structure. We propose a statistical model-based approach, called SIMPLEs (SIngle-cell RNA-seq iMPutation and celL clustErings), which iteratively identifies correlated gene modules and cell clusters and imputes dropouts customized for individual gene module and cell type. Simultaneously, it quantifies the uncertainty of imputation and cell clustering via multiple imputations. In simulations, SIMPLEs performed significantly better than prevailing scRNA-seq imputation methods according to various metrics. By applying SIMPLEs to several real datasets, we discovered gene modules that can further classify subtypes of cells. Our imputations successfully recovered the expression trends of marker genes in stem cell differentiation and can discover putative pathways regulating biological processes.**

## INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) technologies have been widely used for discovering subtypes of cells in the immune system (1–3), the nervous system (4–6), different diseases (7), etc., and for identifying gene modules controlling various cellular processes, such as the developmental process (8,9), or responding to different stimuli (10). A typical scRNA-seq dataset has many zero entries, which can come from two sources: the expression level below the measurement limit ('off' state) and the technical 'dropout' (11). In order to impute missing values caused by the dropout, we need to distinguish technical zeros from the true biological 'off' state. Previous methods usually pool information from similar cells to do imputation. For example, MAGIC defines a diffusion process on the affinity graph of cells for imputation (12); for each of the highly probable dropout genes, scImpute (13) imputes the dropout values in one cell by learning from the same gene in other similar cells, in which the weights of other cells are determined by the genes not severely impacted by the dropout.

Similarly, VIPER uses a sparse non-negative regression method to progressively learn the local neighborhood cells and impute the gene expression based on these cells (14). These methods often over-smooth the gene expression ignoring the cell-to-cell variations, despite the fact that a main purpose of single cell experiments is to identify biological heterogeneity of cells. Moreover, based on different gene functional groups, distances between cells can be different. The aforementioned methods define the nearby cells averaging over all the genes without considering distinctions among genes.

Different from previous methods, we model the structure of gene correlations across similar cells and allow different variability for the imputed values for each gene group. The aggregated effects across multiple correlated genes can distinguish dropouts from low expressions even if the signal to noise ratio is low for each individual gene. This additional freedom of gene-group specific imputations preserves the stochasticity of gene expressions observed in scRNA-seq data. Our method, termed as **SI**ngle-cell RNA-seq i**MP**utation and cel**L** clust**E**ring (SIMPLE), infers the probability of the dropout event for each zero entry, and imputes technical zeros while maintaining biological zeros at a low level. The imputation process depends on gene correlations within similar cell types, which is modeled by a few common gene modules, as well as the gene and cell-type specific dropout rates. Although the dropout rate can be estimated from the empirical distribution of gene expressions in the scRNA-seq, it can interfere with the estimation of the gene correlation structure, especially for lowly

*To whom correspondence should be addressed. Tel: +1 617 495 1600; Fax: +1 617 496 8057; Email: jliu@stat.harvard.edu
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

expressed genes. Bulk RNA-seq data, which reveal average gene expressions across cells and provide an extra source of information on the dropout rate per gene, can also be incorporated into SIMPLE. We name such an extension for integrating bulk RNA-seq data SIMPLE-B and refer to our toolbox including SIMPLE and SIMPLE-B as SIMPLEs.

In addition to obtaining an imputed expression matrix as previous methods usually do, SIMPLEs can output clusters of cells and gene modules that distinguish different subtypes of cells or groups of samples. Also, SIMPLEs can provide measures of uncertainty for imputed values, which can be incorporated into downstream analyses. For example, from multiple imputations, SIMPLEs can provide the posterior variance of each imputed gene expression and a consensus matrix of clustering membership of cells indicating the uncertainty of the clustering results.

Among recent methods, SCRABBLE (15) also uses the bulk RNA-seq information, but only as a constraint to the mean gene expression of the scRNA-seq data instead of a means of estimating dropout rates. Furthermore, it assumes similar gene expressions in a few cell types and does not consider correlations among the genes. SAVER (16) takes advantage of gene–gene correlations for imputation, but does not model dropouts explicitly and treats each cell independently without pooling information from closely related cells. DEsingle (17) can distinguish dropouts from biologically low expression using a zero-inflated model, but its goal is to detect differentially expressed genes. Another method scVI (18) utilizes the deep learning framework to model the generation of scRNA-seq data by involving the information of similar cells and genes, the batch information and technical effects. They found that scVI often underfits gene expressions when the cell numbers are smaller than the number of genes.

By simulating the entire dataset or adding more dropouts to a published scRNA-seq data, we demonstrate the superior performances of SIMPLEs in gene expression imputation and cell clustering compared with prevailing methods for scRNA-seq imputation. Then, we applied SIMPLEs to two real datasets: the human embryonic stem cells (hESCs) differentiation data and the mouse preimplantation embryos data. In both datasets, we discovered gene modules that can further classify subtypes of cells. Moreover, the imputed values for the marker genes by SIMPLEs align well with the developmental stages of each cell, suggesting that SIMPLEs can be used to discover gene markers that regulate the developmental process. Finally, we manifest the scalability of SIMPLEs and its robustness on different parameters using a large scale dataset of mouse immune cells from multiple tissues.

## MATERIALS AND METHODS

### Framework of SIMPLEs

Given the log-normalized data (e.g. logarithm of one plus RPKM, FPKM, or TPM), we model the gene expression within a cell type by a zero-inflated censored multivariate Gaussian distribution, denoted as $ZCN^+$ (Equation 2). If the dataset contains multiple cell types, we assume that the gene expression level across all the cells follows a mixture of $ZCN^+$ distributions. The zero component is used to model the dropout event, and each gene has its own dropout rate in each cell type. Besides random dropouts, a single cell experiment usually fails to capture low-expression genes if the sequencing depth is not enough. To model this measurement limit, we use a multivariate Gaussian distribution censored below zero for the 'amplified' gene expression. This censored Gaussian model was also used in a previous study (8) to model the gene expression in scRNA-seq. We did some model checking to show that the marginal distribution of most genes can be fitted by $ZCN^+$ distribution and the assumption of cell-type specific dropout rate per gene is reasonable (Supplementary Data).

Usually the expression level of genes involved in a common biological function are correlated in such a way that the variability of gene expression can be summarized by the variation of several gene modules. A gene module can represent a pathway such that some genes in the pathway are co-expressed if the pathway is activated. Based on these intuitions, the co-variance matrix of gene expression for each cell type can be expressed as a low-rank structure plus idiosyncratic noises as in the factor analysis. The expression of a gene module in each cell is represented by a latent factor. These gene modules can be shared among closely related cell types. Thus, we reuse the same gene modules for each cell type but the activity of each gene module can be different to allow for gene modules either unique to a cell type or shared among several cell types. Since a gene module may only contain a few genes, we posit a Laplace prior for the loading matrix such that only some of the genes have nonzero weights for a gene module. This model enables us to utilize cell clusters and gene-gene correlations to impute the dropout values and further refine cell clusters and gene modules.

The dropout rate for each gene can be estimated according to the marginal distribution in scRNA-seq. However, for low-expression genes, it is difficult to distinguish zero entries due to biological low-expression from dropout events. To avoid unduly imputation, we set an upper bound on the dropout rate *a priori*. Nevertheless, with the help of bulk RNA-seq data from similar cell population, we can estimate the dropout rate by combining the prior dropout rate and the ratio of the mean expression in the single cell dataset with that in the bulk RNA-seq. The overview of our model is shown in Figure 1. For inference, SIMPLEs employs a nested Monte Carlo EM algorithm and initializes the algorithm using only genes with fewer zero entries. After estimating the model parameters, SIMPLEs runs several MCMC iterations and outputs the mean and variance of each imputed values as well as multiply imputed expression matrices. The multiple imputations can be used to estimate the stability of clustering memberships.

### Details of the model and inference procedure

The log-normalized gene expressions of each single cell are modeled by a zero-inflated negative-censored multivariate normal distribution. Suppose there are $M$ cell types, for cell $i$ of type $C_m$, its vector of gene expressions, $\vec{y}_i = \{y_{gi},\ g \in$

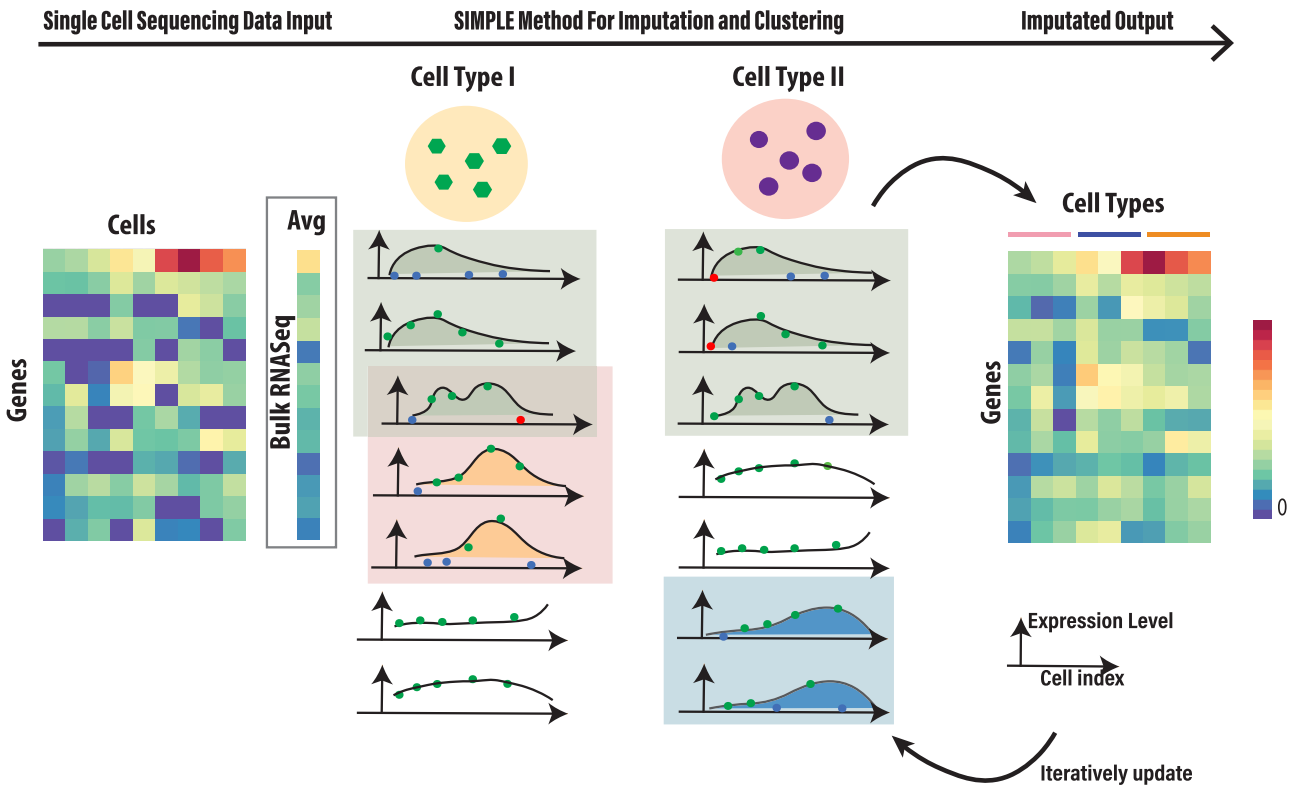


**Figure 1.** Overview of SIMPLEs. The input is a gene expression matrix from scRNA-seq and, optionally, the corresponding bulk RNA-seq data. SIMPLEs identifies the cell clusters and gene modules in which gene expressions are highly correlated. Each gene module is shown as a shaded box and the gene expression pattern across cells is represented by a curve. Each dot is the gene expression in a cell. Green dots are non-zero expressions, blue ones are dropouts and red ones are biological zeros, which represent genes having low expression in the corresponding cells. Gene module 1 is activated in both cell types but others are unique to one cell type. The expression pattern of genes shared in modules 1 and 2 show characteristics of both modules. SIMPLEs outputs the imputed matrix, cell clusters and gene modules.

