Bioinformatics, 36(13), 2020, 4021–4029
doi: 10.1093/bioinformatics/btaa278
Advance Access Publication Date: 29 April 2020
Original Paper

Gene expression

PRIME: a probabilistic imputation method to reduce dropout effects in single-cell RNA sequencing

Hyundoo Jeong[1,*] and Zhandong Liu [2,3,*]

[1]Department of Mechatronics Engineering, Incheon National University, Incheon, Korea, [2]Jan and Dan Duncan Neurological Research Institute, Texas Children's Hospital and [3]Department of Pediatrics, Baylor College of Medicine, Houston, TX 77030, USA

*To whom correspondence should be addressed.

Associate Editor: Jan Gorodkin

Received on January 16, 2019; revised on March 3, 2020; editorial decision on April 18, 2020; accepted on April 22, 2020

Abstract

Summary: Single-cell RNA sequencing technology provides a novel means to analyze the transcriptomic profiles of individual cells. The technique is vulnerable, however, to a type of noise called dropout effects, which lead to zero-inflated distributions in the transcriptome profile and reduce the reliability of the results. Single-cell RNA sequencing data, therefore, need to be carefully processed before in-depth analysis. Here, we describe a novel imputation method that reduces dropout effects in single-cell sequencing. We construct a cell correspondence network and adjust gene expression estimates based on transcriptome profiles for the local subnetwork of cells of the same type. We comprehensively evaluated this method, called PRIME (PRobabilistic IMputation to reduce dropout effects in Expression profiles of single-cell sequencing), on synthetic and eight real single-cell sequencing datasets and verified that it improves the quality of visualization and accuracy of clustering analysis and can discover gene expression patterns hidden by noise.

Availability and implementation: The source code for the proposed method is freely available at https://github.com/hyundoo/PRIME.

Contact: hdjeong@chosun.ac.kr or zhandonl@bcm.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The rapid development of single-cell RNA sequencing technologies (Hashimshony *et al.*, 2012; Islam *et al.*, 2014; Klein *et al.*, 2015; Macosko *et al.*, 2015) has enabled researchers to acquire detailed transcriptomic profiles for individual cells in a high-throughput manner. This technology provides an important means for studying cell-to-cell variability, and it is becoming a critical tool for a variety of research endeavors, including cell type identification, pseudo time ordering and deconvolution of heterogeneous samples (Haque *et al.*, 2017; Hu *et al.*, 2018; Liang and Fu, 2017; Tang *et al.*, 2011; Wang and Navin, 2015).

The principle drawback of current single-cell RNA sequencing technology is its vulnerability to technical and biological noise. Individual cells have only a very small amount of mRNA (compared to tissue samples), which requires enormous amplification before analysis. A low initial quantity of a particular transcript can mean that it will be completely missed during the reverse transcription and DNA amplification steps, and thus will not be detectable by subsequent sequencing. Neighboring cells can have wide variability in gene expression, such that a gene expressed at a moderate or high level in one cell is expressed at a low level in another and thus fails to be detected, leading to a 'false zero' known as a dropout event. Single-cell RNA seq data are notorious for producing an excessive number of artificial zeros in the expression profile, which must be distinguished from 'true zeroes'. Several new analytic tools have been developed using zero-inflated models (Finak *et al.*, 2015; Kharchenko *et al.*, 2014; Pierson and Yau, 2015), but most of the current genomic tools were developed based on the distribution of bulk sequencing data (Love *et al.*, 2014; Robinson *et al.*, 2010), which is inappropriate to the nature of single-cell sequencing data.

Several computational methods have been developed to reduce the dropout events by imputing the missing values in single-cell sequencing (Eraslan *et al.*, 2019; Huang *et al.*, 2018; Kwak *et al.*, 2017; Li and Li, 2018; van Dijk *et al.*, 2018). SAVER (Huang *et al.*, 2018) models single-cell gene expression with unique molecular identifier (UMI) counts through Poisson–Gamma mixture and estimates the prior parameter using Poisson lasso regression. Then it recovers the dropouts based on the weighted average of the observed and predicted counts. DrImpute (Kwak *et al.*, 2017) identifies the set of similar cells through k-means clustering and imputes the missing values by averaging the gene expression in the same cluster. To enhance the robustness of the imputation results, DrImpute averages them for multiple k parameters. ScImpute (Li and Li, 2018) estimates the dropout probability through a mixture model, where it models the gene expression as a Gaussian distribution and the zero-

© The Author(s) 2020. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

inflated dropout event as a Gamma distribution. It then imputes only those genes with a high dropout probability by utilizing gene expression values from similar cells that are less affected by the dropout events. MAGIC (van Dijk *et al.*, 2018) constructs a Markov transition matrix to represent similarities between cells and powers the matrix up to t times in order to model a heat diffusion process. Then, it imputes the missing values through the weighted average of the same genes for the neighboring cells in the Markov affinity matrix. DCA (Eraslan *et al.*, 2019) adopts the zero-inflated negative binomial distribution to model the single-cell RNA sequencing including dropout events and utilizes a modified autoencoder, where it has three outputs to predict important parameters: dropout probability, dispersion and mean of the negative binomial component. Then, it reduces the zero-inflated noise by substituting the original expression count by the learned mean of the negative binomial component.

In this article, we propose a novel imputation method, called PRIME (PRobabilistic IMputation to reduce dropout effects in Expression profiles of single-cell sequencing), to effectively deal with dropout events in single-cell RNA sequencing. First, we construct a cell correspondence network through similarity measurements across cells (Fig. 1). Next, we identify the local subnetwork for a target cell by using an efficient random walk protocol. Finally, we impute the gene expression in the target cell based on the probabilistic weight parameter, which is computed based on the variance of gene expression in the local subnetwork. We perform these steps until it meets the stop conditions (Fig. 1).

