Evaluating the Performance of Dropout Imputation and Clustering Methods for Single-cell RNA Sequencing Data

Junlin Xu[1,#], Lingyu Cui[2,#], Jujuan Zhuang[3], Yajie Meng[1], Pingping Bing[4], Bingsheng He[4], Geng Tian[5,6], Taoyang Wu[7], Bing Wang[8,*], Jialiang Yang[5,6,*]

[1]*College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China*

[2]*College of Life Science, Northeast Forestry University, Harbin, Heilongjiang 150000, China*

[3]*School of Science, Dalian Maritime University, Dalian, Liaoning 116026, China*

[4]*Academician Workstation, Changsha Medical University, Changsha 410219, China*

[5]*Geneis Beijing Co., Ltd., Beijing 100102, China*

[6]*Qingdao Geneis Institute of Big Data Mining and Precision Medicine, Qingdao, 266000, China*

[7] *School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K.*

[8]*School of Electrical & Information Engineering, Anhui University of Technology, Anhui 243002, China*

# The authors contributed equally to this study

* Corresponding author(s).

E-mail: yangjl@geneis.cn (Yang JL), wangbing@ustc.edu (Wang B)

Abstract

Recent advances in single-cell RNA sequencing (scRNA-seq) provide exciting opportunities for transcriptome analysis at the single-cell resolution. Clustering individual cells is a key step to reveal cell subtypes and infer cell lineage in scRNA-seq analysis. Although many dedicated algorithms have been proposed, clustering quality remains a computational challenge for scRNA-seq data, which becomes exacerbated due to excessive zero counts caused by various technical noise. To address this challenge, we assess the combinations of nine popular dropout

imputation methods and eight clustering methods using a collection of 10 well-annotated scRNA-seq datasets with different sample sizes. Our results show that imputation algorithms do typically improve the performance of clustering methods, as well as the quality of data visualization using t-Distributed Stochastic Neighbor Embedding. However, the performance of a particular combination of imputation and clustering methods may vary among datasets with different sizes. For example, the combination of single-cell analysis via expression recovery and Sparse Subspace Clustering (SSC) methods usually works well on small datasets, while the combination of adaptively-thresholded low-rank approximation and single-cell interpretation via multikernel learning (SIMLR) usually achieves the best performance on large datasets.

KEYWORDS: Single-cell RNA sequencing; Dropout imputation; Cell clustering; T-SNE; Adjusted Rand index

Introduction

Recent advances in single-cell sequencing provide a great opportunity for understanding cell-specific gene expressions, cell lineage relationships, and various important biological processes and functions at single-cell resolution [1-3]. Among them, single-cell RNA sequencing (scRNA-seq) is widely used to quantify mRNA expression in a single cell [4, 5]. However, effective analysis of scRNA-seq data remains a challenging task, as they are typically much more complicated than traditional sequencing data [6]. Indeed, because the amount of mRNA in a single cell is very small, a million-fold amplification is usually required, which leads to greater amplification noise [7]. Among these issues, the most common one includes dropout events, referring to the value of certain genes in certain cells being zero or close to zero.

A key step in scRNA-seq transcriptome profiling is to cluster individual cells to reveal cell subtypes and/or subpopulations [8, 9]. To this end, a variety of

unsupervised clustering algorithms are proposed, ranging from simple k-means clustering, hierarchical clustering [10] and its variants (e.g., RaceID [7], SC3 [11], and CIDR [12]), to density-based spatial clustering [13], subspace clustering [14], neural network [15, 16], ensemble clustering, and kernel-based methods (such as SIMLR [17])). However, effective clustering remains a computational challenge due to the high proportion of "dropouts" featured in scRNA-seq datasets. To address this, several promising imputation methods specially designed for scRNA-seq data have been developed [6, 18-21]. These methods are roughly divided into two categories: similarity-based imputation methods, which use similarities between genes and between cells to restore expression levels, and matrix-based imputation methods, which are based on the postulate that the true expression matrix is a low-rank and leverages various advanced techniques in matrix analysis [22].

In this study, we present a critical review of nine promising imputation methods and seven clustering techniques better suited for scRNA-seq data. Using a set of ten well-annotated scRNA-seq datasets, we assessed the performance of various combinations of these imputation and clustering approaches. Our results show that imputation algorithms typically improve the performance of various clustering methods, as well as the quality of data visualization using t-Distributed Stochastic Neighbor Embedding (t-SNE). In addition, the performance of a particular combination of imputation and clustering methods may vary among datasets with different sizes. Therefore, it is critical to choose an appropriate combination of imputation and clustering algorithms for obtaining high-quality clustering from scRNA-seq data. Our results suggest that using single-cell analysis via expression recovery (SAVER) + sparse subspace clustering (SSC) usually provides better clustering results for small datasets with less than 100 cells. In contrast, adaptively-thresholded low-rank approximation (ALRA) + single-cell interpretation via multikernel learning (SIMLR) typically achieves the best performance for large datasets with more than 1000 cells.

Results

The scRNA-seq data cluster evaluation framework

The cluster assessment framework outlined in **Figure 1** can be divided into the following four steps: (1) first, we preprocess the input gene expression matrix from a scRNA-seq dataset by removing rare genes and a logarithmic transformation (see Section 2); (2) next, we use the nine imputation algorithms reviewed in the last section to impute the processed expression matrix to obtain nine estimated expression matrices; (3) then, we use each of the seven clustering algorithms to cluster each of 10 expression matrices (e.g., the original one and nine imputed ones); (4) finally, we compute the NMI, ARI, HOM, and COM scores to quantify the differences between the predefined annotations of cell types and the output cluster labels from each of the 70 combinations of imputation and clustering algorithms.