$1, \ldots, G\}$, is modeled as:

$$\vec{y}_i \mid i \in C_m \sim (1 - \vec{p}_m) \circ \delta(0) + \vec{p}_m \circ \max(\vec{x}_i, 0),$$
$$\vec{x}_i \mid i \in C_m \sim \text{MVN}(\vec{\mu}_m, \Sigma_m) \quad (1)$$

where $\Sigma_m = B\Lambda_m B^T + D_m$, $D_m$ is a diagonal matrix whose diagonal entries are $\{\sigma^2_{gm},\ g = 1,\ ...,\ G\}$, and $\Lambda_m \equiv \text{diag}(\lambda_{km},\ k = 1, \ldots, K)$, for $m = 1, \ldots, M$, is a sequence of $K \times K$ diagonal matrices satisfying $\sum_{m=1}^{M} \lambda^2_{km} = 1,\ \forall 1 \leq k \leq K$. Notation '$\bigcirc$' denotes the entry-wise product, and 'MVN' stands for the multivariate Gaussian distribution. Let $X = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$ be the gene expression matrix without truncation and dropout. The purpose of imputation is to recover $X$ from $Y$. The first component of Equation (1) is a point mass at zero, which models the dropout event, and $1 - \vec{p}_m$ is the dropout rate vector for cell type $m$, where $\vec{p}_m = \{p_{gm},\ g \in 1,\ ...,\ G\}$. Each gene has its own dropout rate in each cell type. The second component, called 'amplified', models the expression amounts when the corresponding mRNA molecules are captured and subsequently sequenced in the scRNA-seq experiment. It is assumed to be a multivariate Gaussian censored below zero, denoted by $\text{CN}^+(\vec{\mu}, \Sigma)$, in which 'censoring' represents the measurement limit of low-expression genes in the single cell experiment. More precisely, we define: $\vec{Y} \sim \text{CN}^+(\vec{\mu}, \Sigma)$ if $\vec{Y} = \max(\vec{X}, 0)$, where the maximum is taken entry-wise and $\vec{X} \sim \text{MVN}(\vec{\mu}, \Sigma)$. Using this notation, we can rewrite Equation (1) as:

$$\vec{y}_i \mid i \in C_m \sim (1 - \vec{p}_m) \circ I(\vec{y}_i = 0)$$
$$+\vec{p}_m \circ \text{CN}^+(\vec{\mu}_m, \Sigma_m) \quad (2)$$

We call the distribution in Equation (2) a zero-inflated negative-censored multivariate Gaussian distribution ($\text{ZCN}^+$). $\vec{\mu}_m = \{\mu_{gm},\ g \in 1,\ ...,\ G\}$ is the vector of average expression of every gene in cell type $m$. The second component takes account of the fact that when $\mu_{gm}$ is smaller, more zero counts are observed, whereas the first component models additional zeros due to dropout for median or high mean expression level. For real applications, we set a threshold $\delta_0$ close to zero and regard the expression below $\delta_0$ as 'zero', allowing for some background noise (typically $\delta_0 = 0.1$). The co-variance matrix ($\Sigma_m$) of gene expression for each cell type can be expressed as a low-rank structure ($B\Lambda_m B^T$) plus idiosyncratic noises ($D_m$). The *loading matrix* $B = [B_{gk}]$ is a $G \times K$ matrix where $K$ is the number of gene modules or pathways. We used the same $B$ for all the cell types allowing gene modules to be shared among cell types, but this does not constrain that every cell type must have the same gene module, since we allow the activities of gene modules vary in different cell types. Without noise, each gene expression can be represented by a linear combination

of these gene modules, where each row of $B$ is the vector of weights for each gene to be mapped to the gene modules. The activity of the gene modules in a cell type is controlled by $\Lambda_m$. When gene module or pathway $k$ is unique to a single cell type, say type $m_0$, then $\lambda_{km} \neq 0$ only for $m = m_0$. Closely related cell types usually have similar gene module activities. Since the log-likelihood does not change when both $B$ and $\Lambda_m$ are scaled by reciprocal amount, we let the sum of $\lambda_{km}^2$ for each factor in different cell types be 1 (Equation 4).

To facilitate imputation and clustering, we augment the data by $z_i \in \{1, 2, .., M\}, i = 1, 2, ..., N$, indicating the cluster membership and the latent factors $f_i, i = 1, 2, ..., N$, for each cell:

$$(\vec{y}_i \mid z_i = m) \sim (1 - \vec{p}_m) \circ \delta(0) + \vec{p}_m \circ \max(\vec{x}_i, 0) \quad (3)$$

$$\begin{aligned}
(\vec{x}_i \mid z_i = m, f_i) &\sim \text{MVN}(\vec{\mu}_m + B\vec{f}_i, D_m), \;\; P(z_i = m) = \pi_m \\
\left(\vec{f}_i \mid z_i = m\right) &\sim \text{MVN}(0, \Lambda_m), \Lambda_m \\
&= \text{diag}\left(\{\lambda_{mk}^2\}_{k=1}^K\right) \text{ with } \sum_{m=1}^M \lambda_{km}^2 = 1, \forall k \\
\sigma_{gm}^2 &\sim \text{Inv-Gam}(\alpha/2, \beta/2), \;\; \mu_{gm} \sim N(0, \sigma_0^2), \;\; \pi \\
&\sim \text{Dir}(A), \;\; B_{gk} \sim \text{Laplace}(\gamma) \qquad (4)
\end{aligned}$$

We use $F$, a $K \times N$ matrix, to denote all the factors, where row $\vec{f}_{k\cdot}$ is the factor associated with gene module $B_{\cdot k}$. Each column $\vec{f}_{\cdot i}$ of $F$ represents the expression of the gene modules in cell $i$, which is i.i.d given the clustering membership. $\pi_m$ is the probability that a cell is in cluster $m$. We assume a Laplace prior for each entry of $B$ so that the weight of each gene within a gene module is sparse. Non-zero entries in each column of $B$ reflect the correlation between genes imposed by a gene module. We rank genes representing a gene module by their (non-zero) weights in the corresponding column of loading matrix $B$. We use inverse Gamma distribution as the prior for each diagonal entry of $D_m$, an Gaussian prior for each entry of the cluster mean $\mu_{gm}$, Dirichlet distribution as the prior for $\pi = (\pi_1, \pi_2, ..., \pi_M)$.

The EM algorithm (19) is often used to estimate parameters in missing data problems and for models with a latent structure. However, the E-step of the algorithm needs to compute the expectation of the log-likelihood over all the latent variables, which cannot be achieved analytically for our model. Thus, we employ the nesting Monte Carlo EM algorithm (20), which alternatively uses Gibbs sampling to impute the missing data conditioning on the current parameter values, computes the expectation over the cluster memberships ($Z = (z_1, z_2, ..., z_N)$) and factors ($F$) conditioning on the imputed data ($X$), and updates the parameters $\Theta = (B, \Lambda, D, \mu, \pi)$ by maximizing the approximated expectation of the full log-likelihood. In other words, the algorithm iteratively clusters the cells ($Z$) and updates the gene modules ($B$).

Once the EM algorithm converges, we fix all the parameters and sample $C$ copies of the imputed matrix to compute the log-likelihood and the Bayesian information criterion (BIC), respectively:

$$\begin{aligned}
LL &= E(\log P(X|\Theta)|\, Y, \Theta, \vec{p}) \approx \frac{1}{C}\sum_{c=1}^{C} \log P(X_c|\Theta) \\
&= \frac{1}{C}\sum_{c=1}^{C}\sum_{i=1}^{N} \log\left(\sum_{m=1}^{M} \pi_m \text{MVN}(\vec{x}_{ic}|\vec{\mu}_m, B\Lambda_m B^T + D_m)\right); \\
\text{BIC} &= -2 \cdot (LL + \log P(B)) + (2GM + G)\log(N) \\
&\quad + K(M-1)\log(NG) + K(G+N)\log\left(\frac{GN}{G+N}\right),
\end{aligned}$$

which are used for choosing tuning parameters, especially $K$ and $M$. The first term of BIC includes both the full log-likelihood and the log-prior probability of $B$; the second term penalizes the number parameters associated with cell clustering including ($\vec{\mu}_m$, $\vec{p}_m$, $D_m$); the third term is associated with the number of free parameters in $\Lambda$; and the last term penalizes the number of parameters associated with $B$, which is adopted from (21) for determining the number of factors in a factor model.

The algorithm is initialized with $B = 0$, implying that each gene is independent of others for cells in the same cluster, and with the dropout rate estimated independently for each gene. For genes with many zero entries, it is hard to distinguish the dropout from the amplified component based only on the marginal distribution of individual gene expression. Thus, we only use genes with a small number of zero entries ('high-quality' genes) for initial imputation. Then, we estimate factors $F$ using 'high-quality' genes, and project the expression of the remaining genes onto $F$. Here we define 'high-quality' genes as those whose percentages of zeros are lower than a threshold and also set a minimal number of genes for initialization. We found that initializing $F$ using only high-quality genes is better than using all the genes since information about the cell cluster and latent factors can be revealed from only a subset of genes (Supplementary Figure S6). Initial imputations for genes with many zeros can be quite noisy and including these genes can drive the estimation toward an undesirable local mode.

In cases without bulk RNA-seq information, we fit each gene's expression by a zero-inflated negative-censored Gaussian distribution and obtain the maximum likelihood estimator of the dropout rate per cell type, $(1 - p_{gm})$. If the dataset has cell type labels, SIMPLE can utilize the cell type information to estimate the cell type specific dropout rate; otherwise it initializes cell clustering by the $K$-means algorithm. We set an upper bound on dropout rates to prevent excessive imputation since we do not have enough information to estimate dropout rates for 'low-quality' genes. With bulk RNA-seq information, SIMPLE-B can estimate the dropout rates more accurately. Assuming that bulk RNA-seq measures the mean expression level for each gene without dropout, we can estimate the dropout rate for each gene by a linear combination of the prior dropout rate $(1 - p_0)$ and the ratio of the mean expressions of the scRNA-seq ($m_g$) and the bulk RNA-seq ($m_g^B$), provided that the scRNA-seq and the bulk RNA-seq are normalized similarly, i.e. $p_g = \frac{p_0 + m_g}{1 + m_g^B}$.

Including a prior dropout rate can alleviate the instability of the ratio when the gene expression value in the bulk

RNA-seq is too low. We use $1 - p_0$ to denote the upper bound or the prior of the dropout rate, which can be estimated experimentally (22) or empirically from the dataset (see 'Implementation details of different imputation methods' section). To account for possibly different scales, we assume that a gene's expression in bulk RNA-seq is a linear transformation of its mean expression in the scRNA-seq. Since highly expressed genes are unlikely to be dropped out, we conducted a weighted least squares method to estimate the scaling factor with weights proportional to the square of the mean expression value. If multiple bulk RNA-seq data for different cell types are available, SIMPLE-B will estimate the gene-specific dropout rate per cell type. For more details, see Supplementary Data.

### Cell clustering and identifying marker genes

Although our method can output the cell clusters directly, those previously published imputation methods we compared with do not provide any clustering result. As a fair comparison, we used *K*-means clustering after projecting the imputed or the original unimputed data onto the space spanned by the top*l* principal components, e.g. $l = 20$, of the data matrix for all methods. Since no existing imputation methods provide an explicit procedure to select the number of clusters $M$, we used the true number of clusters when applying *K*-means clustering to the outputs obtained from all imputation methods. We also applied the true number of cell types to methods that require the number of cell types as input, i.e. SIMPLEs and scImpute, if the number of cell types is available (see Supplementary Data for details of each dataset). If we have multiple imputed expression matrices (with nearly independent imputations), we apply the aforementioned clustering procedure (i.e. the *K*-means) for each imputed expression matrix and record the frequency that each pair of cell is in the same cluster, forming a co-clustering consensus matrix (23) based on which we assign an uncertainty score for each cell (Supplementary Data).

If the cell labels are ambiguous, we ran SIMPLEs for different number of clusters and selected the one with the smallest BIC. The imputed gene expression can be used for clustering extraneously and identifying differentially expressed genes. We showed in simulations that the imputation result is similar with different choices of $M$ (Supplementary Figure S19), thus alleviating the need of estimating the number of clusters precisely.

To identify marker genes for each cell cluster, we applied Student's t-test in simulations and the Wilcoxon rank-sum test for the real datasets, comparing gene expression in one cell cluster with the rest. To separate from clustering error, we used the true labels of the cells for identifying cluster-specific genes. Then, we tested the genes with fold change $>1.2$ and ranked the genes by the *P*-values output from either *t*-test or Wilcoxon test. To compute AUC, we compared the test statistics to the truly differentially expressed genes for each cluster, and obtained the average AUC. For simulations, we know the true markers for each cluster. For the hESC cell types dataset, we identified differentially expressed genes for each of the seven cell types using R package DESeq2 (24) from bulk RNA-seq and treated the genes with FDR $<1e-6$ and fold change $>1.4$ as the true markers. The number of cell type-specific markers varies from 2000 to 4000 for the seven cell types. For the *sc_10x_5cl* dataset, we identified differentially expressed genes from a matched bulk RNA-seq data (GEO number: GSE86337 (25)) using R package DESeq2 as the 'gold standard' for evaluating different imputation methods. Genes are defined as differentially expressed if the fold change is no $<2.0$ and FDR $< 0.05$.

### Simulation procedures

For the first two experiments, we set the number of genes $G = 1000$ and the total number of cells $N = 300$ divided into three clusters evenly. We simulated the mean expression of each gene by $\mu_g \sim$ log-Normal (0.5, 0.5). Then, we randomly selected 20 cluster-specific genes for each cluster, and changed their mean expressions in the corresponding cluster. The fold change for the mean expressions was sampled uniformly from $\{0.2, 0.5, 1.2, 1.5, 2\}$ to make the overall mean expression of cluster-specific genes similar to others. After that, we sampled the gene expression in each cell from a multivariate normal distribution with the specified mean and covariance matrix. Finally, we added dropout to each gene by sampling from the Bernoulli distribution with dropout rate $e^{-0.1 \cdot m_g^2}$ or $e^{-0.3 \cdot m_g^2}$, corresponding to high or low dropout rate scenarios, where $m_g$ is the overall mean expression for each gene. The mean dropout rate of every gene was about 0.7 or 0.4 for high and low dropout rate scenario respectively. We simulated bulk RNA-seq data by taking the mean expression of each gene, i.e. $m_g$.

For simulating independent gene expressions within each cell cluster, we sampled the gene expressions in cell cluster $m$ i.i.d. from $N(\mu_{gm}, \sigma_g^2)$ , where $\sigma_g \sim$ Gamma(2, 0.3). To simulate correlated gene expressions, we constructed the covariance matrix as a low rank matrix plus a diagonal matrix, i.e. $\Sigma = BB^T + \Sigma_0$, where $\Sigma_0$ is a diagonal matrix with the squared root of each diagonal entry drawn independently from Gamma(2, 0.3), $B$ is a $G \times K$ matrix, and $K$ is the number of factors (or gene modules). We considered two scenarios: a large number of small gene modules, where we set $K = 6$ with 32 genes in each module with no genes shared by two modules; and a small number of large gene modules with overlapped genes, where we set $K = 3$ with 100 genes in each module, totaling 190 distinct genes with about 50% genes belonging to multiple modules. Each entry of $B$ is either 1/4 or 0. In addition, we simulated about 800 independently expressed genes so that we had $G = 1000$ in total (details in Supplementary Data). The correlation between genes is stronger in the second scenario than that in the first one. For simulation with correlated gene expression, we only considered the low dropout rate scenario.