2 Materials and methods

2.1 Datasets and preprocessing

To assess and compare the performance of single-cell imputation methods, we utilized eight single-cell RNA sequencing datasets. (i) Buettner *et al.* (2015) provide single-cell RNA sequencing datasets for mouse embryonic stem (ES) cells at different cell cycle stages. There are 59, 58 and 65 cells in G1, G2M and S phases, respectively. The read count for the cell cycle genes is provided in the Supplementary File in Buettner *et al.* (2015). (ii) Usoskin *et al.* (2015) provided single-cell RNA sequencing data for mouse sensory neurons (for peptidergic nociceptors, non-peptidergic nociceptors, neurofilament containing and tyrosine hydroxylase containing). The raw sequencing data are available at gene expression omnibus (GEO) with accession number GSE59739. (iii) Zeisel *et al.* (2015) performed large-scale single-cell RNA sequencing on the mouse somatosensory cortex and hippocampal CA1 region. In this dataset, there are seven major cell types that can be classified into 47 different subclasses. We retained only the four major cell types (interneurons, oligodendrocytes, pyramidal CA1 and pyramidal S1 neurons) because other cell types have relatively smaller gene expression values and its population is also smaller than that of the other major cell types. The raw data are archived at the GEO with the accession number GSE60361. (iv) The Darmanis dataset (Darmanis *et al.*, 2015) provided single-cell RNA sequencing for human brain, and we removed only the cell type labeled 'hybrid' because it can be considered as the intermediate stage between neurons and astrocytes (Picardi *et al.*, 2017). The raw data are deposited at GEO with the accession number GSE67835. (v) Chu *et al.* (2016) provided bulk and single-cell sequencing for human ES cells and also the time series sequencing for cell differentiation to endoderm. The raw data are archived at GEO with the accession number GSE75748. (vi) The peripheral blood mononuclear cell (PBMC) 4k dataset provided single-cell RNA sequencing for PBMCs obtained from a healthy donor. (vii) Brain 9k dataset includes single-cell RNA sequencing for brain cells from E18 mouse. The brain cells are obtained from cortex, hippocampus and subventricular zone of an E18 mouse. For PBMC 4k and Brain 9k datasets, the gene expression count matrix and cell type labeling are obtained from 10× Genomics webpage. We utilized the predicted cell type labels through a graph clustering, where it is originally reported by 10× Genomics. For all datasets, we removed genes that are not expressed across all cells. The number of cells and cell types for different datasets are summarized in Table 1.

We also utilized synthetic single-cell RNA sequencing datasets to verify the robustness of imputation results for different dropout rates. To generate synthetic single-cell RNA sequencing datasets, we utilized the R package called splatter (Zappia *et al.*, 2017). In the synthetic datasets, we generated 1000 cells with 20 000 genes and equally divided cells into three cell types. To generate gene expression matrices with a differentially expressed genes in each cell type, we set `de.prob` as 0.1 and the dropout rates are controlled by setting `dropout.mid`={0, 1, 2, 3}. We generated 10 different gene expression matrices for different parameter settings.

2.2 Parameter settings for each algorithm

To compare the performance of PRIME with SAVER (Huang *et al.*, 2018), DCA (Eraslan *et al.*, 2019), scImpute (Li and Li, 2018), DrImpute (Kwak *et al.*, 2017) and MAGIC (van Dijk *et al.*, 2018), we utilized R implementation to run each method with the respective default parameters. If the method supports parallel processing, we utilized the maximum number of CPU cores. In this study, we utilized 4 CPU cores for SAVER and scImpute. We tested scImpute based on the default parameters and set the number of the clusters as the number of cell types. For a fair comparison, we tested scImputed without the true label for each cell type, as the other methods do not require the true label. To run DrImpute, we utilized the default parameters and performed the normalization as recommended in the package. We also utilized the default parameter for MAGIC so that it optimizes the diffusion parameter based on their own criterion.

2.3 Data normalization and network construction

The proposed single-cell imputation method consists of three major steps: (i) constructing a cell correspondence network, (ii) identifying a local subnetwork for each cell and (iii) performing a probabilistic imputation for each gene expression. These three steps continue until the maximum number of iterations have been reached or there are no meaningful changes in the imputed expression values compared to the previous iteration. The basic intuition of the iterative approach is that since the raw single-cell RNA sequencing data could be corrupted by technical noise such as dropout events, it is

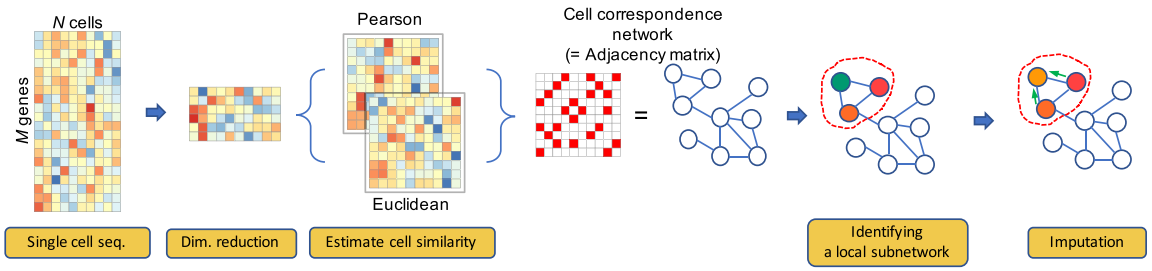

Fig. 1. Overall workflow of PRIME. PRIME reduces the dimensionality of the datasets and estimates two similarities matrix using Euclidean and Pearson correlation. Next, it constructs a network by inserting edges between similar cells. Finally, PRIME adjusts expression levels using the average expression levels of cells in the local subnetwork. This process is iterated till convergence

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

Table 1. Single-cell RNA sequencing datasets

Data source	# cells	# cell types	Source
Buettner *et al.*	182	3	Mouse ES cells
Usoskin *et al.*	622	4	Mouse sensory neurons
Zeisel *et al.*	2448	4	Mouse brain
Darmanis *et al.*	366	4	Human brain
Chu *et al.*	1018	7	Human ES cells
Chu *et al.* (Chu_time)	758	6	Human DE cells
PBMC 4*k*	4340	8	Peripheral blood mononuclear cells
Brain 9*k*	9128	13	Cells from cortex, hippocampus and subventricular zone of mouse

unreliable to impute the dropouts based on noisy datasets. As we impute the technical noise, the reliability of the dataset increases, which improves imputation results in the next iteration.

To begin, suppose that we have single-cell RNA sequencing data and it can be represented as an M by N dimensional matrix, where M is the number of genes and N is the number of cells. We normalize the library size of the single-cell RNA sequencing data matrix using a counts per million (cpm) and take a log-transformation. We have a normalized input $\mathbf{X_n}$, which is given by

$$\mathbf{X_n} = \log_{10}(1 + \mathbf{X}), \tag{1}$$

where $\mathbf{X}$ is cpm transformed input data. Note that the input value is not limited to read counts and it is acceptable if it represents relative expressions of genes across cells. Once we have a normalized data matrix $\mathbf{X_n}$, we start the iterative imputation process by constructing the cell correspondence network based on the cell-to-cell similarity. The cell correspondence network can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where a node $v_i \in \mathcal{V}$ represents an i-th cell and the cell correspondence can be represented as an edge $e_{i,j} \in \mathcal{E}$ so that if cells v_i and v_j are similar to each other, the edge weight $e_{i,j}$ can have a positive value. In this study, we utilized both Euclidean distance and Pearson correlation to estimate cell-to-cell similarity. Before estimating the similarity, since a single-cell sequencing generally includes a number of cells and genes, we first reduce the dimension of the input data $\mathbf{X_n}$ in order to reduce the computational complexity and shorten the running time of the method. To this end, we select highly variable genes across all cells using Seurat (Macosko *et al.*, 2015) and obtain a low-dimensional representation for each cell using principal component analysis (PCA).