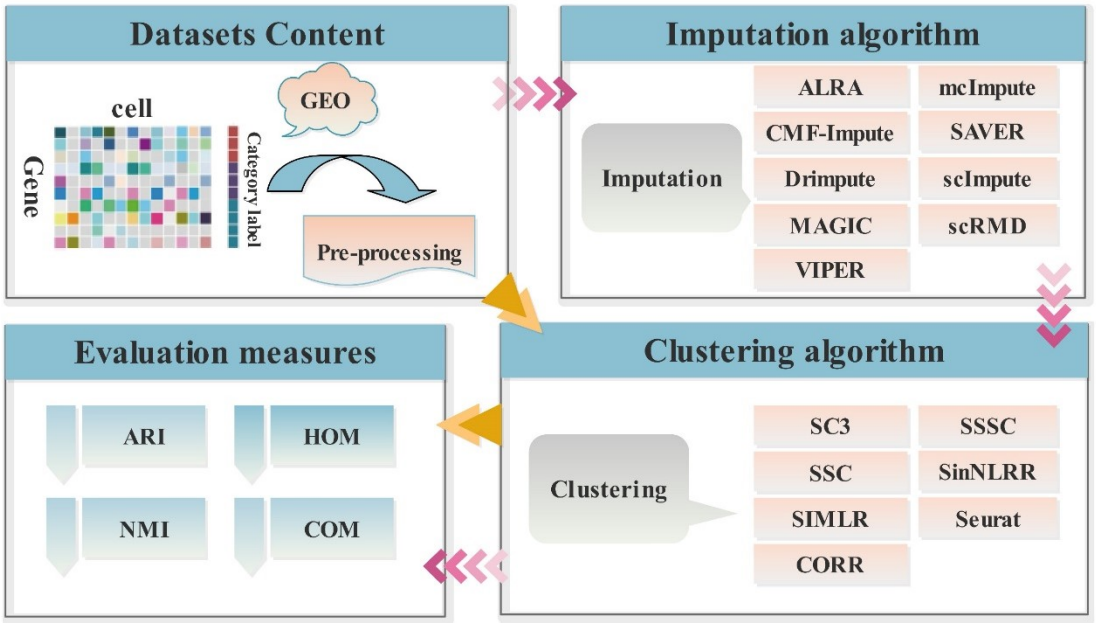

Fig. 1. Schematic workflow of the scRNA-seq dataset cluster evaluation framework. The framework is mainly divided into four parts: the collection of data sets, the direct analysis of the original data sets using different clustering algorithms, the clustering analysis of the data sets imputationed with various algorithms, and the evaluation of the clustering results.

Imputation on scRNA-seq data can often improve visualization

Finding an effective low-dimensional visualization of scRNA-seq data remains a key

computational challenge in single-cell data analysis. One popular dimensionality reduction visualization algorithm is t-SNE, which visualizes high-dimensional data by giving each data point a location in a two- or three-dimensional space. Since the t-SNE algorithm is not designed to handle the high rate of dropouts featured in scRNA-seq data, which may make this algorithm less suitable for some scRNA-seq datasets.