Besides the simulations above, we also generated both UMI and non-UMI based scRNA-seq data using a popular tool SymSim (26). SymSim simulates the entire experimental procedures of scRNA-seq by kinetic modeling of RNA transcription in different cells and polymerase chain reaction amplification in either UMI or non-UMI experimental protocols. It also takes into account batch effects. We set the number of genes $G = 1000$ and the numbers of cells $N = 300$, which is divided randomly into five distinct cell populations with at least 50 cells in each population.

We set the dimension of kinetic parameters for RNA transcription to be 10, and varied five of them among genes. For sequencing depth and RNA capture efficiency, we used the default parameters provided by SymSim: the mean and variance of the sequencing depth for the UMI-based protocol are 45 000 and 4500, respectively, and the mean RNA capture efficiency is 0.1; the mean and variance for the non-UMI protocol are 100 000 and 10 000, respectively, and the mean RNA capture efficiency is 0.4. The simulations were repeated 20 times.

Finally, to imitate the gene expression pattern from real scRNA-seq dataset, we added dropouts to the hESC cell types dataset, by either randomly set 20 or 40% entries to zero uniformly for all genes or set the probability of dropout decreasing with the mean expression of each gene, e.g. $p_g = e^{-0.3 \cdot m_g^2}$. In this simulation, we also subsampled 50 or 75% of all the cells, which correspond to 509 and 763 cells, respectively, and randomly selected 3000 genes that were expressed in more than 50 cells for each experiment to reduce the computational cost and test the stability of different methods.

### Implementation details of different imputation methods

The results shown in the main text were obtained by setting the upper bound or the prior of dropout rate to be $1 - p_0 = 0.6$ in SIMPLEs and scImpute unless otherwise stated. For simulations based on the hESC cell types data, the results shown in the main text were obtained by setting $1 - p_0 = 0.3$ when we added 20% more dropouts and $1 - p_0 = 0.5$ for other scenarios. For real applications, we set $1 - p_0 = 0.2$ as default. The parameter in the Laplace prior of the loading matrix ($\gamma$) was set to be 1 unless otherwise stated. We set $\delta_0 = 0.1$ for SIMPLEs in all the experiments. These parameters are typical choices for SIMPLEs and we did not tune these parameters further. Results varying other parameters in the simulations are shown in Supplementary Figures S3–6.

For real data applications, we recommended choosing genes with their fraction of zero entries <50% but keeping a minimum of 2000 genes for initialization. For the prior of dropout rate, we adopted the empirical Bayes approach, which first estimates the dropout rate per gene based on the marginal distribution for SIMPLE and bulk RNA-seq for SIMPLE-B, and then set $1 - p_0$ as the 75% quantile of the dropout rates across all the genes. For choosing the penalization parameter ($\gamma$) for weights in the loading matrix and the number of factors ($K$), we suggest to either use the BIC output by SIMPLEs or add some 'fake' dropouts to the dataset and find the optimal parameters to recover the original entries as we did in the simulations based on the hESC cell types dataset. Typically, the penalization parameter should be small when the number of factors is small. Also, if the dropout rate is high, SIMPLEs prefer relatively large gene modules so that more correlated genes can be incorporated for imputation. Thus, we recommend choosing a smaller penalization parameter and fewer number of factors when the data is more sparse. For other imputation methods, we used the default parameters for all the datasets. The posterior mean output from scVI was used as the imputed matrix for downstream analyses. The DDRTree algorithm from Monocle 2 was used to order the cells in the hESC time course dataset and obtain the pseudo times.

### Datasets

The hESC differentiation (GEO series number: GSE75748) and the mouse preimplantation embryos datasets (GEO series number: GSE45719) were downloaded from https://hemberg-lab.github.io/scRNA.seq.datasets. For the mouse immune cells dataset (GEO series number: GSE109774), the FACS scRNA-seq data was downloaded from https://s3.amazonaws.com/czbiohub-tabula-muris/TM_facs_mat.rds. The cell annotation was obtained from https://raw.githubusercontent.com/czbiohub/tabulamuris-vignettes/master/data/TM_facs_metadata.csv, which contains the cell ontology class for each cell. The cell ontology data were downloaded from https://raw.githubusercontent.com/obophenotype/cell-ontology/master/cl-basic.obo. Finally, the *sc_10x_5cl* data (GEO series number: GSM3618014) was obtained from the CellBench datasets (27).

### Data preprocessing

For non-UMI based scRNA-seq and bulk RNA-seq datasets, we normalized the read count by the total number of read counts in each cell and scaled by a factor of 10e6. For the hESC datasets, we selected genes with nonzero entries in at least 10% cells. After log-normalization and further filtering, we included 8148 genes with the standard deviation >1.2 in the hESC cell types dataset and 5135 genes with the standard deviation >1.5 in the hESC time course dataset. For the mouse embryos dataset, we filtered out genes with fractions of zero entries <5%, which are more likely housekeeping genes that are expressed in most cells, or >60%, and used the remaining 8648 genes for further analysis.

The mouse immune cells dataset was obtained from a collection of scRNA-seq experiments of more than 100 000 cells from 20 mouse organs and tissues (28). We extracted all the immune cells from FACS-based scRNA-seq data for our analysis. The cell annotation was obtained from the original study, which contains the cell ontology class for each cell. We collected cells with cell ontology ID CL:0000738 or its descendants from the cell hierarchy (29). According to the data preprocessing procedure in (28), we excluded cells with fewer than 500 detected genes or larger than 20 000 detected genes, or cells with fewer than 50 000 reads or larger than $2 \times 10^6$ reads. Moreover, we filtered out genes in the immune cells if they are only expressed in fewer than 5% cells or the standard deviations of the log-normalized values are less than 1.2. Finally, 7809 genes and 12 905 cells are kept in our analysis. These cells are from 12 different tissues, and can be classified into 22 immune cell types.

The *sc_10x_5cl* dataset from CellBench (27) is based on 10× Genomics platform, which contains five cell lines and 3918 single cells. We normalized the read counts by the total number of read counts in each cell and rescaled by a factor of 10 000 as in the common procedure for UMI-based protocols. After filtering out genes expressed in fewer than 10% of the cells and mitochondrial genes, we obtained 9071 genes for further analysis.

## RESULTS

### Simulations

In the first experiment, we simulated data following the procedure in Li and Li (13), in which the gene expression was sampled from mixtures of normal distributions and each gene was independent within each cluster of cells (see 'Materials and Methods' section). We simulated three cell clusters and randomly selected 20 cluster-specific genes for each cluster that are only differentially expressed in one of the cluster. Then, we randomly added dropouts to each gene by sampling from a Bernoulli distribution with gene specific dropout rate. We considered both high and low dropout rate scenarios in which dropout rates decay either slowly or quickly along with mean expression levels, respectively. The average dropout rate per gene is about 0.7 or 0.4 for the high and the low dropout rate scenario, respectively. The bulk RNA-seq data were simulated as the mean expression of each gene.

We compared SIMPLEs with MAGIC, scImpute, VIPER, SAVER and SCRABBLE, using the original unimputed data as a baseline. To compare the performance, we measured the clustering performance after imputation using the adjusted rand index (aRI); the mean-squared error (MSE) of imputed values compared with the truth; the ability of calling cluster-specific genes from the imputed data matrix using Area Under the ROC Curve (AUC). A larger aRI means that the clustering result is closer to the true labels, and aRI is 1 when the true labels are completely recovered. From Table 1, SIMPLE-B performed better than others; SCRABBLE and SIMPLE did the second best in all aforementioned performance evaluations. When the dropout rate was high, all methods except SCRABBLE and SIMPLEs were not able to identify the true cell types, resulting in poor aRI values. VIPER was not able to recover the true gene expression and its MSE between the imputed and true values was even worse than the unimputed data. The imputed value of VIPER was often much smaller than the true value. VIPER also had the longest running time among all the methods.

SIMPLE-B, which incorporates the simulated bulk RNA-seq information, was consistently better than SIMPLE, reflecting that the bulk RNA-seq provides substantial information about the dropout rate that could not be reliably identified especially for lowly expressed genes. Figure 2 shows the projections of a few simulated low and high dropout datasets onto two-dimensional spaces, based on either the original unimputed data or the imputed data using different methods. SIMPLEs recovered the cell types even when cell clusters were indiscernible because of high dropout rate (Figure 2F and H). Moreover, the performances of SIMPLE-B were not sensitive to the number of factors chosen in the model (Supplementary Figure S3); while the clustering performance of SIMPLE was worse with too large K. In this simulation, the true number of factors should be zero but the results shown were from a misspecified model where the number of factors was set at 2. The clustering performances of SIMPLEs were not sensitive to other parameters but the performances for imputation and identifying the true marker genes deteriorated

**Table 1.** Performance of simulations with no gene correlation

| | No impute | SIMPLE | SIMPLE-B | scImpute | MAGIC | SCRABBLE | VIPER | SAVER |
|---|---|---|---|---|---|---|---|---|
| *Low dropout rate* | | | | | | | | |
| **aRI** | 0.88 ± 0.03 | **1.00** ± 0.00 | **1.00** ± 0.00 | 0.93 ± 0.02 | 0.85 ± 0.05 | 0.99 ± 0.00 | 0.96 ± 0.02 | 0.95 ± 0.02 |
| **MSE** | 2.52 ± 0.02 | 0.86 ± 0.02 | **0.78** ± 0.02 | 1.04 ± 0.01 | 1.08 ± 0.01 | 0.81 ± 0.01 | 1.87 ± 0.03 | 1.71 ± 0.02 |
| **AUC** | 0.91 ± 0.01 | 0.95 ± 0.01 | **0.96** ± 0.01 | 0.93 ± 0.01 | 0.59 ± 0.01 | 0.94 ± 0.01 | 0.90 ± 0.01 | 0.92 ± 0.01 |
| *High dropout rate* | | | | | | | | |
| **aRI** | 0.06 ± 0.02 | 0.72 ± 0.08 | **0.90** ± 0.04 | 0.42 ± 0.07 | 0.33 ± 0.06 | 0.70 ± 0.06 | 0.06 ± 0.03 | 0.07 ± 0.02 |
| **MSE** | 3.5 ± 0.02 | 1.39 ± 0.02 | **0.97** ± 0.01 | 1.58 ± 0.01 | 1.82 ± 0.01 | 1.21 ± 0.01 | 4.45 ± 0.05 | 2.84 ± 0.02 |
| **AUC** | 0.83 ± 0.01 | 0.86 ± 0.01 | **0.88** ± 0.01 | 0.83 ± 0.01 | 0.57 ± 0.01 | 0.87 ± 0.01 | 0.78 ± 0.01 | 0.84 ± 0.01 |

The numbers shown are mean ± SD over 10 repetitions. Upper panel: low dropout rate; lower panel: high dropout rate. The best scores are labeled in bold.

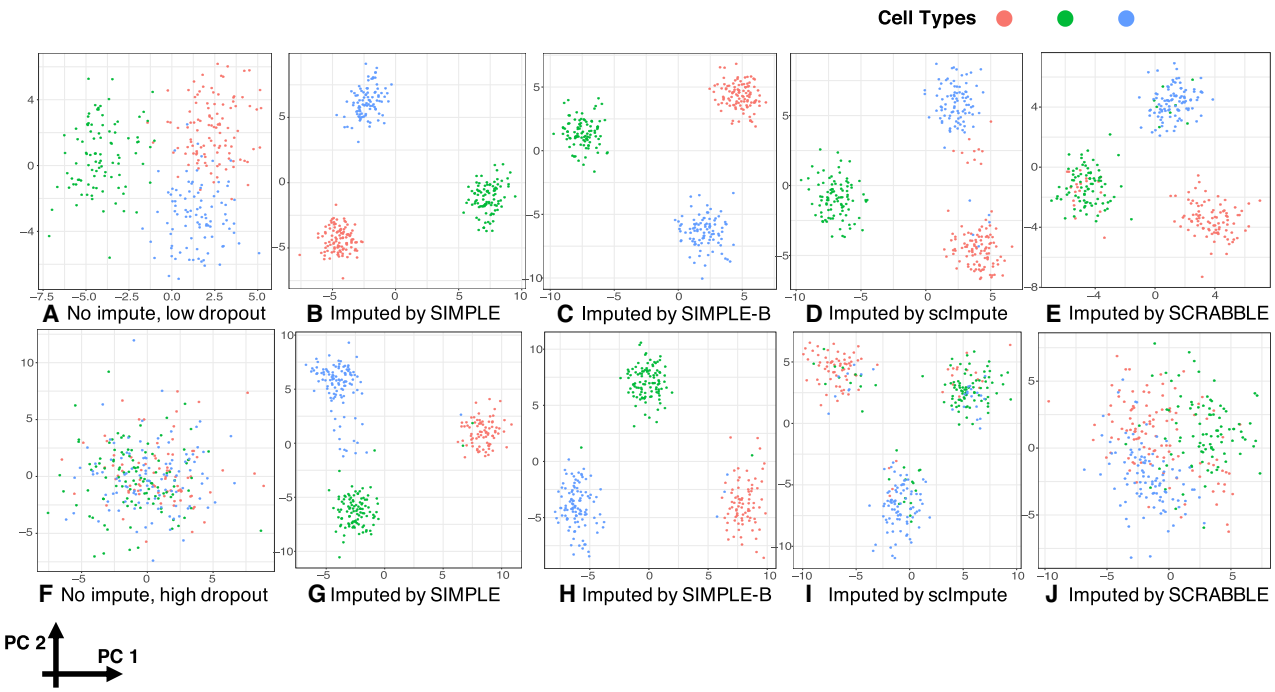


**Figure 2.** Examples of a simulation dataset. We projected both the original unimputed data and the imputed data matrix onto the space spanned by the first two principal components (represented by the *X* and *Y*-axes, respectively). Each point is a cell colored by its true cell-type color. (**A**–**E**): low dropout rate scenarios, and aRI = 0.85, 1, 1, 0.86 and 1 for respective methods. (**F**–**J**): high dropout rate scenarios, and aRI = 0.03, 0.81, 0.95, 0.30 and 0.80 for the respective methods. Other methods that were tested performed worse (Table 1).

if the prior upper bound for the dropout rate in SIMPLE was set much smaller than the true dropout rate (Supplementary Figure S3). The impact of the dropout rate prior in SIMPLE-B was much lessened because of incorporating bulk RNA-seq.