Next, we construct the cell correspondence network ($\mathcal{G}_E$) based on the Euclidean distance and the network ($\mathcal{G}_C$) for the Pearson correlation and combine them to obtain a comprehensive cell correspondence network $\mathcal{G}$. First, we compute the Pearson correlation between each cell using top 20 principal components (PCs) to construct a cell correspondence network for the Pearson correlation. Note that we empirically utilized first 20 PCs in order to estimate the cell-to-cell correlation because it leads acceptable results for various datasets based on our experimental results. For a given cell v_i, we select the cells having a high correlation and consider the cells as the neighboring nodes in the cell correspondence network, $\mathcal{G}_C$. Note that the neighboring nodes indicate the set of cells that can be classified as the same cell type with similar expression patterns, and the neighboring nodes for the cell v_i can be selected based on the following criterion:

$$\mathcal{N}_C(v_i) = \{v_j | C_{i,j} \geq c_t(v_i)\}, \tag{2}$$

where $C_{i,j}$ is a Pearson correlation between the cell v_i and v_j, and $c_t(v_i)$ is the threshold to select the neighboring cells. We adaptively select the threshold by taking $\min\{N_{th}, (Q_{th} \cdot \text{percentileof}C_{i,j}, \forall j)\}$. Note that we utilized the default parameter for N_{th} as 0.85 and Q_{th} as 0.9 in experiments and these parameter setting could lead an acceptable result, but it can be adjusted depending on datasets. Then, we insert edges between the cell v_i and the cell $v_j \in \mathcal{N}_C$. The adjacency matrix for a cell correspondence network based on Pearson correlation is given by

$$A^C_{i,j} = \begin{cases} 1, & v_j \in \mathcal{N}_C(v_i), \forall v_i \\ 0, & o.w. \end{cases}. \tag{3}$$

The above adjacency matrix is asymmetric and it generates a directed network because the selection of cell v_j as the neighbor of cell v_i does not necessarily guarantee the opposite case even though the Pearson correlation matrix is symmetric. To make it an undirected network and give more confidence to the bidirectional edges (i.e. the cell v_i selects the cell v_j as its neighboring node, and vice versa), the adjacency matrix for the undirected network can be obtained by linear combination of $\mathbf{A}_\mathrm{C}$ and its transpose, which is given by

$$\mathbf{P}_\mathrm{C} = 0.5 \times (\mathbf{A}_\mathrm{C} + \mathbf{A}^\dagger_\mathrm{C}). \tag{4}$$

Similar to the cell correspondence network based on the Pearson correlation, we first compute the Euclidean distance between the first 20 PCs for each cell. Since the Euclidean distance becomes smaller as the cell-to-cell similarity increases, we utilize the Gaussian kernel to obtain the Euclidean similarity, which is given by

$$\mathbf{E} = \exp(-\tilde{\mathbf{E}}), \tag{5}$$

where $\tilde{\mathbf{E}}$ is the element-wise square of the rescaled Euclidean distance matrix $\tilde{\mathbf{M}} = 2 \cdot \mathbf{M}/\max(\mathbf{M})$ and $\mathbf{M}$ is the $|N| \times |N|$ dimensional matrix representing Euclidean distance between each cell. Then, we can obtain the adjacency matrix $\mathbf{P}_\mathrm{E}$ for the cell correspondence network, $\mathcal{G}_E$, through the same process described in Equation (2) to Equation (4).

After computing the adjacency matrices for the Pearson correlation and Euclidean similarity, we combine both similarities through a linear combination with an equal weight and take an element-wise square, where it gives more weight on the edges consistently identified by both metrics and decreases the weight on the edges identified by only one similarity measurement. The resulting adjacency matrix for the cell correspondence network $\mathcal{G}$ is given by

$$\mathbf{P} = 0.5 \times (\mathbf{P}_\mathrm{C} + \mathbf{P}_\mathrm{E}). \tag{6}$$

In the network construction step, starting from a directed network with an equal weight, we iteratively rescale the edge weights by performing a linear combination of matrices and element-wise square, and this process gives more weight to the consensus edges (e.g. the edge connecting the cells v_i and v_j identified simultaneously by both Pearson correlation and Euclidean criteria).

2.4 Identifying local subnetwork and probabilistic imputation

To reduce the dropout effects in single-cell sequencing in a biologically unbiased manner, it is necessary to distinguish dropouts from true biological zeros (Li and Li, 2018). To this end, we take advantage of the surrounding cells. The basic intuition is that if the gene is not expressed in the majority of cells of the same type, the observed zero is highly likely to be a true zero. However, if the gene has a positive expression in the most of the cells in the same type (i.e. cells in the local subnetwork) but it is not expressed in a particular cell, the detected zero has a higher probability of being a dropout event, where it should be recovered to the true (or expected) values. Thus, we compare the gene's expression in a particular cell to its expected

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

expression in the set of similar cells—a local subnetwork identified by a random walk approach inspired by local graph partitioning using a personalized PageRank (PPR) vector (Andersen et al., 2006). Although the PPR vector can be used to identify the exact local network clustering, it has high computational complexity when dealing with the large-scale networks. Since our aim is not to derive the whole set of cells of the same type but to identify a reasonable local subnetwork, we adopt a heuristic approach to approximate the PPR vector. First, we perform a column-wise normalization in order to obtain the legitimate stochastic matrix (i.e. the transition probability matrix for the random walker). Then, we obtain the transition probability matrix for the random walker over the cell correspondence network $\mathcal{G}$ by performing a matrix product to consider a secondary structural similarity.

Next, to identify the set of similar cells for the cell v_i, we identify the local subnetwork in the cell correspondence network $\mathcal{G}$ through a random walk. Starting from the i-th cell v_i, the random walker performs a random movement for J steps over the cell correspondence network $\mathcal{G}$ and identifies the local subnetwork for the cell v_i by selecting K neighboring cells based on the visiting frequency of the random walker for each node. Note that the parameter J is empirically set to 3 and K is selected by $\min\{(K_{min} \cdot N), (K_{max} \cdot |\mathcal{N}(v_i)|)\}$, where $\mathcal{N}(v_i)$ is the neighboring nodes for the cell v_i in $\mathcal{G}$ and N is the number of cells. Note that we utilized a default parameter 0.2 for K_{min} and 1.25 for K_{max}, respectively. After identifying a local community for the cell v_i, we computed the mean vector $\boldsymbol{\mu}_i$ and variance vector $\boldsymbol{\sigma}_i$ for M genes in the local subnetwork, where the m-th element in these vectors is the mean and variance for the m-th gene of the cells in the local subnetwork. Then, we impute the expression $\boldsymbol{x}_i$ for M genes in the cell v_i based on the mean and variance of the expression values in the local subnetwork. That is, if the m-th gene expression in the cell v_i is reliable (i.e. similar to the mean value for the local subnetwork), we will assign more weight to the expression of cell v_i and utilize the least information from the neighboring cells to adjust the gene expression values. However, if it significantly deviates from the local mean toward zero, we will assign more confidence to the information obtained from the neighboring cells so that the potential dropout events can be recovered. Based on this intuition, the updated rule for gene expression in the cell v_i is given by

$$\mathbf{x}_i^{n+1} = \mathbf{p} \cdot \mathbf{x}_i^n + (1 - \mathbf{p}) \cdot \boldsymbol{\mu}_i, \tag{7}$$