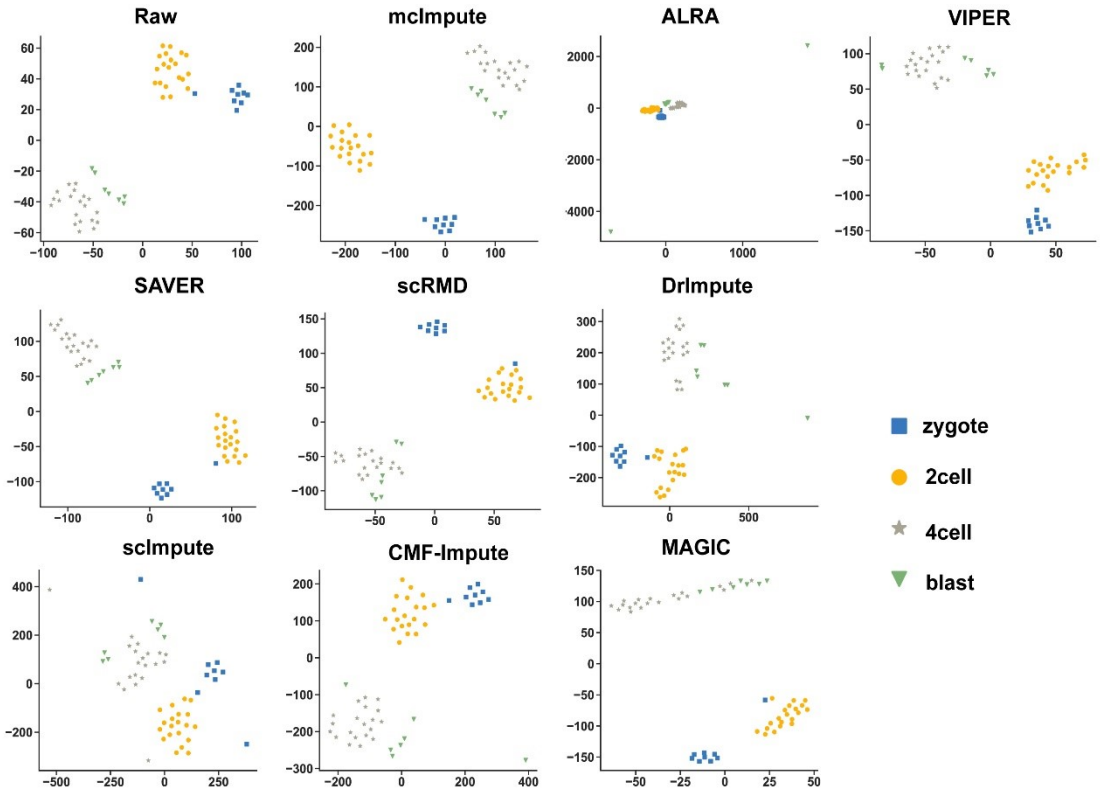

Fig. 2. T-SNE visualization of cells from the Biase scRNA-seq dataset. Among them, the mcImpute algorithm improves the visualization of the Biase dataset. Note: Cells are color-coded by the cell type annotation of the original study.

To assess the impact of imputation algorithms on visualization, we start with the smallest dataset in our collection, namely the Biase dataset [23]. As shown in **Figure 2**, both MAGIC and scImpute did not improve the visualization of the Biase dataset. Indeed, cells of type "4-cell" and those of "blast" are often confused. We also perform k-means clustering on the resulting datasets transformed by t-SNE. The evaluation results summarized in **Figure 3(A)** indicate that CMF-Impute and McImpute algorithms improved the accuracy of clustering for the Biase dataset.

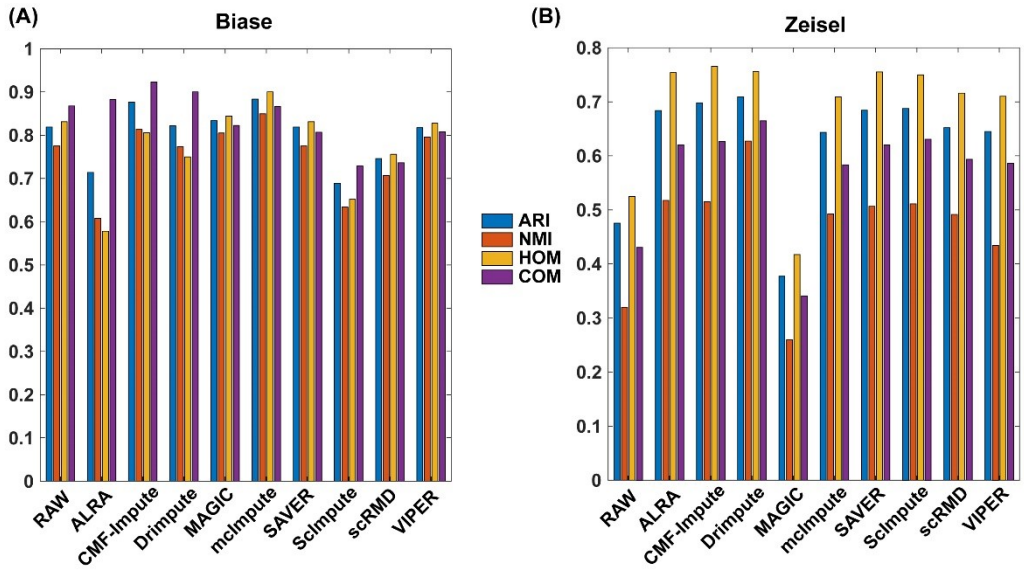

Fig. 3. Benchmark of imputation algorithms on the t-SNE+k-means clustering of scRNA-seq dataset. (A) The use of different imputation algorithms to compare the four evaluation indicators obtained on the Biase scRNA-seq dataset through t-SNE+K-means clustering. CMF-Impute and McImpute algorithms improved the accuracy of clustering. (B) The use of different imputation algorithms to compare the four evaluation indicators obtained on the Zeisel scRNA-seq dataset through t-SNE+K-means clustering. Except for MAGIC, most of the imputation algorithms can improve the accuracy of clustering.

Next, we consider the largest dataset in our collection, namely the Zeisel dataset [24]. After applying various imputation algorithms to interpolate dropout events in this dataset, we use t-SNE to visualize the imputed dataset in **Figure 4**, where the cells are color-coded according to their types annotated in the original study. This demonstrates that t-SNE typically produces better decomposition or visualization results when imputation is applied. Furthermore, we perform K-means clustering on the datasets transformed by t-SNE. The evaluation results summarized in **Figure 3(B)** indicate that, except for MAGIC, all other imputation algorithms have a desirable effect on the visualization of the Zeisel dataset.

To systematically evaluate the impact of imputation algorithms on the performance of data visualization, we apply the above evaluation framework to each of the scRNA-seq datasets in **Table 1**. As shown in the evaluation results presented in Supplementary Table S1, various imputation algorithms contribute to the visualization performance of t-SNE. Among them, SAVER, DrImpute, and CMF-Impute