In the second experiment, we simulated data with correlated gene expression in each cluster of cells. The mean gene expression for each cluster was simulated in the same way as in the first experiment; the covariance matrix was the summation of two matrices: one for modeling the genes' correlations within each module, and the other, a diagonal matrix, for modeling the idiosyncratic noise for each gene (see 'Materials and Methods' section). We considered two scenarios: (i) a large number of small sized gene modules with no overlapping genes; (ii) a small number of large-sized modules with some genes shared by a pair of modules. In addition, we simulated independently expressed genes so that the total number of genes was 1000, the same as in the previous simulation. Genes are positive correlated only if they are in the same module; otherwise, their correlation is zero. To focus on the effect of correlation between genes, the dropout rate was simulated the same as the low dropout rate scenario in the previous experiment (more details in 'Materials and Methods' section). Clustering and identifying marker genes are more difficult in the second scenario, as the within-cluster variations of gene expression are larger but the mean differences between clusters stay the same. However, large gene modules are beneficial for imputation using SIMPLEs since the method can incorporate correlated genes for imputing the missing entries. Besides the evaluation metrics proposed in the previous simulation, we also compared the gene–gene correlation estimated from the imputed data to the true correlation. Estimating correlations are of interest in some applications, such as constructing gene regulatory networks. To separate from the clustering performance, we only considered gene correlation within each simulated 'true' cell cluster. The positive correlations between genes within a module and zero correlations in different modules were evaluated separately. Table 2 shows the MSEs between the estimated correlation from imputed data and the true positive or zero correlations, denoted as 'cor1' and 'cor0', respectively.

Most methods did well in clustering and identifying marker genes in this experiment since the dropout rate was relatively low (Table 2). MAGIC and scImpute were not as good as other methods in the large gene modules scenario because the strong correlations among the genes overwhelm the differences between clusters. SIMPLE-B had the smallest MSE of the imputed values and SCRABBLE performed the second best. Comparing large and small gene modules scenarios, SIMPLE-B and SIMPLE had smaller MSEs in the scenario with large gene modules, where more correlated genes can be used for imputation. Imputation MSEs of other methods were similar in these two scenarios. For estimating correlations within a cluster, SIMPLE was better than others even without incorporating bulk RNA-seq. Because of random dropout, the sample gene-gene correlation matrix is usually smaller than the true one. SIMPLEs recovered the gene modules and restored the correlations between genes, but other methods often imputed gene expression close to the cluster mean and underestimated the non-zero within cluster gene correlations. MAGIC performed

**Table 2.** Performance of simulation with gene correlation

| | No impute | SIMPLE | SIMPLE-B | scImpute | MAGIC | SCRABBLE | VIPER | SAVER |
|---|---|---|---|---|---|---|---|---|
| | | | | *Small gene modules* | | | | |
| **aRI** | 0.87 ± 0.04 | **1.00** ± 0.00 | **1.00** ± 0.00 | 0.93 ± 0.04 | 0.82 ± 0.06 | **1.00** ± 0.00 | 0.95 ± 0.02 | 0.95 ± 0.02 |
| **MSE** | 2.53 ± 0.02 | 0.89 ± 0.02 | **0.81** ± 0.01 | 1.05 ± 0.01 | 1.10 ± 0.01 | 0.82 ± 0.01 | 1.89 ± 0.03 | 1.73 ± 0.01 |
| **AUC** | 0.92 ± 0.00 | 0.95 ± 0.01 | **0.95** ± 0.00 | 0.94 ± 0.01 | 0.70 ± 0.02 | 0.95 ± 0.01 | 0.90 ± 0.01 | 0.92 ± 0.00 |
| **cor0** | **0.010** ± 0.000 | 0.011 ± 0.000 | 0.011 ± 0.000 | 0.012 ± 0.000 | 0.781 ± 0.012 | **0.010** ± 0.000 | 0.011 ± 0.000 | **0.010** ± 0.000 |
| **cor1** | 0.072 ± 0.003 | 0.034 ± 0.002 | **0.025** ± 0.001 | 0.052 ± 0.002 | 0.813 ± 0.012 | 0.043 ± 0.002 | 0.059 ± 0.003 | 0.067 ± 0.003 |
| | | | | *Large gene modules* | | | | |
| **aRI** | 0.80 ± 0.08 | **1.00** ± 0.00 | **1.00** ± 0.00 | 0.87 ± 0.05 | 0.78 ± 0.07 | 0.99 ± 0.01 | 0.96 ± 0.06 | 0.93 ± 0.1 |
| **MSE** | 2.54 ± 0.02 | 0.86 ± 0.02 | **0.75** ± 0.02 | 1.06 ± 0.01 | 1.10 ± 0.01 | 0.83 ± 0.01 | 1.89 ± 0.03 | 1.74 ± 0.01 |
| **AUC** | 0.91 ± 0.01 | **0.95** ± 0.01 | **0.95** ± 0.01 | 0.93 ± 0.01 | 0.70 ± 0.02 | 0.94 ± 0.01 | 0.90 ± 0.01 | 0.92 ± 0.01 |
| **cor0** | **0.010** ± 0.000 | 0.011 ± 0.000 | 0.011 ± 0.000 | 0.012 ± 0.000 | 0.779 ± 0.026 | **0.010** ± 0.000 | 0.011 ± 0.000 | **0.010** ± 0.000 |
| **cor1** | 0.068 ± 0.003 | 0.021 ± 0.001 | **0.019** ± 0.001 | 0.046 ± 0.002 | 0.813 ± 0.039 | 0.038 ± 0.002 | 0.054 ± 0.003 | 0.061 ± 0.002 |

mean ± standard error over 10 repetitions. Upper panel: small gene modules and $K = 6$; lower panel: larger gene modules and $K = 3$. The best scores are labeled in bold.

much worse than other methods. Due to its forcing the imputed gene expressions to follow a common trend, MAGIC substantially overestimated gene correlations. As a consequence, MAGIC also performed poorly in cell clustering and identifying differentially expressed genes.

Moreover, we computed the BIC for different $M$'s and $K$'s and selected the pair that gave us the minimum. When the total number of cells is large enough (e.g. 900 in this simulation), the minimal BIC was obtained at the true values of $K$ and $M$. The differences of BICs are larger for different $M$'s compared to $K$'s, and the BIC can select the correct $M$ no matter what $K$ is (Supplementary Figure S7). However, when the total number of cells is small (e.g. 300 in this simulation setting), there is no guarantee that BIC can select the true model. For this example, the BIC can still select the true $M$ when the true $K$ is 6; however, the BIC tends to select a smaller $K$ and $M$ in other scenarios. We varied the tuning parameters of SIMPLEs for this experiment and observed similar results as in the first experiment (Supplementary Figure S4). The performances were reported under the true $K$ in Table 2, but they remained almost the same for other $K$s.

In the third experiment, we generated both the UMI-based and non-UMI scRNA-seq datasets using SymSim (26) (more details in the 'Materials and Methods' section). We compared SIMPLEs with other methods on cell clustering and detecting differentially expressed genes after imputations. The clustering is challenging in this experiment. As shown in the t-SNE plot using simulated true gene expressions, some of the cell types are in close proximity to each other (Supplementary Figure S17). None of the imputation methods obtained perfect clustering results. In non-UMI based experiments, only SIMPLE outperformed the control, which used the original unimputed data for cell clustering (Supplementary Figure S18a). In UMI-based experiments, more methods, such as SAVER, outperformed the control (using the original unimputed data) , but SIMPLE still performed the best (Supplementary Figure S18c). For detecting differentially expressed genes, SIMPLE is one of the best methods for non-UMI experiments; but it is slightly worse than some of the other imputation methods in the UMI-based experiments (Supplementary Figure S18b and d). As the model assumption of SIMPLEs is valid for non-UMI protocol, it is expected that SIMPLEs has better performance for non-UMI based simulations. On the other hand, scVI and MAGIC were inferior to any other methods including the control in detecting differentially expressed genes, as they do not model the gene expression variation within each cell cluster adequately. We showed the results for two choices of the numbers of factors and cell clusters: the setting most similar to the truth, and the one most frequently selected by the BIC. The performances of these two settings were similar. Results for full ranges of parameters are shown in Supplementary Figure S19. Although SIMPLEs models data generated from non-UMI platform, it also performs well for UMI data, especially in clustering.

### Human embryonic stem cell differentiation

We applied SIMPLEs to a study of hESC differentiation toward definitive endoderm (30). It includes a single-cell

RNA-seq dataset for seven cell types (1018 cells in total), referred to as 'hESC cell types': two types of embryonic stem cells (H1 and H9), definitive endoderm cells (DEC), endothelial cells (EC), human foreskin fibroblasts (HFF), neuronal progenitor cells (NPC) and trophoblast-like cells (TB) and the bulk RNA-seq dataset for each cell type. DEC, EC and NPC are differentiated cells from three germ layers. DEC and EC share a transient precursor state called mesendoderm. In this study, they also conducted another scRNA-seq experiment at different time points in the differentiation process toward DEC, referred to as 'hESC time course' and produced the corresponding bulk RNA-seq data at each time point.

First, we compared the clustering performance of SIMPLEs with other methods for the 'hESC cell types' dataset. We only input the average gene expression across all cell types in the bulk RNA-seq to SIMPLE-B and SCRABBLE. Otherwise, the gene expression from bulk RNA-seq can disclose the true cell type information we wanted to recover. However, for real application, we can definitely incorporate the bulk RNA-seq of different cell types for better imputation and identification of differential expressed genes among different cell types. This dataset has good quality with low dropout rate. The cells can be nearly perfectly clustered using either the original unimputed data or imputed ones by SIMPLEs, VIPER, or SAVER. However, the aRI was only about 0.75 using scImpute, MAGIC and SCRABBLE (Figure 3B). Since SAVER was too conservative to impute the data (also shown in the following experiments), it is not surprising that it had similar clustering performance as using the original unimputed data.

A main utility of scRNA-seq data in comparison with bulk RNA-seq is to explore cell heterogeneity within major cell types. From the distributions of the latent factors (F) in each cell type, we observed several factors showing larger variations in DECs than in other cell types (Supplementary Figure S8a), indicating hyper transcriptional stochasticity especially for genes in the corresponding gene modules. As an illustration, we re-performed clustering for DECs and ECs only, based on top 1000 genes with largest absolute weights in gene module 1. We showed the results from SIMPLE without involving additional information from bulk RNA-seq to distinguish DEC and EC. DECs and ECs were still well separated but both of them can be further divided into subtypes (Supplementary Figure S8b). The expressions of these top genes further validate the distinct transcriptomes of cell subtypes (Supplementary Figure S8d). The heatmaps using the original unimputed data and the imputed ones look similar since the dropout rate is low for this dataset, yet the differences among subtypes were more discernible after imputation (Supplementary Figure S8c and d). We observed similar subtypes and gene modules by SIMPLE-B.

Then, as another evaluation of the imputation methods in consideration, we took the original hESC cell types dataset as the ground truth and added dropouts to evaluate how well each method can recover the original expression matrix. We designed two dropout schemes: (i) randomly select 20 or 40% entries and set them zero, uniformly for all genes; (ii) set the probability of dropout decreasing with the mean expression of each gene, i.e. $1 - p_g = e^{-0.3 \cdot m_g^2}$. We also subsampled 50 or 75% of all the cells and 3000 genes uniformly in order to test how the performance is influenced by the sample size and to examine the stability of the performance as data vary due to subsampling.