where $\mathbf{p}$ is the probabilistic weight for the current gene expression and the probabilistic weight $\mathbf{p}$ is determined by the following sigmoid function:

$$\mathbf{p} = \frac{1}{1 + \exp\left(-\boldsymbol{\alpha} \cdot (\mathbf{x} - \boldsymbol{\mu}_i)\right)}, \tag{8}$$

where $\boldsymbol{\alpha}$ is scaling coefficients for the sigmoid function and it is selected using the following criterion: $\min\{2 \cdot \exp(-0.05 \cdot \boldsymbol{\sigma}_i), 1\}$. To avoid the extreme case, we set the marginal value for the scaling coefficient α as 1 because if the α is set to very low value, the sigmoid function can approximate a step function. Note that the above framework using a sigmoid function can reduce zero-inflated noise as well as the extremely high expression values because we supposed that the extreme values as an artificial noise with a high chance and it would be desirable to be corrected into a moderate level. However, based on the selected genes, we confirmed that it has more impact on the zero values rather than the highly expressed values (see Supplementary Figs S8 and S12).

We perform the iterative imputation process until it converges or exceeds the maximum number of iterations. Empirically, we stop the iteration if $|\mathbf{X}_{n+1} - \mathbf{X}_n|^2/(|M| \cdot |N|)$ is smaller than 0.05 or the number of iteration exceeds five.

3 Results

3.1 Simulation results using synthetic datasets

To verify the robustness of PRIME for various levels of dropout events in a single-cell RNA sequencing, we generated synthetic datasets using R package called splatter (Zappia *et al.*, 2017). The synthetic dataset includes a reference data with rare dropout events and noisy data having different number of dropout events, where it depends on the parameter setting of `dropout.mid`.

For a simple and intuitive comparison, we compared a low-dimensional visualization of the raw and imputed gene expressions (Fig. 2). We utilized uniform manifold approximation (UMAP) (McInnes *et al.*, 2018) to obtain a low-dimensional visualization of a gene expression for a single-cell sequencing. As we can see in the visualization results for the raw datasets, it is difficult to separate the different cell types as the dropout rate increases. However, PRIME shows a clear separation for different cell types even though the dropout rate increases, where it shows a robustness of PRIME against to the various dropout rates (Fig. 2). Low-dimensional results using PCA andstochastic neighbor embedding stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) also show the similar trend (Supplementary Fig. S1). Although scImpute can separate cells into three types, if we carefully examine the cell type labels, we can easily recognize that the different cell types are clustered together even though the dropout rates are not very high. DrImpute can make a clear separation across different cell types. However, its robustness is not stronger than PRIME because different cell groups in a low-dimensional visualizations are quickly closed as the dropout rate increases. MAGIC can make a clear grouping for datasets having a number of dropout events. However, as we will see in later, it could decrease a variety of gene expressions in each cell, where it is a crucial advantage of single-cell RNA sequencing. Although DCA could separate three cell types for the synthetic data with the least dropout events, DCA failed to clearly separate different cell types as the dropout event increases.

In order to quantitatively evaluate the visualization results, we also compared the within-cluster sum of squares based on two-dimensional representation for each method. Note that the within-cluster sum of squares is given by $\sum_{k=1}^{3} \sum_{j=1}^{|N_k|} \|y_{k,j} - \bar{y}_k\|^2$, where $|N_k|$ is the number of cells with k-th label and $\bar{y}_k = \frac{1}{|N_k|} \sum_{j=1}^{|N_k|} y_{k,j}$. To this aim, we generated 10 synthetic datasets for different dropout rates and computed the average within-cluster sum of squares. For various dropout rates, PRIME achieves much smaller average within-cluster sum of squares then other methods, where it shows the potential of PRIME for cell type identification (Supplementary Fig. S2).

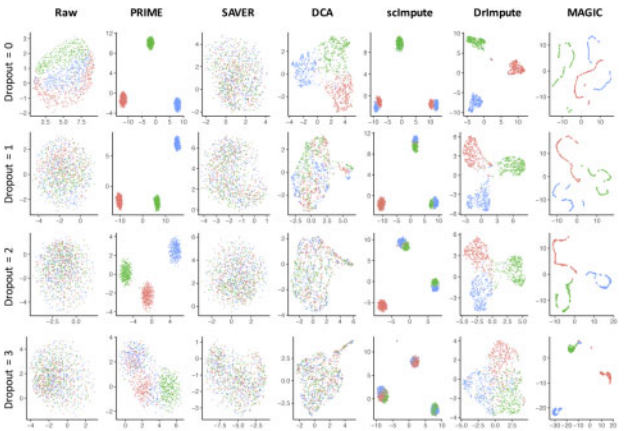

Fig. 2. Low-dimensional representation of synthetic single-cell RNA sequencing datasets for various dropout rates. Tow dimensional representation is obtained through UMAP. Dropout rates are controlled by setting the model parameter `dropout.mid` as 0–3. Note that as the parameter increases, the number of dropout events in the dataset also increases

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

3.2 Better identification of cell types from expression profiles

In analyzing single-cell RNA sequencing datasets, the fundamental first step is to visualize each cell in a low-dimensional space and to identify known and novel cell types through clustering-based methods. Dropout events can decrease cell-to-cell similarity within the same cell type, resulting in mistaken identification of cell type. To visualize single cells in a low-dimensional space, we utilized the cell type labels reported in the original papers and employed three popular dimensional reduction methods, PCA, t-SNE and UMAP. Since the imputation recovers dropouts in each cell and depends strongly on the frequency of the dropout events for each cell, the total number of counts for each cell can change dramatically after the imputation. We therefore renormalized each single cell using the library size after the imputation. In fact, MAGIC normalizes the imputation result by default, but the other methods do not consider post-normalization after imputation. For a fair comparison, we renormalized the imputed gene expression matrix using cpm and perform a log-transformation to obtain low-dimensional visualizations. Our approach was able to impute dropout values across different cell types and led to a better separation between different cell types (Fig. 3).