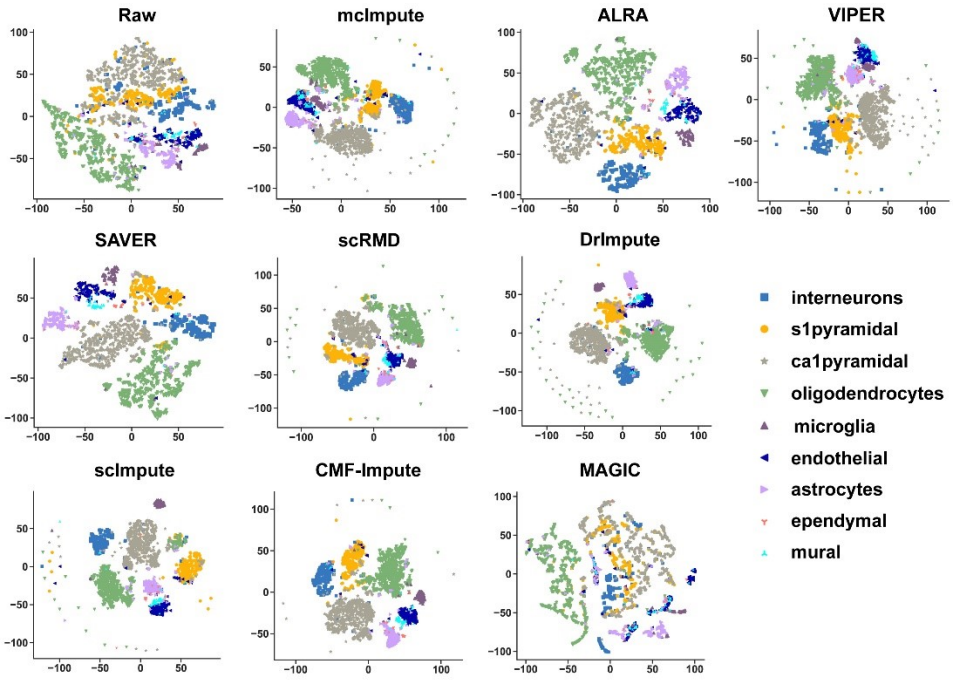

Fig. 4. T-SNE visualization of cells from the Zeisel scRNA-seq dataset. Except for MAGIC, all algorithms improve the visualization of the Zesiel dataset. Note: Cells are color-coded by the cell type annotation of the original study.

Table 1. scRNA-seq dataset for analysis and comparison.

No. of datasets	Names	No. of cells	No. of genes	No. of cell types
1	Biase	56	25,733	4
2	Deng	268	22,431	6
3	Goolam	124	41,427	5
4	Grun	251	23,459	4
5	Kolodziejczyk	704	38,615	9
6	Patel	430	5,948	5
7	Pollen	301	23,730	11
8	Usoskin	622	25,334	4
9	Yan	90	20,214	6
10	Zeisel	3,005	32,738	9

outperform other algorithms. Furthermore, the performance of some imputation algorithms may vary among the datasets, depending on the size of the datasets. For

example, ALRA performs significantly better compared to many other imputation algorithms in large datasets. However, ALRA does not work well on small datasets (such as the Deng and the Yan datasets), and all four evaluation indicators indicate poor performance. Therefore, it is important to choose an appropriate imputation algorithm for a given dataset, with the size of the dataset being a critical factor.

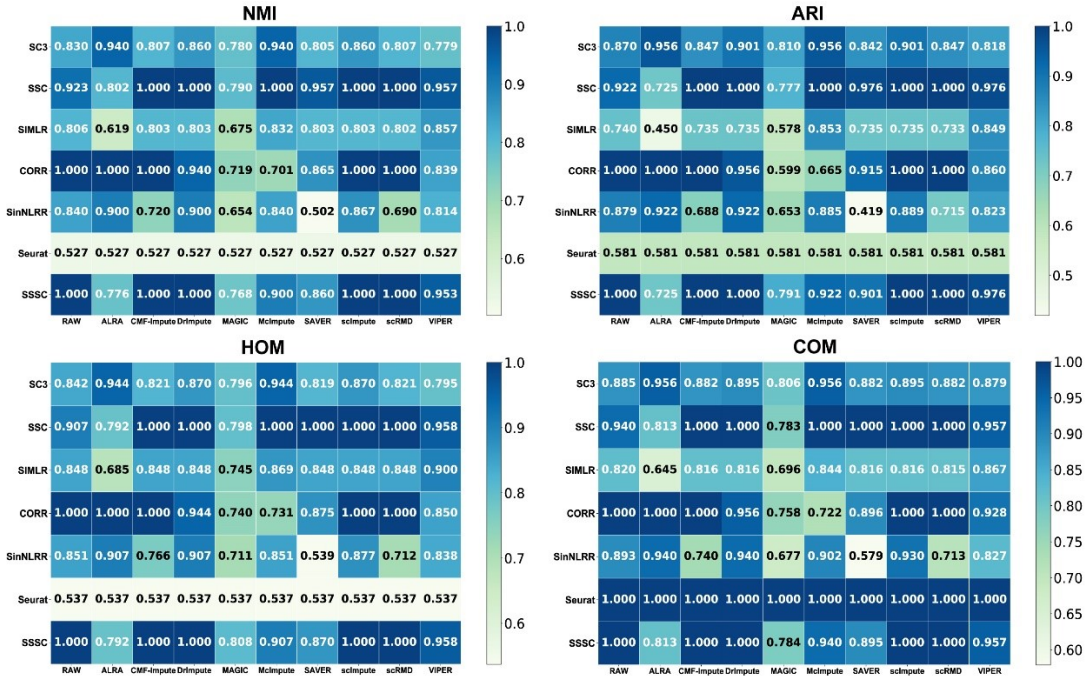

Fig. 5. Comparison of the impact of different imputation algorithms on clustering performance on the Biase scRNA-seq dataset. Among the seven clustering algorithms, SSC, Corr, and SSSC have the best overall performance.