For this experiment, SIMPLEs performed the best in almost all scenarios for all the evaluation metrics (Figure 3). scImpute performed the second best and its clustering results were about same as that of SIMPLEs. The imputed data by SAVER incurred almost the same MSE as the original unimputed data, as SAVER was too conservative to impute any dropouts (Figure 3A). None of the methods could get perfect clustering result (Figure 3B). As we subsampled genes and cells for this experiment, the variance of aRI was large for some cases. To test the performances of identifying differentially expressed genes, we used the marker genes obtained from bulk RNA-seq as the ground truth. Although the AUC was higher for identifying marker genes using SIMPLEs, the markers identified from single-cell datasets and bulk RNA-seq only agreed to some extent, as the AUC was around 0.75 when we used the original dataset without additional dropouts (Figure 3C). Comparing with SIMPLE, SIMPLE-B showed better performance incorporating bulk RNA-seq when we added artificial dropouts whose rate is a monotonically decreasing function of the mean expression (Figure 3A). Furthermore, we varied the parameters in SIMPLEs (Supplementary Figures S5 and 6). The clustering performance and identifying differential expressed genes were not sensitive to the choices of parameters. The imputation performance was worse if specifying too small the upper bound of the dropout rate for SIMPLE, but the prior dropout rate did not affect the imputation performance for SIMPLE-B. Moreover, when assuming a large number of factors and a small penalty parameter in the prior of the loading matrix, the model was susceptible to over-fitting, as the imputation MSE was larger. The imputation performance of SIMPLE-B stayed similar as varying the number of 'high-quality' genes for initialization. On the other hand, for SIMPLE, the performance was better if only using 500 genes with fewest zero entries for initialization, indicating that the cell type information can be obtained using only a few genes. However, setting the number of 'high-quality' gene too small might lose the information of cell subtypes in the real data.

Finally, in order to show that the imputed expression levels by using SIMPLEs can reconstruct gene expression trends in biological processes, we also applied SIMPLEs to the hESC time course dataset. It contains 758 single cells captured at 0, 12, 24, 36, 72 and 96 h in the cell developmental process from pluripotent state through mesendoderm to DE, as well as bulk RNA-seq data at each time point. Since cells' developmental process is asynchronous, cells collected at the same time could be at different developmental stages. Thus, the true developmental state of each cell should be correlated but not exactly the same as the cell's time stamp. Clustering results using either the original unimputed data or the imputed data indicated that similar cell types are observed at 72 and 96 h, whereas cells from other time points are well separated (Supplementary Figure S9). Cells from 72 and 96 h form two clusters, of which one contains purely

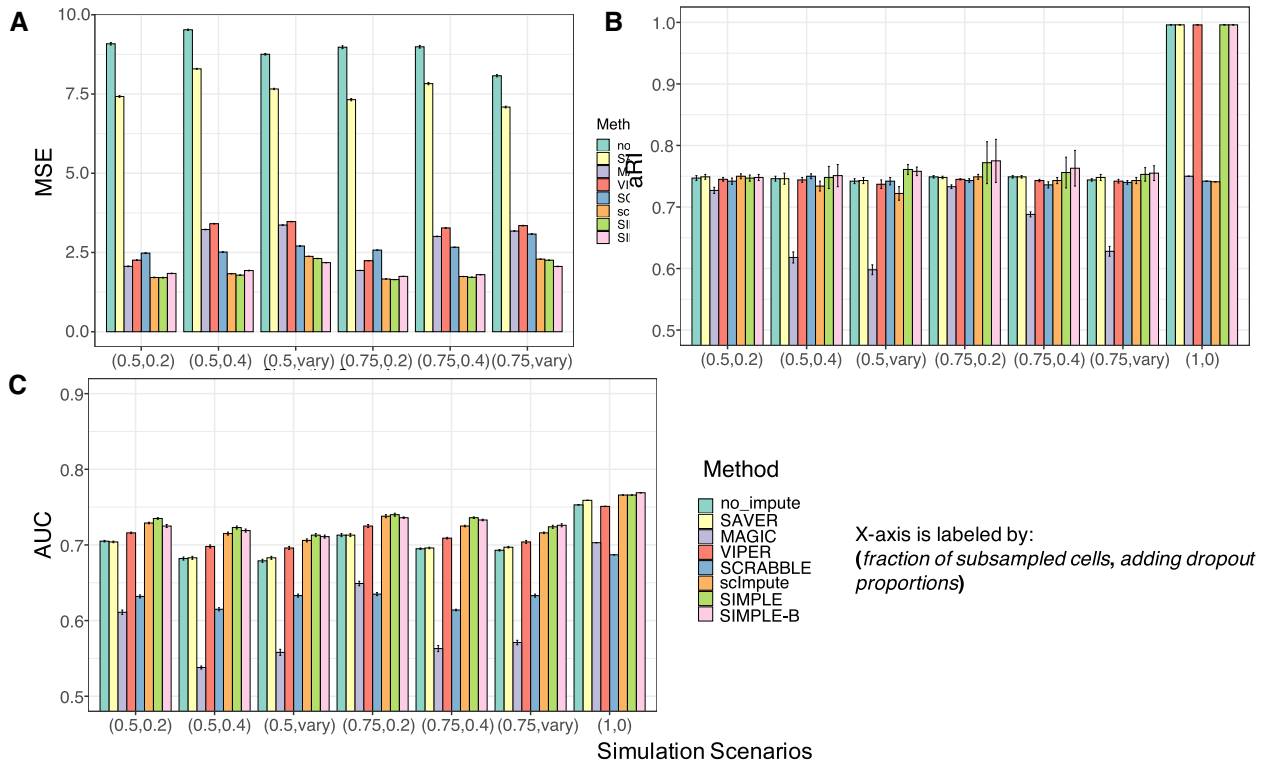


**Figure 3.** Performance comparisons for various simulation scenarios based on the hESC cell types data. (**A**) MSE of imputed values compared with the original dataset without 'fake' dropouts; (**B**) aRI comparing *K*-means clustering results with the true cell labels; (**C**) AUC comparing differentially expressed genes identified by different imputation methods with the genes identified by bulk RNA-seq using DESeq2 (24) as ground truth (see 'Materials and Methods' section). Each bar represents the result from an imputation method or data without imputation ('no_impute'); *X*-axis marks different simulation scenarios. The first number in the parenthesis is the fraction of cells subsampled and the second is either the dropout probability adding to the original dataset or 'vary' for cases where the dropout rate per gene decreases with mean expression. The error bar is the standard error over 10 repetitions.

cells at 96 h, but the other is composed of the cells from both time points, indicating that cells entered the final stage of differentiation asynchronously. As a consequence, in order to identify gene markers for each developmental stage, it is not sufficient to compare gene expression for cells at each time point from bulk RNA-seq. On the other hand, single cell data provides information of the developmental stage of each cell and can be used to identify key genes governing the developmental process.

If the imputation reflects the true biological process, the expression level of the known marker genes in the developmental process should be correlated well with the developmental stage. Considering the fact that the true developmental stage of each cell is not known, we applied Monocle 2 (8) to order the cells using the original unimputed data. It assigned a pseudo-time to each cell indicting the developmental stage of the cell. The pseudo time agreed with the true time label from 0 to 36 h. However, cells from 72 and 96 h cannot be distinguished by pseudo-time. Then, we checked the imputed gene expression of several known markers along the pseudo-time (Figure 4 and Supplementary Figure S10). Imputed values by SIMPLEs followed the cell developmental process and preserved the variability of gene expressions in a single cell, while other methods (e.g. scImpute and MAGIC) tended to impute the gene expression as the mean expression in each cell cluster. Although SIMPLEs utilizes correlated expression changes of the genes with similar functions to infer the dropout values, the imputation will not interfere genes with different functions that can be turned on or off at different stages. For example, *PRDM1*, a DE-specific gene, was expressed at a high level after 72 h indicating that cells differentiated toward the DE state, whereas pluripotent state marker *NANOG* was downregulated during differentiation (30). MAGIC changed all the expression values to the mean in each cell cluster and completely ignored the heterogeneity of single cell expression. SCRABBLE mis-identified the cell state and imputed *PRDM1* by the overall mean expression before 12 h when *PRDM1* is not expressed (Figure 4A). Imputed values by VIPER were correlated with the cell differentiation timeline for most of the cells, but had a similar weakness as SCRABBLE in that it imputed the expression of *PRDM1* for the cells at the very early stage as high as the ones at the intermediate stage. Compared to VIPER, SIMPLEs imputed zero entries at a relatively low expression level for *PRDM1* in the cells at early stages. Moreover, SIMPLEs preserved the stochasticity of single cell expression while other methods reduced the variability of gene expression after imputation. SIMPLEs estimates the variance of expression for each gene. For genes with large variances, the probability of observing zeros from the amplified component is high, so SIMPLEs imputes less frequently and

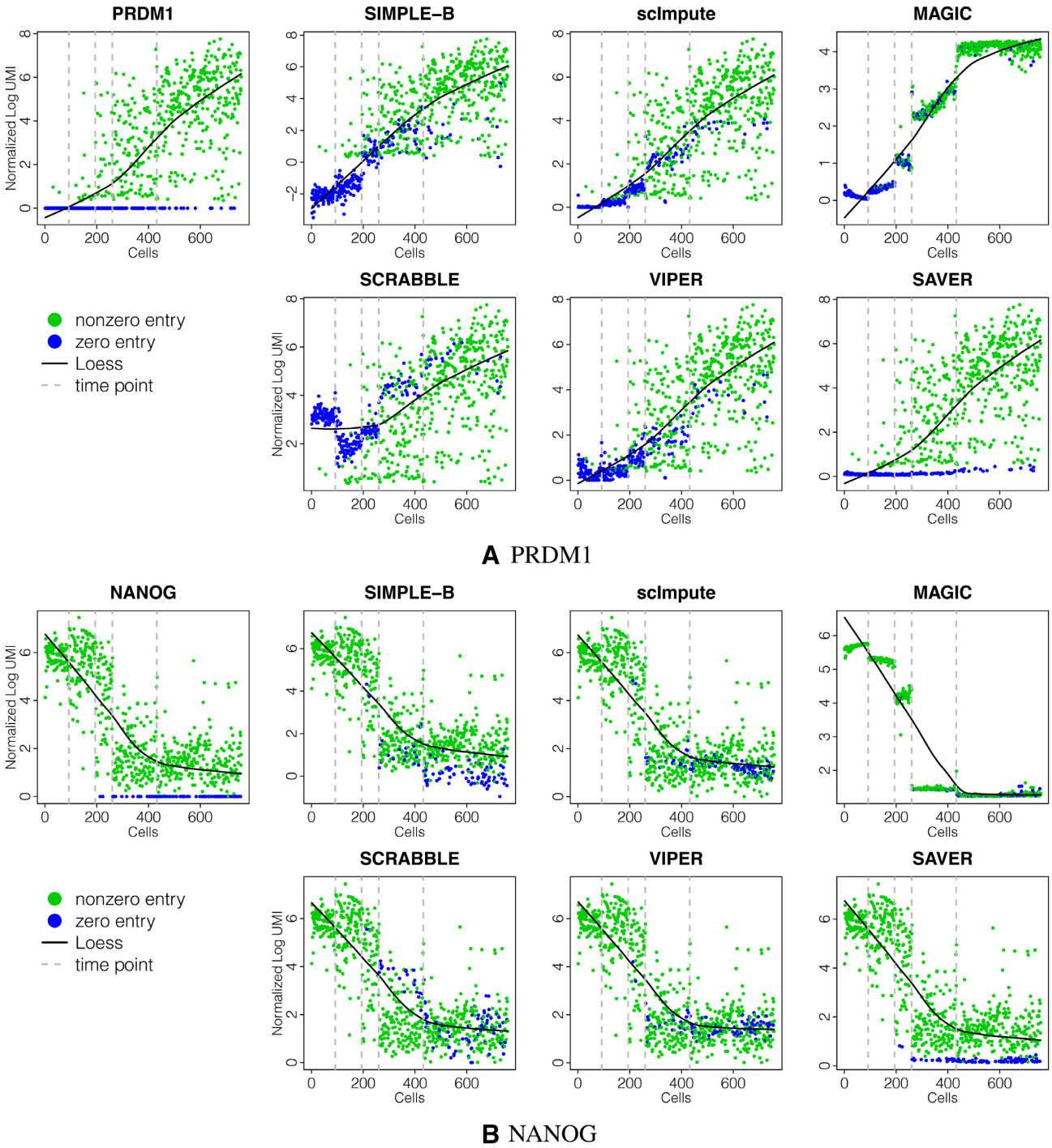


**Figure 4.** Examples of imputed marker gene expressions along the hESC developmental pseudo-time. (**A**) PRDM1 gene; (**B**) NANOG gene. Each dot is the gene expression level in a cell. Green dot: non-zero expression in the original unimputed data. Blue dot: zero in the original unimputed data but is imputed by different methods. The solid black lines are the smoothed LOESS curves computed for each data. Cells are first ordered by time label, then within each time stamp, are ordered by pseudo-time output by Monocle 2, except that we treated cells at 72 and 96 h as one group and ordered them by pseudo time. The dashed lines are the boundaries of 0, 12, 24 and 36 h cells.

retains low expression level for zero entries. For example, *NANOG*, a pluripotent state marker, was expressed at a relatively low level and had a high variance in cells at late stages. VIPER imputed the expression of *NANOG* by its mean after 36 h, but SIMPLEs imputed zero entries at low level and maintained the variability of gene expression at late stages (Figure 4B). More imputed expressions of the marker genes along with the cell developmental pseudo-time are shown in Supplementary Figure S10, demonstrating that SIMPLEs can faithfully recover the gene expression pattern designated by the gene's biological functions.