We compared our method to five other approaches on eight datasets (Buettner *et al.*, 2015; Chu *et al.*, 2016; Darmanis *et al.*, 2015; Usoskin *et al.*, 2015; Zeisel *et al.*, 2015) with the cell type labels provided by each original publication. PRIME generated clear visualization results for all eight datasets (Fig. 3). SAVER showed a negligible effect on the visualization results for most cases. One possible explanation is that SAVER assumed that the gene count can be modeled as a Poisson–Gamma mixture, which can be effective for single-cell sequencing based on UMI counts. But if the count does not follow a Poisson–Gamma mixture distribution, SAVER could fail to effectively impute the dropouts. Even though the count data fits the assumption [e.g. Zeisel dataset (Zeisel *et al.*, 2015)], SAVER does not clearly separate different cell types in a low-dimensional space (Supplementary Figs S3 and S4). ScImpute showed a

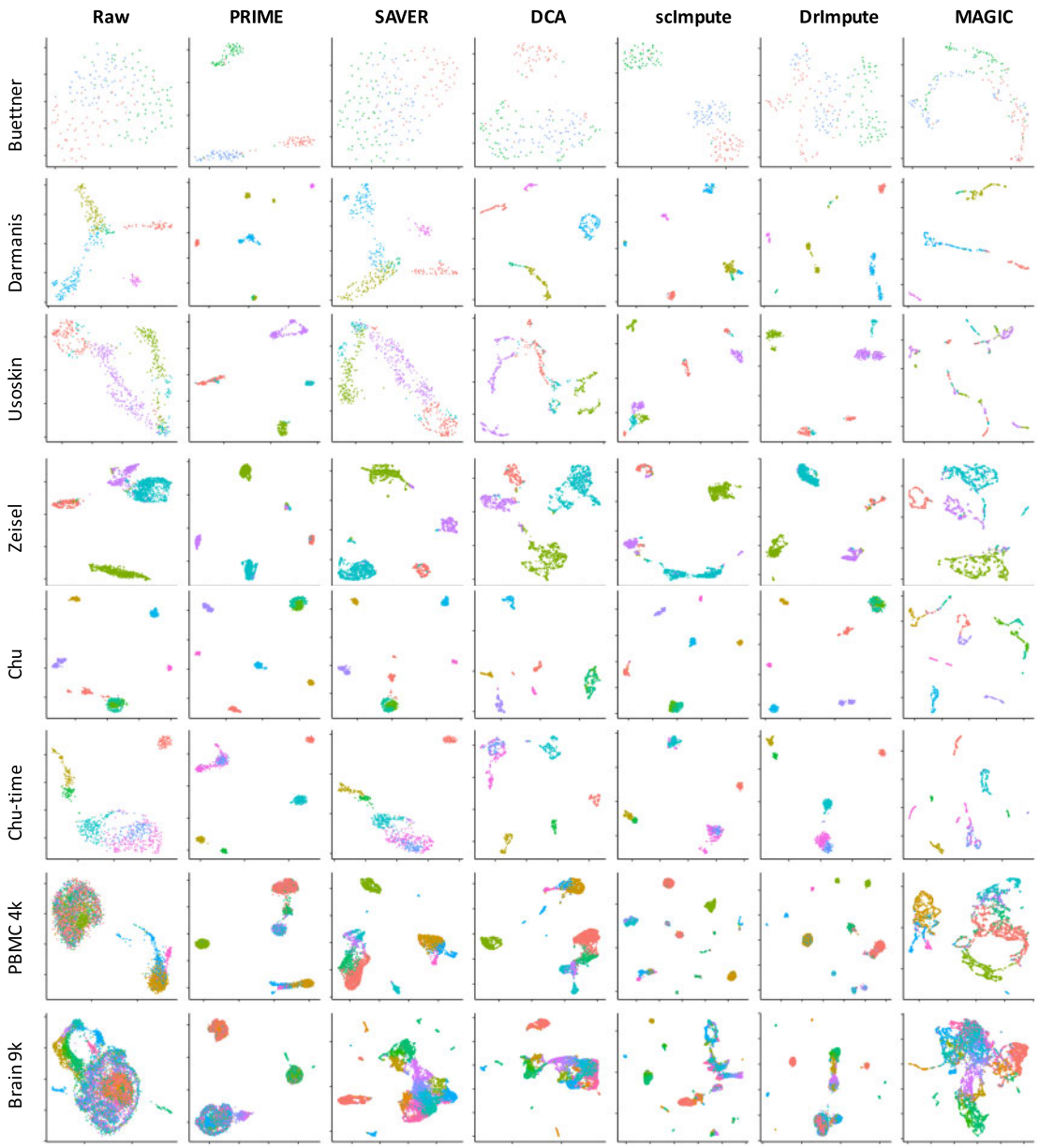

Fig. 3. Low-dimensional embedding of imputed scRNAseq data over eight benchmark datasets. The first two components in the UMAP were plotted. PRIME tends to yield a more clear separation for different cell types while preserving the original features demonstrated in the UMAP plot using raw input

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

comparable result to PRIME in the Buettner dataset (Buettner *et al.*, 2015), but in the Usoskin (Usoskin *et al.*, 2015) and Zeisel (Zeisel *et al.*, 2015) datasets, it divided the same cell types into different clusters and merged different cell types into the same cluster. We also checked the similar issues using a synthetic dataset in the previous section. In fact, if the true label is not available, scImpute first perform a clustering to identify the set of cells that can be potentially classified to the same type. However, the clustering method in scImpute could be inaccurate and unreliable so that it can group different cell types into the same group, where it can also lead incorrect imputation results. MAGIC failed to separate cell types in the low-dimensional space in most of the datasets. In PBMC 4*k* and Brain 9*k* datasets, since the cell types that are originally provided by 10× Genomics are computationally predicted labels, it could be possible that the different labels are assigned to the same cell type, and vice versa. Although the cell type labels for PBMC 4*k* and Brain 9*k* could have a mislabeling, PRIME clearly separates the major cell types and PRIME produced better visualization results than the competing algorithms; the PCA and t-SNE plots for the different imputation methods are provided in Supplementary Figures S3 and S4. Overall, PRIME improves the visibility of single cells when compared to the raw datasets.

Next, we compared the number of dropouts corrected by each algorithm in order to confirm the impact of imputation. For this purpose, we counted how many zeros are corrected to a positive value after performing each imputation method (Supplementary Fig. S5). Interestingly, SAVER corrects almost all the zeros in the raw single-cell RNA sequencing datasets but its visualization result is not much striking. One possible explanation is that SAVER could cautiously correct the majority of artificial zeros so that it can lead negligible changes for the majority of dropouts. We also verified the same results for the selected genes (Supplementary Figs S8 and S12). scImpute changes the least number of zeros for all test cases. PRIME generally corrects about 75–90% of observed zeros to a positive value, and it keeps about 10–25% of estimated zeros as true zeros.

To measure the improvement on cell type clustering of imputation methods, we used pcaReduce (Yau *et al.*, 2016), hierarchical clustering based on a normalized Euclidean distance among cells, Louvain algorithm (Blondel *et al.*, 2008) and spectral clustering using the first two PCs as in Li and Li (2018). We obtained spectral clustering through R package, `speccalt` (Bruneau *et al.*, 2014). To obtain clustering results using Louvain algorithm, we utilized `cluster_louvain` in igraph R package. We evaluated the quality of the clustering based on adjusted rand index, normalized mutual information, Jaccard index and Purity. PRIME outperformed the other methods (Supplementary Fig. S6). These results demonstrate the effectiveness of PRIME in improving cell type discovery.