Imputation for most scRNA-seq data can improve clustering

To assess the impact of imputation algorithms on clustering methods, we apply the framework outlined in Figure 1 to three representative datasets in Table 1. The first one is the Biase dataset, which is a small dataset containing 56 cells of four types. **Figure 5** shows the results for various combinations of imputation and clustering on the Biase dataset, which indicates that, in most cases, clustering after using various imputation algorithms can help improve performance. Among the seven clustering algorithms, SSC, Corr, and SSSC have the best overall performance.

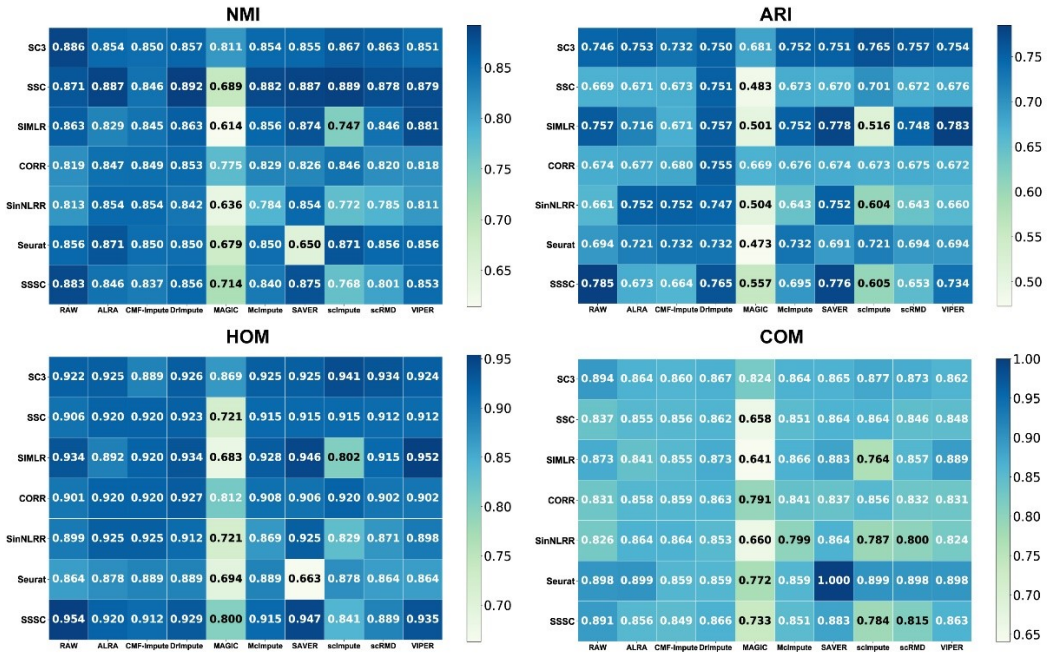

Fig. 6. Comparison of the impact of imputation algorithms on clustering performance on the Pollen scRNA-seq dataset. Different imputation algorithms are selected for different sizes of data sets to improve the accuracy of clustering.

The dataset reported by Pollen [25] and colleagues is a medium dataset with 301 cells from 11 different cell types. Compared with other datasets, the dropout rate of this dataset is relatively low. Consequently, clustering algorithms without imputation already achieve good performance. As shown in **Figure 6**, imputation algorithms do not necessarily lead to improvements in the performance of various clustering algorithms. On the contrary, some imputation algorithms (for example, MAGIC) indeed have an undesirable impact on the performance of most clustering algorithms.

The Zeisel dataset is a large dataset containing 3005 single cells from the mouse cortex and hippocampus, collected by the unique molecular identifier technology and divided into nine categories. **Figure 7** shows the performance of various combinations of imputation and clustering algorithms. For this dataset, most imputation algorithms can improve the performance of various clustering algorithms. For example, after using the ALRA algorithm to repair the missing values, the SIMLR algorithm has much better performance, achieving an ARI score of 0.827 and an NMI score of 0.774, which is much higher than that of other imputation algorithms.