Moreover, SIMPLEs can restore the (anti-)correlation between genes. For example, both *CER1* and *GATA4* are early DE-specific markers, which were turned on as early as 36 h in the DE differentiation process. However, the correlation between them was attenuated because of dropout. Nevertheless, the close relationship between these two genes was revealed after imputation by SIMPLE-B (Figure 5A). For MAGIC, most pairs of genes had extremely high correlations after imputation. scImpute and VIPER also did well, but imputation from SCRABBLE and SAVER underestimated the correlation between *CER1* and *GATA4*. As another example, *T* and *CXCR4* are negatively correlated as cells transform from a T-positive to a CXCR4-positive state during differentiation toward DE cells (30). The negative correlation between *T* and *CXCR4* was weakened in the original imputed data due to dropouts. All of the imputation methods retrieved the negative correlation but to different extent, from the strongest correlation by MAGIC, then by SIMPLE-B, to the weakest by SAVER which is the same as the original imputed data (Supplementary Figure S11). Furthermore, we identified differentially expressed genes at each time point based on the imputed data (see 'Material and Methods' section). From the gene expression heatmap, cells at different stages showed distinct transcriptome profiles (Figure 5B). These differentially expressed genes of each developmental stage are putative key regulators controlling the developmental process. They included known regulators of the developmental process, such as pluripotent state markers: *SOX2*, *POU5F1*, *NANOG* and *DNMT3B*; early cell state markers (expressed in 12–24 h of differentiation): *NODAL*, *ID1*, *T*, *MSX2*; late cell state markers (turned on at 36–72 h of differentiation): *CER1*, *DKK4*; and DE markers: *KIT*, *PRDM1* and *POU2AF1*.

### Mouse preimplantation embryos

As another example, we applied our method to a single-cell RNA-seq dataset of mouse embryos (31). It contains 12 zygotes, 22 cells at the 2-cell stage, 14 cells at 4-cell stage, 36 cells at 8-cell stage, 50 cells at 16-cell stage and 133 blastocysts (267 cells in total). The dataset does not have the corresponding bulk RNA-seq, so we only compared SIMPLE with other methods. Using either the original unimputed or the imputed data by various methods, we derived similar cell clustering results, which separated blastocysts into two clusters but merged 16-cell and 8-cell stages into one cluster, indicating that blastocysts is even more diverse than cells at 8-cell and 16-cell stages (Figure 6). Indeed, these major cell stages can be further divided into 10 subtypes (31). Although different imputation methods had similar results for clustering the major cell stages, SIMPLE can distinguish subtypes better than others (Figure 6). When trying to cluster the cells into 10 subtypes, SIMPLE achieved aRI = 0.8, whereas the second best method SAVER had aRI = 0.6. In contrast, the clustering aRI was only 0.4 without imputation. Some subtypes have very few cells, e.g. $<10$, rendering them unrecognizable by any method. Yet this dataset contains many more cells at the blastocyst stage, which can be further divided into three subtypes. SIMPLE could clearly separate three stages of blastocysts, i.e. early, mid and late blastocyst, while other methods missed the substructure of cells because the imputation might have over-smoothed the gene expression. In addition, SIMPLE can output multiple imputations and compute the stability of the clustering result for each cell. For example, a small group of cells at 8-cell stage separated from others on the t-SNE plot have a high clustering uncertainty as they sometimes were clustered as an isolated group while sometimes joined with other cells at 8-cell stage; clusters of 8-cell and 16-cell were also not stable as they were often clustered together but occasionally separated into two clusters (Figure 6B). It indicates that the distinction between 8-cell and 16-cell are relatively small and concealed by the noise of the data. If the cluster label is unknown, the consensus clusters identified by multiple imputations is more trustful than clusters that only appear in few imputations, which might due to random noise in the data.

Looking into the distributions of cells' latent factors in each subtype, we observed that subtypes of cells can be discriminated by several latent factors (Supplementary Figure S13). For example, factors 2 and 3 can separate late and early blastocysts from the rest of blastocysts, which explains why SIMPLE was able to distinguish different stages of blastocysts. Factor 6 can distinguish different stages of 2-cell; and factor 1 can differentiate 8-cell and 16-cell. Other factors did not show significant differences in any particular subtype, indicating that the expression of associated gene modules vary uniformly across all subtypes. In summary, by modeling within cluster covariance structure, SIMPLEs can discover different subtypes beyond major cell types.

Then, we compared the expressions of marker genes for each subtype before and after imputations. To identify marker genes, we merged some of the subtypes with $<10$ cells in the original study, and obtained eight subtypes: zygote, 2-cell, 4-cell, 8-cell, 16-cell stage and early, mid, late blastocyst. Marker genes were identified by comparing their expression levels in each subtype with the rest of the cells using the imputed data (see 'Materials and Methods' section). The imputed expression of marker genes by SIMPLE showed clearer patterns of specifically expressed in one or more subtypes than that of other methods (Figure 7; imputed gene expressions by other methods are shown in Supplementary Figure S12).

### Mouse immune cells from multiple organs

To show the scalability of SIMPLE on a large number of cells with diverse cell types, we applied SIMPLE on 12 905 immune cells from 12 mouse organs including 22 known immune cell types (28). Compared with the previous example, different cell types are less discernible since some cell types

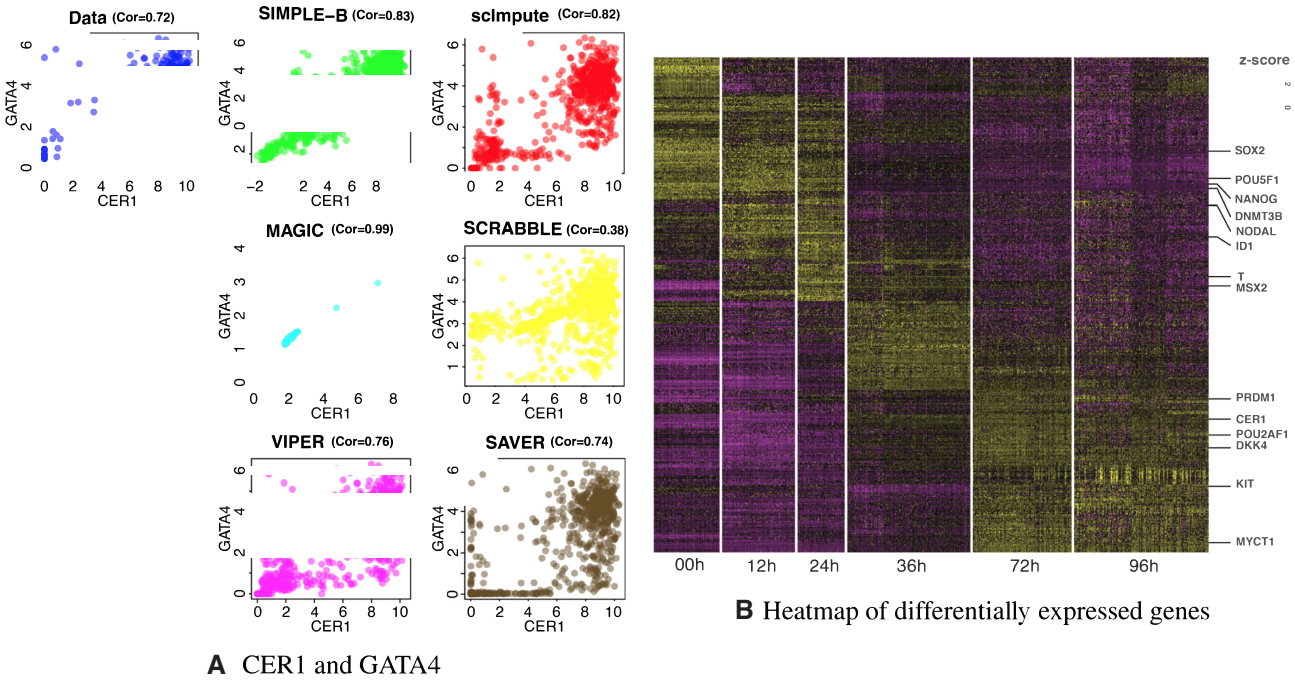


**Figure 5.** hESC time course data. (**A**) Original unimputed or imputed expression of CER1 and GATA4 by different methods. Each dot represents a cell and the correlations between the two genes are shown in the title of each sub-figure. (**B**) Imputed gene expression by SIMPLE-B. Each row is a gene and each column is a cell ordered by the time label. The color indicates the *z*-score (centered and scaled by gene). The heatmap shows top 100 differentially expressed genes ranked the *P*-values from Wilcoxon rank-sum test for each time point.

are of the same origin. The preprocessing procedure is described in the 'Materials and Methods' section. The original unimputed data and the imputed data obtained by SIMPLE with various numbers of clusters and factors were displayed using t-SNE in Figure 8, where colors indicate different cell types. Results from the imputed data resembled the cell clusters identified from the original unimputed data. The overlapping of cells with different labels, especially in the result obtained from the original data, is mainly due to similar cell types in the same lineage on the cell ontology. These cell types come from different tissues but has the same ancestor on the cell hierarchy (Supplementary Figure S14). Compared with the original unimputed data, the imputed data show qualitatively tighter clusters for cells from the same type, but separated different cell types further apart, e.g. the immature B cell from marrow and B cell mainly from other tissues. As the cell labels have some ambiguities and no ground truth of differentially expressed genes is available, we focused on exploratory analysis on clustering result and biologically functional analysis of gene modules in this example.

Varying the parameters in SIMPLE, such as the number of clusters ($M$) and the number of factors ($K$), did not change the t-SNE plot in any noticeable way when the imputed data was used, which demonstrates the robustness of SIMPLE with respect to parameter choices. However, as shown in Figure 8, when increasing $M$ or $K$, different types of T cells reflecting the organ of origin becomes more discernible: immature T cells mostly from thymus and marrows are separated from the T cells from other organs such as fat, lung, limb, muscle and spleen (labeled as T cell) and immature NK T cells. The BIC value of the model reaches the minimum when $M = 1$ or $M = 2$. From the t-SNE plot, as cells form a cell spectrum rather than discrete cell types, it is no doubt that a smaller $M$ fits the data better. After choosing $M$, we select the number of factors at the elbow of the BIC curve, i.e. $K = 15$ or $K = 20$ (Supplementary Figure S15a). Although $M$ is chosen to be 1 or 2, the loading matrix $B$ represents gene modules that can distinguish cell types. Based on these observations, for a given dataset without any prior knowledge about the cell type composition, we set $M$ as 1 and $K$ about 10 to 20 for SIMPLEs for a preliminary exploration and initial imputation.

Furthermore, we focused on 2125 T cells from T-cell family including immature T cell, regulatory T cell, immature NK T cells and others (labeled as 'T cell' without specific subclass information) and analyzed the latent factors and gene modules captured by SIMPLE. We used $M = 1$ and $K = 10$ in the following analysis as indicated by the BIC (Supplementary Figure S15b). We identified representative genes for each gene module based on the weights in the corresponding column of the loading matrix $B$ (see 'Materials and Methods' section) and the biological functions of these genes. Distributions of several latent factors (e.g. medians) showed significant differences across six organs (Supplementary Figure S16), indicating that these gene modules identified by SIMPLE can reflect the origin of T cells. We took as example two gene modules, corresponding to latent factors 3 and 9, since they both have large average weights over genes and their latent factors show distinct patterns across organs (Figure 9A). The medians of latent factor 9 varied the most in organs considered, indicating differential

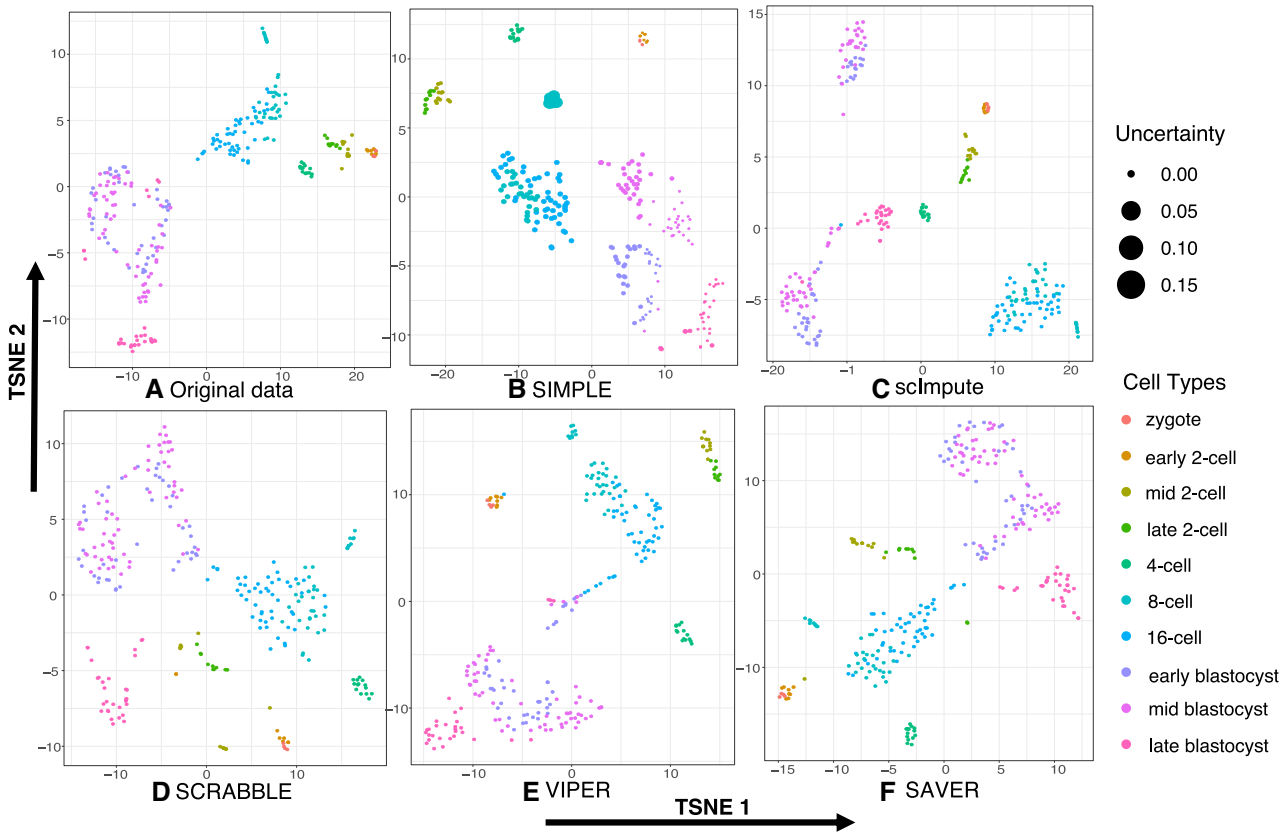


**Figure 6.** Visualizing the original unimputed data or imputed data by different methods using t-SNE for the mouse embryos dataset. (**A**)-(**F**) t-SNE plots for different methods. Each point is a cell colored by subtypes. For SIMPLE, the size of the point indicates the uncertainty of clustering membership of each cell across multiple imputations (the larger the size, the more uncertain the cell is). For SIMPLE, we varied the number of factors from five to eight and the results were similar. The result shown here was obtained from $K = 8$. For SIMPLE and scImpute, we set the number of clusters equal to six, which is the number of major cell stages.

expression of gene module 9 in various organs. The medians of latent factor 3 in cells from thymus was higher than that of other tissues, suggesting specific expression of gene module 3 in thymus. These distinct patterns imply that the genes in these two gene modules play different roles in T-cell developments.