3.3 Uncover cell state-dependent gene expression patterns

To demonstrate that effective imputation methods can reduce dropouts and lead to the discovery of hidden gene expression patterns in the single-cell data, we compared all the methods on a set of transcriptomic profiles (Buettner *et al.*, 2015) of 182 mESCs over different cell cycle stages (G1, G2M and S). Cells belonging to the same cell cycle phase were not clustered together using hierarchical clustering (i.e. the color label for the column annotation). In each cell cycle phase, the expression of cell cycle genes typically changes in a periodic pattern—one not observable at the normalized raw single-cell expression level (Fig. 4). To determine whether PRIME can improve the signal-to-noise ratio and detect cell cycling patterns, we identified differentially expressed genes with a large fold change across different cell cycle stages in the raw dataset using DEseq2 (Love *et al.*, 2014) for 892 cell cycle genes reported in Buettner *et al.* (2015). Specifically, to identify differentially expressed genes for each cell cycling stage, we obtained a log2 fold change for 892 cell cycle genes through DEseq2. Next, we filtered out genes with the adjusted *P*-value greater than 0.01 and finally selected a set of genes if their log2 fold change is greater than 1.5. Then, we plotted the heatmap for differentially expressed genes with a row-wise normalization to visualize cyclical patterns in gene expression in the different cell cycle stages and performed hierarchical clustering to validate the consistency of gene expression at the same stages. The row-wise normalization can be obtained by computing a *z*-score for each row. Note that the *z*-score is given by $z_i = \frac{x_i - \bar{x}}{\sigma}$, where $\bar{x}$ is a sample mean and σ is a sample standard deviation. The legend in Figure 4 indicates *z*-scores for corresponding genes.

After PRIME imputation, we observed a clear pattern where cells were grouped correctly based on their cell cycle stage. In contrast, no cyclical patterns in DrImpute or SAVER were apparent, and cells across multiple cell cycle phases were clustered together. Although DCA shows better cyclical patterns than DrImpute and SAVER, the cells that are actually in a different cell cycle stage are grouped together by showing the similar expression patterns, which could lead to incorrect differential analysis results. MAGIC showed a noticeable periodic pattern in the heatmap, but careful examination of the hierarchical clustering results reveals that cells in different phases are clustered together, indicating that gene expression in the same cell cycle phase is highly incoherent. The choice of clustering methods is not the main reason for the mis-clustering of different cell cycle stages (Supplementary Fig. S7).

We verified that PRIME recovers dropouts while keeping biological differences between different cycles through the gene expression profiles for the selected genes at different stages of the cell cycle (Supplementary Fig. S8). SAVER yields negligible imputation effects, and MAGIC and DCA decrease biological variations across different cell cycle phases. For example, although Cdc25b, Troap and Katnb1 are not highly expressed in G1 phase but they are well expressed in S phase, MAGIC and DCA could remove this biological heterogeneity and they make their gene expression values almost similar to different cell cycling stages. It clearly shows that the proposed method effectively recovers a greater number of dropouts while maintaining biological heterogeneity (i.e. changing gene expression patterns in different cell cycling stages) across different cycling phases (Supplementary Fig. S8).

3.4 Stable inference on co-expression network

Many bioinformatic studies estimate gene co-expression networks from bulk RNA sequencing data. The number of observations provided by single-cell RNA sequencing—from 500 to 10 K cells in a typical experiment—makes this method even more powerful. A large number of dropouts could decrease the stability of the co-expression network, however, resulting in spurious inferences on gene–gene co-expression patterns. To determine whether PRIME can improve the stability of the network inference, we estimated co-expression networks from both raw and imputed data on single-cell profiles generated from human ES cells and differentiated definitive endoderm (DE) cells at 0, 12, 24, 36, 72 and 96 h. We estimated network stability by subsampling 758 cells in this time series dataset. We then used the method of Meinshausen and Bühlmann (2006) to estimate over 100 regularization parameters for the network. For a fair comparison, networks estimated from different imputation methods were aligned by their network sparsity level, where it can be computed through the ratio of the edges to the number of nodes in the inferred co-expression network. Among these inferred edges, we only counted the reliable edges where it can be reproducible >90%. PRIME consistently identified more reliable edges, no matter the sparsity level (Fig. 5). The number of reliable edges for SAVER and DrImpute is smaller than the raw dataset.

For each imputation method, the Meinshausen and Buhlmann method (Meinshausen and Bühlmann, 2006) also selected an optimum network at a global stability threshold of 0.90. The optimum network for the raw data has 276 singleton nodes and 391 edges; the optimum network for PRIME has the lowest number of singleton nodes (212) and the largest number of edges (483) of all the methods (Table 2). Since the learned network based on MAGIC is an almost fully connected network that violates the sparsity rule, we excluded it for this comparison. Additionally, we also excluded the inferred network based on DCA as it yields extremely sparse network and clearly breaks the sparsity constraint. When comparing the raw data, the proposed method can decrease 64 singleton nodes

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

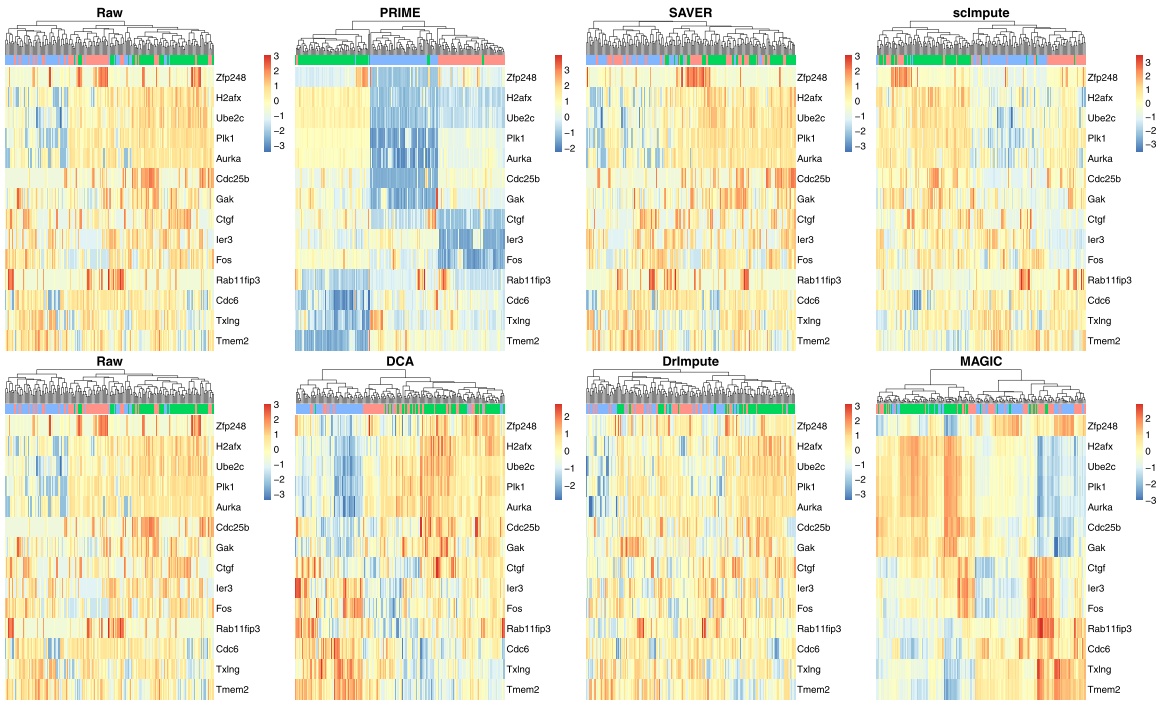

Fig. 4. Heatmap plot for the cell cycle genes over single cells at different cell cycle stages. Color indicates row-wise normalized expression values and the legend indicates the *z*-scores. The column color bar indicates the cell cycle stage; blue for G1, red for S phase and green for G2M. A clear cycling pattern is observed in PRIME imputed data. The similar pattern is only observed MAGIC but not the other imputation methods. (Color version of this figure is available at *Bioinformatics* online.)