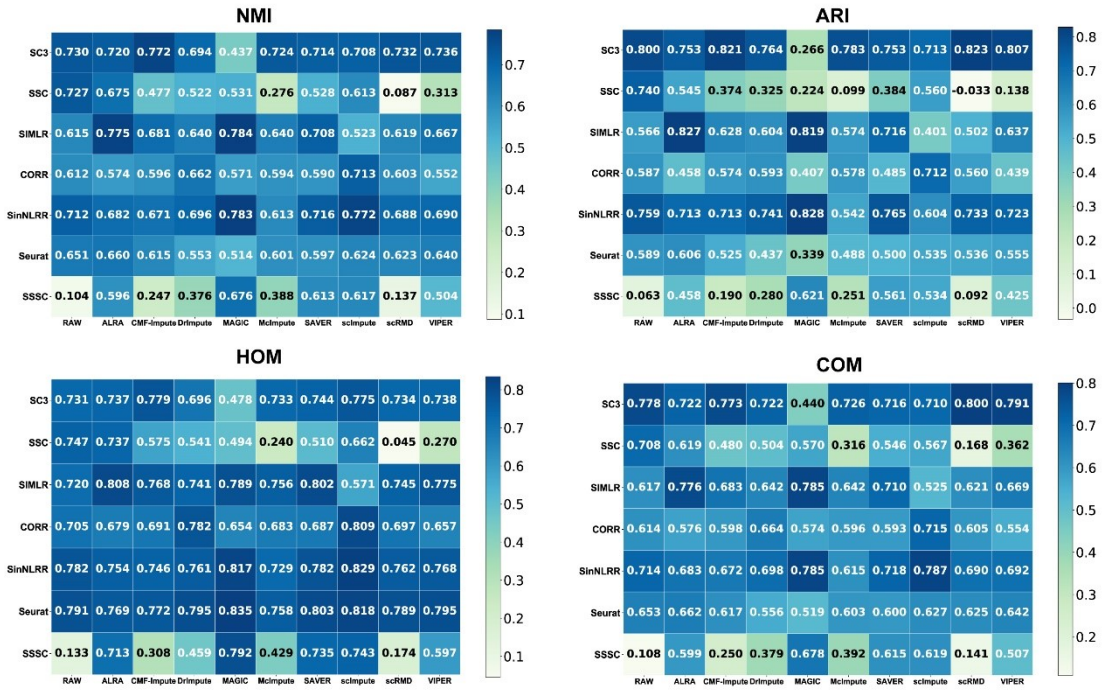

Fig. 7. Comparison of the impact of imputation algorithms on clustering performance on the Zeisel scRNA-seq dataset

Finally, we use the framework outlined in Figure 1 to assess the performance of various combinations of imputation and clustering methods for all other datasets in Table 1. As shown in Supplementary Table S2-S11, clustering methods generally have a better performance when an imputation algorithm is applied. Since the performance of each combination of imputation and clustering methods may vary among datasets with different sizes. Based on these results, we observe that a combination of SAVER and SSC usually performs well on small datasets (such as the Biase and the Yan datasets [26]). In contrast, combining adaptively-thresholded low-rank approximation ARLA and SIMLR typically achieves the best performance for large datasets (such as the Zeisel dataset).

Discussion

In this study, we use 10 well-annotated scRNA-seq datasets and an objective assessment framework to evaluate the performance of various combinations of imputation and clustering algorithms that are suitable for scRNA-seq datasets. Our empirical results show that imputation algorithms typically improve the performance

of various clustering methods, as well as the quality of data visualization using t-SNE. However, the performance of a particular combination of imputation and clustering methods may vary among datasets with different sizes. These results provide concrete choices for how to choose an appropriate combination of imputation and clustering algorithms for obtaining high-quality clustering from scRNA-seq data. Moreover, it remains interesting to see how to utilize the insights obtained from here to design better imputation and clustering algorithms.

In addition to this clustering problem, many other computational problems in single-cell data analyses also face the same challenge derived from a high dropout rate, such as standardization, differential expression analysis, and cell cycle identification. Therefore, algorithms aiming to solve those problems could benefit from various imputation techniques reviewed here, and the assessment framework proposed here can be naturally extended to study their performance.

Methods

Data preparation and preprocessing

To determine the impact of different imputation algorithms on the clustering of individual cells to their corresponding cell types, we collected 10 scRNA-seq datasets with cell type annotations, which come from the National Center for Biotechnology Information Gene Expression Omnibus (NCBI-GEO). Table 1 summarizes these 10 scRNA-seq datasets. At the same time, to reduce the technical noise in the scRNA-seq datasets, genes expressed in less than or equal to two cells were filtered [27]. In order to prevent the effect of highlighting genes with higher expression and weakening the remaining genes, we used Equation 1 to perform a logarithmic transformation with pseudo count 1 on the original expression data of single cells before analysis.

$$X = log_2(X + 1)\,. \qquad (1)$$

Dropout imputation

An important technical flaw in scRNA-seq data is the introduction of “dropout” events [28, 29]. Deletion events usually refer to the incorrect quantification of genes that are not expressed due to transcripts that are introduced during the reverse transcription step or have low expression levels [30]. A large number of studies have shown that simply deleting the less expressed genes and then normalizing them cannot completely solve this issue in scRNA-seq data analysis. In order to better perform downstream analysis of scRNA-seq data, a large number of missing value repair algorithms have been proposed. Therefore, in this study, we analyzed in detail the impact of nine better imputation algorithms on the clustering analysis of scRNA-seq data.

scImpute

Li and Li proposed a three-step approach called scImpute to determine and impute values that are affected by dropout events in scRNA-seq data [31]. Since this method uses information of the same gene from similar cells to impute missing values, the first step is to construct a candidate pool of neighboring cells for each cell, which is achieved by principal component analysis (PCA) and spectral clustering. The second step computes the dropout probability of each gene in each cell. To this end, the expectation–maximization (EM) algorithm is utilized to estimate a gamma-normal mixture model. In the final step, a separate regression model for each cell is constructed to impute the expression of genes with high dropout probabilities, for which information about the same genes in its neighboring cells identified in the first step is used.

It is demonstrated that scImpute can automatically identify zero values of high dropout probabilities and only perform imputation on these values without introducing new deviations to the remaining data [31]. Furthermore, the method can also detect outlier cells and exclude them from imputation. Evaluation based on simulated and real human and mouse scRNA-seq datasets indicates that scImpute is an effective tool for restoring transcriptome dynamics masked by dropouts. scImpute

can detect possible deletions, enhance the aggregation of cell subpopulations, improve the accuracy of differential expression analysis, and help the study of gene expression kinetics. Because scImpute requires the true number of cell subpopulations in the data a priori, this is not friendly to unknown structured data.