To verify the above observations, we plotted the expression values of the top 10 genes with the largest loadings in these two modules in Figure 9B. After imputations, the percentages of cells expressing these top ranked genes increased as expected, yet the imputed data maintained gene expression variations across organs. Among the top genes from gene module 9, *S100a4* showed a higher expression in fat and limb muscle, which is consistent with the result from a previous study (28). Many of the top genes from gene module 3 had higher expression levels in thymus than in other organs and function in the early stage of T-cell development. For example, *Myb* encodes c-Myb transcription factor, and is essential in early T-cell development (32); *Dnnt* encodes a DNA nucleotidylexotransferase, which is a specific DNA polymerase in pre-B and pre-T cells; finally, *Rag2* involves in the recombination during B- and T-cell development.

The time cost and memory usage of SIMPLEs for this example is approximately linear to the number of cells. Comparing the running times with other methods, SIMPLEs is the second fastest method and is six times faster than scImpute and SAVER for a dataset with 12K cells (Supplementary Figure S21).

## DISCUSSION

SIMPLEs impute dropout values in the single cell RNA-seq data based on both cell similarities and gene correlations. The imputed data matrix can be used to reduce dimensionality for visualizing the cell spectrum, to identify markers between different samples, and to construct gene co-expression networks. The underlying model of SIMPLEs shares some similarities with previous zero-inflated latent factor models for scRNA-seq, e.g. ZINB-WaVE (33) and ZIFA (34). All of them assume a low-rank latent structure among genes, described by several latent factors. However, SIMPLEs assume that cells are composed of a mixture of cell types that can share latent factors; whereas previous methods usually assume a single cell cluster. SIMPLEs iteratively cluster the cells, identify correlated gene modules and latent factors, and impute dropout values within each cluster utilizing the expressions of other correlated genes. Although these latent factors are shared among cell types in SIMPLEs, they are allowed to have different levels of variation in different cell types. The latent factor models in ZINB-WaVE and ZIFA are equivalent to SIMPLEs when

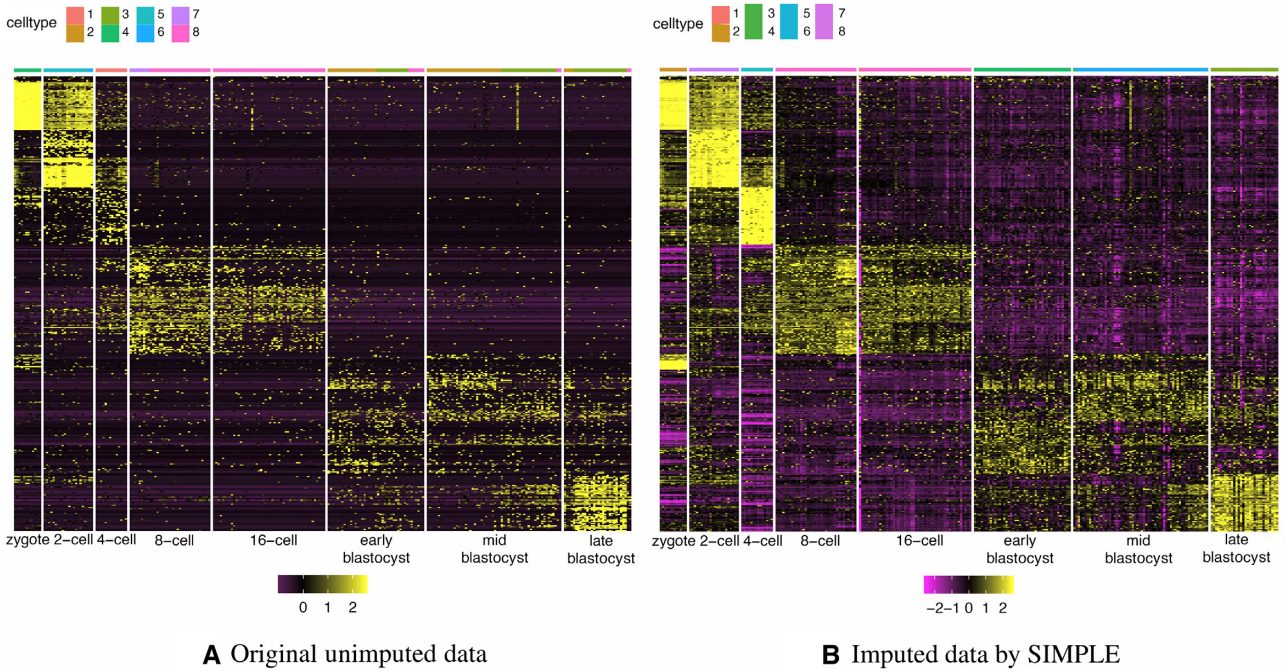


**Figure 7.** Gene expression values based on the mouse embryos dataset (**A**) without imputations, and (**B**) with imputations produced by SIMPLE. Each row is a marker gene and each column is a cell ordered cell types. The purple to yellow color indicates the *z*-score range (centered and scaled by gene). The color bar above the heatmap shows the clustering result by each gene expression matrix. The order of the genes is the same in these heatmaps, but the cells within each cell type are ordered by the clusters obtained using each data matrix. Differentially expressed genes for each of the eight cell types were identified by Wilcoxon rank-sum test using imputed data. For clarity, we show the top 50 marker genes for each cell type ranked by the *P*-values.

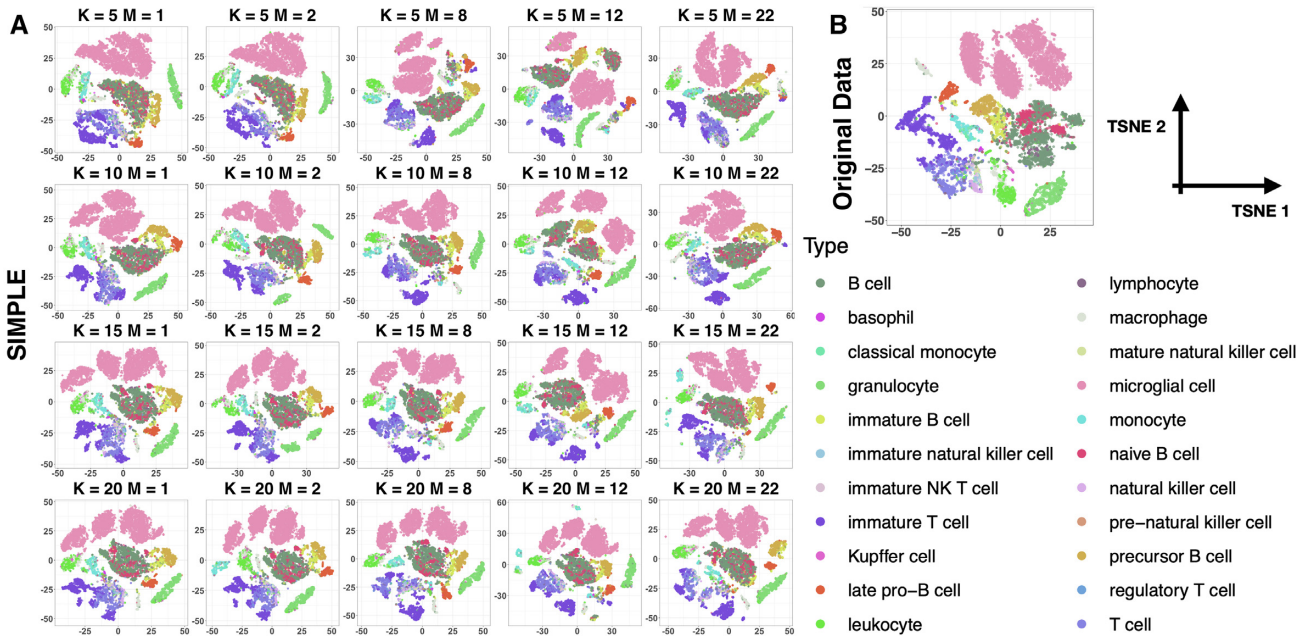


**Figure 8.** t-SNE based data visualization of both the original unimputed data and the imputed data by SIMPLE with different parameters. Each dot represents a cell and colors indicate different cell types. (**A**) The visualization of imputed data by SIMPLE using different combinations of $K = 5$, 10, 15, 20 and $M = 1$, 2, 8, 12, 22. (**B**) The visualization of the original unimputed data.

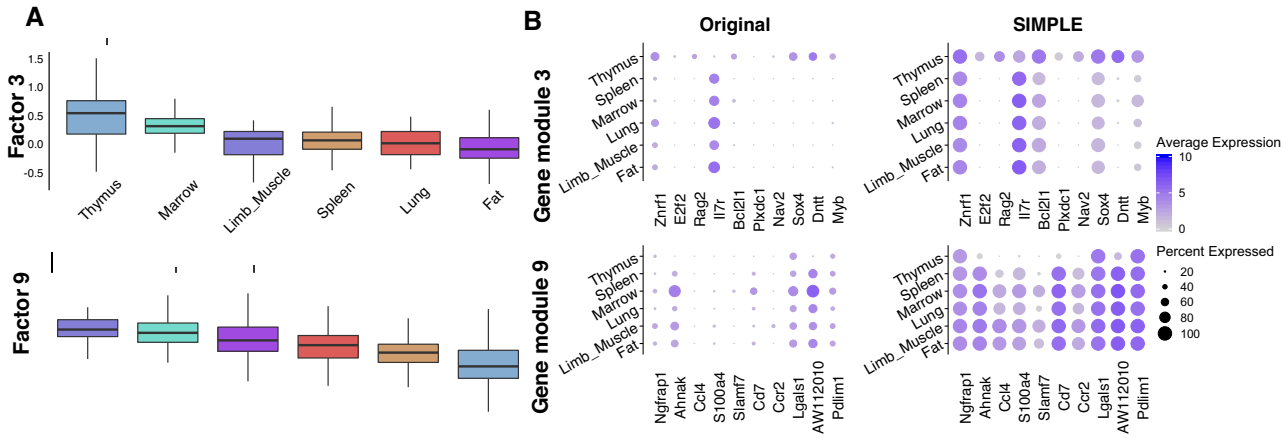


**Figure 9.** Analysis of the latent factors 3 and 9 discovered by SIMPLE for T-cell family in six organs. (**A**) The boxplots show the distributions of factors 3 and 9 in six organs. (**B**) The dot-plots show the gene expression patterns of the top 10 genes across six organs in the original unimputed and imputed data. These genes have largest coefficients in the corresponding gene module.

assuming the number of cell clusters is one. Assuming one broad cell type, the learned latent factors usually represent various cell types, and the associated gene modules are often composed of cell-type markers. However, when combined with cell clustering, SIMPLEs can not only model differential gene expression among cell types but also the correlation between genes within each cell type. In this way, SIMPLEs is more flexible to model gene expression patterns between and within cell types, and better preserve gene modules and cell cluster variations, compared to previous zero-inflated latent factor models.

Integrating with the corresponding bulk RNA-seq data, SIMPLE-B can give an improved estimate of dropout rates, which influences strongly how much the data should be imputed. It is shown in simulations that SIMPLE-B combining bulk RNA-seq has significant advantages over SIMPLE in recovering the dropout values (e.g. Table 1). Only one very recent method, SCRABBLE, can incorporate bulk RNA-seq information for imputation. Compared with SIMPLE-B, SCRABBLE is less optimal in estimating gene-gene correlation and in cell clustering when high dropout rates are present. SCRABBLE also underestimates the variability of gene expressions in single cell, as shown in our analysis of the hESC dataset.

Most existing imputation methods only consider a single imputation, whereas SIMPLEs can output multiple imputations, which can be used to assess clustering stability and reliability (Figure 6B). To the best of our knowledge, only SAVER can output the variance of each imputed value, but it does not reflect the joint variability of multiple genes' expressions. Furthermore, SIMPLEs can identify activated gene modules with a high variability in one or more cell types, which can be used to identify subtypes of cells or gene modules that are associated with the attributes of the samples.