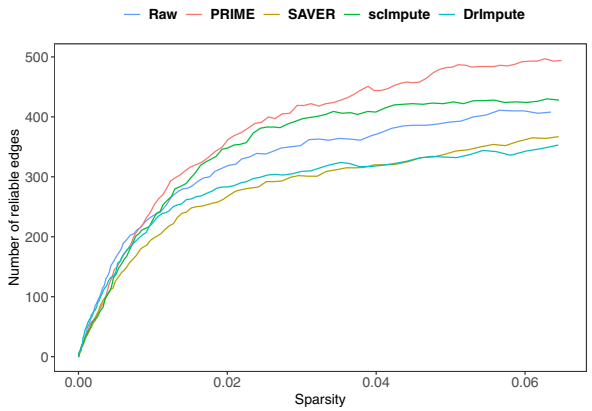

Fig. 5. Gene expression network analysis on imputed data. PRIME can detect more edges at a large range of sparsity levels. The number of reliable edges (90% reproducible in sub-sampled data) was plotted against different sparsity levels for all imputation methods

Table 2. Optimum network structures for various methods

	Singleton nodes	Edges
Raw	276	391
PRIME	212	483
SAVER	292	343
scImpute	245	425
DrImpute	298	332

by identifying more reliable edges, and it can identify 14% more reliable edges than the next best algorithm, scImpute.

Lastly, we compared the false signaling edges induced by each imputation method. To this aim, we added 40 pseudo genes that are almost constantly expressed with very small standard deviation. Among 40 pseudo genes, 20 pseudo genes have a standard deviation of 0.05 and the rest of 20 pseudo genes have a standard deviation of 0.1 for their gene expression values in log scale. We controlled the mean expression values across 758 cells ranges from 0.1 to 3.0 in a log scale with a step size of 0.1. Then, we performed the same procedure to infer a robust co-expression network and considered the inferred edges connecting to a pseudo gene as the false signaling edges. We confirmed that SAVER and scImpute induce no false signaling edges and PRIME induces a few false signaling edges. However, when we considered the number of reliable edges in the inferred co-expression network using PRIME imputation, the number of false signaling edges introduced by PRIME is negligible and there are more advantages (i.e. the number of inferred reliable edges) using PRIME to infer the stable co-expression network (Supplementary Section S8).

3.5 Improved correlation with bulk RNA sequencing

Because there tends to be less heterogeneity in single-cell RNAseq data from cell lines than from tissues, single-cell transcriptomes tend to be tightly correlated with bulk RNA sequencing when profiled on cell lines. We took advantage of this property to evaluate the accuracy of various imputation methods. In particular, we used single-cell sequencing for human ES cells, where the dataset includes 173 neuronal progenitor cells, 138 DEs, 105 endothelial cells, 69 trophoblast-like cells, 159 human foreskin fibroblasts, 212 H1 and 162 H9 human ES cells. For each of the respective cell lines, Chu *et al.* (2016) also profiled the bulk RNA samples at the same time points using Illumina single-end sequencing. Since the gene expression in bulk RNA sequencing approximates the average gene expression of cells in the tissue, bulk sequencing has greater sequencing depth and is less susceptible to dropout events. We therefore hypothesized that effective imputation would increase the ability to find gene expression correlations by effectively removing dropouts.

The correlation between raw data and bulk sequencing was low even though both were sampled from the same cell types. After reducing dropout effects, however, the correlation between bulk and single-cell sequencing clearly increased (Fig. 6). MAGIC and DCA showed the best performance, and PRIME recorded the next highest

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

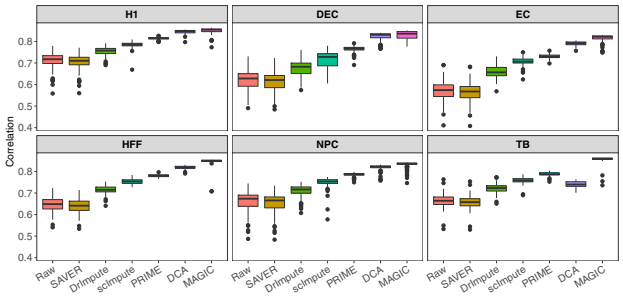

Fig. 6. Correlation between scRNAseq and Bulkseq on cell lines. A high correlation is expected between the scRNAseq and Bulk seq. This box plot shows the correlation between bulk expression and the imputed single-cell expression using MAGIC and PRIME across multiple cell types

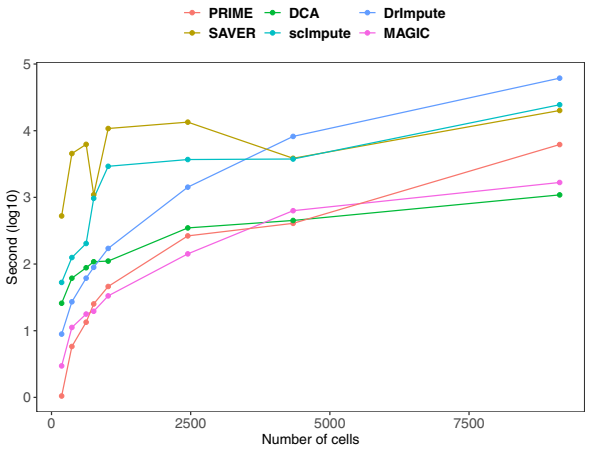

Fig. 7. Running time comparison. Six datasets were sorted based on the number of cells sequenced. The running time is plotted on the *y*-axis in logarithm scale. PRIME and MAGIC are the fastest among these methods. SAVER requires a longer running time compared to all other methods

correlation in average. Chu *et al.* (2016) also generate bulk and single-cell RNA sequencing data to produce endoderm derivative cells from human ES cells at different time points (12, 24, 36, 72 and 96 h). Supplementary Figure S11 shows the correlation between bulk and single-cell sequencing at different time points and it shows a similar trend to the Figure 6. Supplementary Figure S12 shows the genes differentially expressed between each pair of adjacent time point at different time points, and it shows that PRIME effectively recovers the gene expression that is close to the bulk sequencing. For example, NANOG, GATA4 and PRDM1 clearly show the similar expression patterns to the bulk sequencing. .