DrImpute

Gong et al. designed a simple and fast hot-platform imputation method called DrImpute to estimate missing events in scRNA-seq data [27]. Similar to scImpute, DrImpute performs cell clustering before imputation and also borrows information of the same gene from similar cells to impute missing values. DrImpute takes a consensus approach to obtain a more robust estimate. First, the clustering method used in DrImpute is single-cell consensus (SC3) clustering, which, as we review in the next subsection, is a consensus approach. Second, imputation is performed multiple times using different unit clustering results. Finally, the multiple estimates are averaged to get the final imputation. Specifically, let H be the number of cluster configurations (for example, the combination of distance metric and the number of clusters used in clustering), and C_1, C_1,..., C_H be the clustering results, one for each configuration. Assuming that the clustering of C_h is a true hidden cell classification result, the expected value of the dropout event can be obtained by averaging the entries in a given cell cluster:

$$E(X_{ij}|C_h) = mean(X_{ij}|W), \quad (2)$$

where X is the input matrix, and W represents X_{ij} in the same cell group in cluster C_h.

Therefore, the final calculation of the estimated drop events X_{ij} and $E(X_{ij})$ can be calculated by a simple average:

$$E(X_{ij}) = mean\left(E(X_{ij}|C)\right) = \frac{1}{H}\sum_{h=1}^{H} E(X_{ij}|C_h). \quad (3)$$

Experimental results show that DrImpute greatly improves several existing statistical tools including SC3, t-SNE, and Monocle, which cannot solve dropout events in the three most popular research areas in scRNA-seq analysis, namely cell clustering,

visualization, and lineage reestablishment. However, since the cluster number is usually unknown in DrImpute, the results are not as accurate as expected.

VIPER

Both scImpute and DrImpute perform cell clustering before imputation, using cells belonging to the same subgroup of cells for imputation. However, the cell subpopulations used in this type of imputation algorithm are often not true cell subpopulations, which cause serious deviations in the imputation results. To address this concern, Chen and Zhou proposed a simple, accurate, unadjusted, and computationally effective scRNA-seq imputation method called VIPER [32]. Compared to scImpute and DrImpute, VIPER mainly borrows information among cells with similar expression patterns to estimate the expression measurement value in a target cell. This is achieved by using a sparse non-generative regression model to actively select the sparse local set of local neighborhoods that best predicts the target cell. These sparse unit sets are selected in a progressive manner, and the attribution weights associated with them are estimated in the final estimation step to ensure robustness and computational scalability. In addition, VIPER uses cell type-specific and gene-specific methods to model the deletion probability, which clearly illustrates the uncertainty of the zero-value expression measurement in scRNA-seq. VIPER uses efficient auxiliary programming algorithms to infer all modeling parameters from existing data while maintaining low computational costs. The key feature of VIPER is that the imputed data can retain the gene expression variability of the whole cell. Compared to several existing imputation methods in several actual analysis experiments based on scRNA-seq data, VIPER can obtain higher imputation accuracy.

MAGIC

One key challenge in many imputation algorithms for scRNA-seq data (e.g., scImpute and DrImpute reviewed above) is to accurately find neighborhoods of similar cells. To

address this issue, van Dijk et al. developed a cell map imputation algorithm called Markov affinity-based graph imputation of cells (MAGIC) [33], which mainly uses data diffusion to share information among similar cells to denoise the cell count matrix and fill in missing transcripts. Central to this approach is a Markov matrix M, which is derived using a Gaussian kernel and a normalization process from an appropriate cell-by-cell distance matrix constructed from the input data. Once the Markov matrix, whose (i,j)-entry represents the probability of transitioning from cell i to cell j in a single diffusion step, is obtained, a data diffusion step is performed through exponentiation of M to identify neighborhoods of similar cells. Then the imputation step of MAGIC involves sharing information between cells in the resulting neighborhoods through matrix multiplication.

Since MAGIC leverages the observation that cell phenotypes can often be roughly embedded in a low-dimensional structure corresponding to the low-frequency trend of the data containing the biological signal of interest, experiments have shown that it can effectively alleviate the sparsity and noise caused by random mRNA captures and reveal gene–gene relationships in scRNA-seq data. Moreover, unlike many other imputation algorithms that only fill in "missing values," MAGIC uses the value diffusion between similar pixels along the affinity-based graph structure to correct the entire data matrix and connect it to the basic manifold structure. However, imputation on a low-dimensional space will likely eliminate gene expression variability across cells and thus abolish a key feature of single-cell sequencing data.

SAVER

Huang et al. developed an expression restoration method for scRNA-seq data called SAVER [34], which mainly uses information between genes and cells to estimate missing values and improve the expression estimation of all genes. SAVER aims to restore true gene expression patterns by eliminating technical differences and retaining biological differences. It uses observed gene counts to form a prediction model for each gene, and then uses the observed counts and the weighted average of

the prediction to estimate the true expression of the gene. Experimental results show that SAVER can reliably restore cell-specific gene expression concentration, cross-cell gene expression distribution, and gene-to-gene and cell-to-cell expression. SAVER's powerful performance is attributed to its adaptive estimation of discrete parameters at the gene level and its cross-validation-based model selection, which can prevent unnecessary model complexity. But SAVER relies on a Markov chain Monte Carlo algorithm to tune all parameters, which is computationally costly and might not be scalable to large datasets