Our analyses of the first two real datasets illustrate that latent gene modules discovered by SIMPLEs can be used to further classify subtypes of human DECs and stages of blastocyst in mouse embryos respectively. Moreover, in the hESC time course data, we showed that our imputed values for marker genes followed the developmental stages of each cell and can be used to discover candidate regulators that are important to the developmental process. Finally, the analysis of a large-scale dataset on mouse immune cells from multiple organs demonstrates the scalability and robustness of SIMPLEs. We also identified several gene modules that varied in expression in T cells from multiple organs, including key genes that regulate T-cell development.

SIMPLEs is designed for non-UMI based single-cell sequencing protocols, e.g. Smart-seq2, as the distribution of log-normalized data can be approximated by a censored Gaussian distribution when the total number of counts is large. Compared to methods assuming zero-inflated negative binomial distribution as the marginal distribution of gene expression, e.g. ZINB-WaVE, it is computationally faster and more robust to fit the censored Gaussian distribution than to estimate the parameters of a negative binomial distribution. Estimating the overdispersion parameter of negative binomial distribution is usually troublesome when the sample size is small or the dropout rate is high. In fact, SIMPLEs shows a modest difference in performances when the input data is from a non-UMI based protocol versus that from the UMI-based protocol. We evaluated SIMPLEs on a recent single-cell benchmark dataset from a UMI-based 10× Genomics platform (27). SIMPLEs showed better performances than other methods on the detection of differentially expressed genes (Supplementary Figure S20c and d). Almost all the methods showed nearly perfect performances on clustering (Supplementary Figure S20e) due to the high quality of this dataset and the purity of the five cell lines. SIMPLEs' assumption on the marginal distribution of gene expression is related to the zero-inflated normal distribution in MAST (35). They used a hurdle model, in which all of the zeros are treated the same without distinguishing dropout or biologically low expression. However, we use a censored Gaussian distribution to model the

'amplified' gene expression, which can generate zeros due to low expression levels. This prevents SIMPLEs from overly imputation, as the zero entries because of biologically low expression will be kept as zeros after imputation.

As shown by Jin *et al*. (36), in high dimensional settings when the number of features is much more than the number of samples, feature selection is crucial for recovering the true clusters. Since the likelihood function of our model is non-convex, our algorithm is also influenced by the initialization of the clustering and imputation. We found it a practically robust strategy to initialize with clusters and latent factors estimated from high quality genes. To further improve clustering performance, it may be worthwhile to consider more sophisticated gene selection procedure for initialization (37). We used a nested Monte Carlo EM algorithm for the estimation, but optimizing the factor loading matrix $B$ in the M-step is still computationally intensive. In order to analyze large-scale single-cell sequencing data, a stochastic gradient descent algorithm can be employed to further reduce the computational time.

SIMPLEs utilizes the BIC for choosing tuning parameters, which appears to perform well when the number of cells per type is large. However, when the number of cells or the difference among cell types is small, the BIC tends to choose fewer cell types than the truth, which is consistent with the prevailing wisdom that the BIC tends to be more conservative than other information criteria. Indeed, since the number of cells in each cell type and the distance between cell types vary, the number of clusters chosen by the information criteria, can be different from biological knowledge. On the other hand, the number of clusters chosen for imputation should also depend on the dropout rate. If the dropout rate is high, one requires pooling information from more cells so as to obtain more reliable and robust imputations. As a consequence, the number of clusters should be small, and *vice versa*. Thus, choosing a smaller number of clusters than truth can be beneficial in these cases. In the mouse immune cells dataset, we showed that the imputation results were robust to the input number of clusters. We also showed that SIMPLEs are not sensitive to the number of factors and the priors of loading matrix and dropout rate in simulations.

After initial imputation and visualization, one may rerun SIMPLEs with different $M$'s for clustering or apply other clustering algorithm and visualization method to the imputed gene expression matrix. We only implemented a simple clustering method as we focused on imputation in this paper, but more sophisticated clustering methods may be necessary to improve the clustering accuracy for rare cell types (38) and cell types with a complex hierarchy.

Our present model does not account for the uncertainty in estimating $B$, which is relevant for quantifying the uncertainty of the imputed values. $B$ can be estimated accurately when the number of cells increases, but a more careful study of the uncertainty in estimating $B$ with the presence of a large amount of missing data are needed. Moreover, SIMPLE-B relies on the fact that the cell composition in the bulk RNA-seq is the same as that in the single cell experiment, which might be violated in some cases. One possible approach is to characterize cell type compositions in bulk RNA-seq data by deconvolution with respect to the expression patterns of high quality genes in each cell type observed in the single cell data (39). Finally, the relationship among genes and cells can be much more complicated than the linear latent factors model assumed in SIMPLEs. It is of interest to incorporate complex nonlinear relationships into our model, which may enhance our understanding about the interactions of cell clusters and gene functional modules at the single cell level.

## DATA AVAILABILITY

SIMPLEs is implemented using R. The software package is freely available under the MIT license, and is deposited at the GitHub repository (https://github.com/JunLiuLab/SIMPLEs). The datasets used in the manuscript can be downloaded from Zenodo (https://doi.org/10.5281/zenodo.3958371). The details of how the datasets are analyzed in this study are included in Supplementary Data. The codes used to analyze the experiments in the manuscript are available from the GitHub repository (https://github.com/JunLiuLab/SIMPLEs2020).

## SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

## ACKNOWLEDGEMENTS

We thank Professor Junying Yuan for inspiring discussions.
*Author contributions:* Z.H. designed the method SIMPLEs. Z.H. and S.Z. conceived and conducted the experiments and analyzed the results. J.L. supervised the study. All authors read and approved the final manuscript.

## FUNDING

National Science Foundation [DMS-161303,DMS-1903139]; National Institutes of Health [1RF1AG055521-01A1]. Funding for open access charge: National Institutes of Health [1RF1AG055521-01A1].
*Conflict of interest statement.* None declared.

## REFERENCES

1. Papalexi,E. and Satija,R. (2017) Single-cell RNA sequencing to explore immune cell heterogeneity. *Nat. Rev. Immunol.*, **18**, 35–45.
2. Villani,A.-C., Satija,R., Reynolds,G., Sarkizova,S., Shekhar,K., Fletcher,J., Griesbeck,M., Butler,A., Zheng,S., Lazo,S. *et al.* (2017) Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science*, **356**, eaah4573.
3. Zheng,G.X., Terry,J.M., Belgrader,P., Ryvkin,P., Bent,Z.W., Wilson,R., Ziraldo,S.B., Wheeler,T.D., McDermott,G.P., Zhu,J. *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, **8**, 1–12.
4. Zeisel,A., Moz-Manchado,A.B., Codeluppi,S., Lönnerberg,P., Manno,G.L., Juréus,A., Marques,S., Munguba,H., He,L., Betsholtz,C. and et,al. (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, **347**, 1138–1142.
5. Macosko,E.Z., Basu,A., Satija,R., Nemesh,J., Shekhar,K., Goldman,M., Tirosh,I., Bialas,A.R., Kamitaki,N., Martersteck,E.M. *et al.* (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, **161**, 1202–1214.
6. Poulin,J.F., Tasic,B., Hjerling-Leffler,J., Trimarchi,J.M. and Awatramani,R. (2016) Disentangling neural cell diversity using single-cell transcriptomics. *Nat. Neurosci.*, **19**, 1131–1141.

7. Keren-Shaul,H., Spinrad,A., Weiner,A., Matcovitch-Natan,O., Dvir-Szternfeld,R., Ulland,T.K., David,E., Baruch,K., Lara-Astaiso,D., Toth,B. *et al.* (2017) A unique microglia type associated with restricting development of Alzheimer's disease. *Cell*, **169**, 1276–1290.
8. Trapnell,C., Cacchiarelli,D., Grimsby,J., Pokharel,P., Li,S., Morse,M., Lennon,N.J., Livak,K.J., Mikkelsen,T.S. and Rinn,J.L. (2014) The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.*, **32**, 381–386.
9. Klein,A.M., Mazutis,L., Akartuna,I., Tallapragada,N., Veres,A., Li,V., Peshkin,L., Weitz,D.A. and Kirschner,M.W. (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**, 1187–1201.
10. Shalek,A.K., Satija,R., Shuga,J., Trombetta,J.J., Gennert,D., Lu,D., Chen,P., Gertner,R.S., Gaublomme,J.T., Yosef,N. *et al.* (2014) Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature*, **510**, 363–369.
11. Stegle,O., Teichmann,S.A. and Marioni,J.C. (2015) Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.*, **16**, 133–145.
12. van Dijk,D., Sharma,R., Nainys,J., Yim,K., Kathail,P., Carr,A.J., Burdziak,C., Moon,K.R., Chaffer,C.L., Pattabiraman,D. *et al.* (2018) Recovering gene interactions from single-cell data using data diffusion. *Cell*, **174**, 716–729.
13. Li,W.V. and Li,J.J. (2018) An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat. Commun.*, **9**, 1–9.
14. Chen,M. and Zhou,X. (2018) VIPER: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies. *Genome Biol.*, **19**, 1–15.
15. Peng,T., Zhu,Q., Yin,P. and Tan,K. (2019) SCRABBLE: single-cell RNA-seq imputation constrained by bulk RNA-seq data. *Genome Biol.*, **20**, 88.
16. Huang,M., Wang,J., Torre,E., Dueck,H., Shaffer,S., Bonasio,R., Murray,J.I., Raj,A., Li,M. and Zhang,N.R. (2018) SAVER: gene expression recovery for single-cell RNA sequencing. *Nat. Methods*, **15**, 539–542.
17. Miao,Z., Deng,K., Wang,X. and Zhang,X. (2018) DEsingle for detecting three types of differential expression in single-cell RNA-seq data. *Bioinformatics*, **34**, 3223–3224.
18. Lopez,R., Regier,J., Cole,M.B., Jordan,M.I. and Yosef,N. (2018) Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, **15**, 1053–1058.
19. Dempster,A.P., Laird,N.M. and Rubin,D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B Methodol.* , **39**, 1–22.
20. Van Dyk,D.A. (2000) Nesting EM algorithms for computational efficiency. *Stat. Sin.*, **10**, 203–225.
21. Bai,J. and Ng,S. (2002) Determining the number of factors in approximate factor models. *Econometrica*, **70**, 191–221.
22. Schroth,G.P., Gertz,J., Myers,R.M., Williams,B.A., McCue,K., Marinov,G.K. and Wold,B.J. (2013) From single-cell to cell-pool transcriptomes: stochasticity in gene expression and RNA splicing. *Genome Res.*, **24**, 496–510.
23. Strehl,A. and Ghosh,J. (2003) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, **3**, 583–617.
24. Love,M.I., Huber,W. and Anders,S. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, **15**, 550.
25. Holik,A.Z., Law,C.W., Liu,R., Wang,Z., Wang,W., Ahn,J., Asselin-Labat,M.-L., Smyth,G.K. and Ritchie,M.E. (2017) RNA-seq mixology: designing realistic control experiments to compare protocols and analysis methods. *Nucleic Acids Res.*, **45**, e30.
26. Zhang,X., Xu,C. and Yosef,N. (2019) Simulating multiple faceted variability in single cell RNA sequencing. *Nat. Commun.*, **10**, 1–16.
27. Tian,L., Dong,X., Freytag,S., Lê Cao,K.-A., Su,S., JalalAbadi,A., Amann-Zalcenstein,D., Weber,T.S., Seidi,A., Jabbari,J.S. *et al.* (2019) Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat. methods*, **16**, 479–487.
28. Schaum,N., Karkanias,J., Neff,N.F., May,A.P., Quake,S.R., Wyss-Coray,T., Darmanis,S., Batson,J., Botvinnik,O., Chen,M.B. *et al.* (2018) Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature*, **562**, 367–372.
29. Ong,E., Xiang,Z., Zhao,B., Liu,Y., Lin,Y., Zheng,J., Mungall,C., Courtot,M., Ruttenberg,A. and He,Y. (2017) Ontobee: a linked ontology data server to support ontology term dereferencing, linkage, query and integration. *Nucleic Acids Res.*, **45**, D347–D352.
30. Chu,L.F., Leng,N., Zhang,J., Hou,Z., Mamott,D., Vereide,D.T., Choi,J., Kendziorski,C., Stewart,R. and Thomson,J.A. (2016) Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biol.*, **17**, 1–20.
31. Deng,Q., Ramsköld,D., Reinius,B. and Sandberg,R. (2014) Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, **343**, 193–196.
32. Allen,R.D., Bender,T.P. and Siu,G. (1999) c-Myb is essential for early T-cell development. *Gene. Dev.*, **13**, 1073–1078.
33. Risso,D., Perraudeau,F., Gribkova,S., Dudoit,S. and Vert,J.-P. (2018) A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat. Commun.*, **9**, 1–17.
34. Pierson,E. and Yau,C. (2015) ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.*, **16**, 1–10.
35. Finak,G., McDavid,A., Yajima,M., Deng,J., Gersuk,V., Shalek,A.K., Slichter,C.K., Miller,H.W., McElrath,M.J., Prlic,M. *et al.* (2015) MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. *Genome Biol.*, **16**, 1–13.
36. Jin,J. and Wang,W. (2016) Influential features PCA for high dimensional clustering. *Ann. Stat.*, **44**, 2323–2359.
37. Andrews,T.S. and Hemberg,M. (2018) M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics*, **35**, 2865–2867.
38. Jiang,L., Chen,H., Pinello,L. and Yuan,G.-C. (2016) GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome Biol.*, **17**, 144.
39. Wang,X., Park,J., Susztak,K., Zhang,N.R. and Li,M. (2019) Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nat. Commun.*, **10**, 1–9.