3.6 Computation time

One of the major advantages of single-cell RNA sequencing is its ability to profile thousands to millions of cells simultaneously. The computation time of the algorithm is thus an important factor to consider for large-scale single cell analysis. We compared the running time for different imputation algorithms implemented by R script (Figure 7). We utilized SAVER version 1.0, scImpute version 0.0.6 and the latest versions of MAGIC, DrImpute and DCA. In this experiment, we used a laptop computer equipped with Intel i5 3.4 GHz and 16 GB RAM. If the algorithm supports a parallelization, we utilized the maximum number of cores (i.e. 4 CPU cores). PRIME and MAGIC required the least computation time even though they utilize only a single core. Although DCA required slightly longer computation time for the small-scale datasets, it showed the least computation time as scale of dataset increases. SAVER and scImpute utilized multiple CPU cores, but they required much longer computation times and are the least scalable methods. Please note that, although it depends on the computational resource for each user, based on our simulation settings, the current version of PRIME cannot handle single-cell RNA sequencing >10 000 cells and we are working on optimize the source code.

4 Discussion

Here, we describe a probabilistic imputation method (PRIME) to reduce dropout effects in a single-cell RNA sequencing data. PRIME iteratively recovers the missing values in single-cell RNA sequencing data based on the expected gene expression in the set of cells that putatively belong to the same cell type. First, we construct a cell correspondence network through Euclidean distance and Pearson correlation, and we identify the local subnetwork (i.e. set of highly relevant cells to the imputation target cell) through an efficient random walk protocol to employ the wisdom of crowd. Finally, to decrease dropout events, we impute the gene expressions based on the mean and variance of the gene expressions in the local community. Through a comprehensive evaluation using synthetic and real-world single-cell sequencing datasets, we demonstrate that the proposed imputation method can provide better visualization in a low-dimensional space and cell type clustering, enhanced gene expression patterns and improved stability for the network inference with rapid computation times. PRIME is also compatible with other single-cell analysis methods. Since it does not change the dimension (i.e. the number of genes and cells) of the input data and it effectively recovers the dropouts in the raw count matrix, it can be directly employed within existing analysis pipelines without complicated and time-consuming manipulation. We propose that this method can be used as the preprocessing step for various single-cell analyses such as a visualization, single-cell clustering and gene expression analysis. More importantly, the proposed method does not require prior information, such as the number of cell types and cell-type specific marker genes: such information might not be available or it may require additional biological experiments. The proposed method is thus quite practical and versatile to most of the real-world single-cell sequencing studies. It also requires less computational time than the other state-of-the-art algorithms, and it is effective at dealing with large-scale single-cell datasets. Although the proposed method can effectively impute the dropout events in single-cell RNA sequencing, there are certain limitations. For example, if all the genes in a particular cell type are corrupted by dropouts, PRIME would not be able to impute the missing values because there is not enough information from the local community to recover the missing values. In fact, this is a common problem for most of the current imputation methods. To effectively address the problem and improve the accuracy of imputation results, we would integrate an effective data mining strategy with the imputation method. Then, it can automatically identify the prior information such as gene regulatory relationships and cell-type specific marker genes and accurately infer the missing values even under conditions of extreme dropout events.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (NRF-2019R1G1A1004803).

Conflict of Interest: none declared.

References

Andersen,R. *et al.* (2006) Local graph partitioning using pagerank vectors. In 47th Annual IEEE Symposium on Foundations of Computer Science, 2006 (FOCS'06). IEEE, Berkeley, CA, USA, pp. 475–486.

Blondel,V.D. *et al.* (2008) Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.*, **2008**, P10008.

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020

Bruneau,P. *et al*. (2014) A heuristic for the automatic parametrization of the spectral clustering algorithm. In *2014 22nd International Conference on Pattern Recognition (ICPR)*. IEEE, Stockholm, Sweden, pp. 1313–1318.

Buettner,F. *et al*. (2015) Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat. Biotechnol*., **33**, 155.

Chu,L.-F. *et al*. (2016) Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biol*., **17**, 173.

Darmanis,S. *et al*. (2015) A survey of human brain transcriptome diversity at the single cell level. *Proc. Natl. Acad. Sci. USA*, **112**, 7285–7290.

Eraslan,G. *et al*. (2019) Single-cell RNA-seq denoising using a deep count autoencoder. *Nat. Commun*., **10**, 390.

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

Haque,A. *et al*. (2017) A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Med*., **9**, 75.

Hashimshony,T. *et al*. (2012) Cel-seq: single-cell RNA-seq by multiplexed linear amplification. *Cell Rep*., **2**, 666–673.

Hu,Y. *et al*. (2018) Single cell multi-omics technology: methodology and application. *Front. Cell Dev. Biol*., **6**, 28.

Huang,M. *et al*. (2018) Saver: gene expression recovery for single-cell RNA sequencing. *Nat. Methods*, **15**, 539–542.

Islam,S. *et al*. (2014) Quantitative single-cell RNA-seq with unique molecular identifiers. *Nat. Methods*, **11**, 163.

Kharchenko,P.V. *et al*. (2014) Bayesian approach to single-cell differential expression analysis. *Nat. Methods*, **11**, 740.

Klein,A.M. *et al*. (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**, 1187–1201.

Kwak,I.-Y. *et al*. (2017) Drimpute: imputing dropout events in single cell RNA sequencing data. *BMC bioinformatics* 19.1 (2018), 220.

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

Liang,S.-B. and Fu,L.-W. (2017) Application of single-cell technology in cancer research. *Biotechnol. Adv*., **35**, 443–449.

Love,M.I. *et al*. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with deseq2. *Genome Biol*., **15**, 550.

Macosko,E.Z. *et al*. (2015) Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, **161**, 1202–1214.

McInnes,L. *et al*. (2018) Umap: uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426.

Meinshausen,N. and Bühlmann,P. (2006) High-dimensional graphs and variable selection with the lasso. *Ann. Stat*., 1436–1462.

Picardi,E. *et al*. (2017) Single-cell transcriptomics reveals specific RNA editing signatures in the human brain. *RNA*, **23**, 860–865.

Pierson,E. and Yau,C. (2015) Zifa: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol*., **16**, 241.

Robinson,M.D. *et al*. (2010) edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.

Tang,F. *et al*. (2011) Development and applications of single-cell transcriptome analysis. *Nat. Methods*, **8**, S6–S11.

Usoskin,D. *et al*. (2015) Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing. *Nat. Neurosci*., **18**, 145.

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

van Dijk,D. *et al*. (2018) Recovering gene interactions from single-cell data using data diffusion. *Cell*, **174**, 716–729.e27

Wang,Y. and Navin,N.E. (2015) Advances and applications of single-cell sequencing technologies. *Mol. Cell*, **58**, 598–609.

Yau,C. *et al*. (2016) pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinform*., **17**, 140.

Zappia,L. *et al*. (2017) Splatter: simulation of single-cell RNA sequencing data. *Genome Biol*., **18**, 174.

Zeisel,A. *et al*. (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, **347**, 1138–1142.

Downloaded from https://academic.oup.com/bioinformatics/article/36/13/4021/5826995 by CNRS user on 17 September 2020