ALRA

Linderman et al. proposed a highly scalable method called ALRA to recover the true expression level of scRNA-seq data [22]. It is a singular value decomposition (SVD) followed by a thresholding scheme that takes advantage of the non-negativity of the true expression matrix. A key assumption used in this approach is that the underlying true matrix is non-negative and low-rank and contains many zeros, but none of these zeros are associated with dropouts. The observed matrix from a scRNA-seq experiment is then sparser as many values are incorrectly measured as zero due to the dropout effect. Consequently, ALRA uses SVD to find the best k approximation of this matrix, then transforms it into an imputed matrix where each element corresponding to a dropout is not zero. Experimental results show that ALRA improves the separation between cell types in the high-dimensional space of original cells and restores the true expression of marker genes while retaining the biological zero position. As ALRA has only one parameter, the approximate rank k of the matrix, which is automatically selected based on the statistical information of the interval between consecutive singular values, the method is widely applicable to various scRNA-seq datasets.

scRMD

Chen et al. developed a single-cell RNA sequence imputation method based on robust

matrix decomposition (RMD) called scRMD [20]. A key postulate in this approach is that a gene expression matrix Y has the following decomposition:

$$Y = L - S + E, \quad (4)$$

where L is a low-rank matrix, S is a sparse matrix, and E represents the combined effect of measurement errors and random fluctuations.

Moreover, this method also leverages the observation that the expression of gene i observed in cell j is less likely to be affected by the dropout if the value Y_{ij} is large enough. Formally, the index set of candidate dropouts can be represented by $\Omega = \{(i,j): y_{ij} \leq c\}$ for a threshold constant c. Then we have $\mathcal{P}_\Omega(S) \geq 0$, where the mask operator $\mathcal{P}_\Omega$ is defined as $\mathcal{P}_\Omega(s_{ij}) = s_{ij}$ if index (i,j) Is contains in Ω and $\mathcal{P}_\Omega(s_{ij}) = 0$ otherwise. Then the scRMD model can be formulated as the following optimization problem:

$$\min_{L,S} \frac{1}{2}\|Y - L + S\|_F^2 + \lambda_1\|L\|_* + \lambda_2\|S\|_1 \quad \text{s.t.} \quad \mathcal{P}_\Omega(S) \geq 0, \mathcal{P}_{\Omega^c}(S) = 0 \; and \; L \geq 0 \quad (5).$$

Here, λ_1 and λ_2 are regularization parameters, $\Omega^c = \{(i,j): y_{ij} > c\}$. Moreover, $\|\cdot\|_*$, $\|\cdot\|_F$, and $\|\cdot\|_1$ represent the nuclear norm, Frobenious norm, and elementwise l_1 norm of a matrix, respectively. This optimization problem can be effectively solved by an alternating direction multiplier method (ADMM). Extensive data analysis shows that scRMD can accurately restore missing values and help improve downstream analyses, such as differential expression analysis and cluster analysis.

mcImpute

Mongia et al. presented an imputation algorithm based on matrix completion. The postulate used in this algorithm is similar to that in scRMD [18]; that is, the gene expression matrix Y is a low-rank matrix. Let $\mathcal{P}_Y$ be the mask operator associated with Y, which is defined as $\mathcal{P}_Y(X_{ij}) = X_{ij}$ if $Y_{ij} > 0$ and $\mathcal{P}_Y(X_{ij}) = 0$ otherwise. Then the optimization problem can be formulated as:

$$\min \|X\|_* \; s.t. \; \|Y - \mathcal{P}_{\Upsilon}(X)\|_F^2 < error. \quad (6)$$

This problem is solved in mcImpute by iteratively limiting the singular values of the expression matrix. Compared with many other imputation algorithms, one distinguishing feature of mcImpute is that it does not assume any distribution of genes and maintains a complete biological silent expression value. Experiments using several real datasets show that mcImpute is competitive with other algorithms in improving the accuracy of cell clustering, identifying differentially expressed genes, enhancing the separability of cell types, and improving dimensionality reduction. But the complexity of mcImpute algorithm is very high and the running time is very long

CMF-Impute

Xu et al. proposed a novel method based on collaborative matrix decomposition to estimate missing items in a given scRNA-seq expression matrix [21]. A key step in the CMF-Impute algorithm is to find two characteristic matrices so that their product provides the best approximation to the original matrix. Specifically, for a gene expression matrix Y with g rows and n columns, the CMF-Impute algorithm seeks to find a k-dimensional cell feature matrix W and a k-dimensional gene feature matrix H such that $Y = WH^T$ and $k \ll \min(g, n)$. Noting that similar cells tend to have similar gene expression patterns, CMF-Impute explicitly incorporates a cell-to-cell similarity matrix S_c and a gene-to-gene similarity matrix S_g into its optimization formulation:

$$\min_{W,H} \|Y - WH^T\|_F^2 + \lambda_1 \|W\|_F^2 + \lambda_2 \|H\|_F^2 + \lambda_c \|S_c - WW^T\|_F^2 + \lambda_g \left\|S_g - HH^T\right\|_F^2, \quad (7)$$

where $\lambda_1, \lambda_2, \lambda_c, \lambda_g$ are regularization parameters.

Experiments on several simulated and real scRNA-seq datasets show that CMF-Impute improves the performance of existing cell clustering algorithms and methods for reconstructing cell-to-cell, gene-to-gene correlations, and inferring cell lineage trajectories.

Clustering techniques

Cell type identification based on single-cell sequencing data is one of the key computational challenges in single-cell biology and has thus received widespread attention [35, 36]. In this section, we review the application of eight clustering methods to scRNA-seq data.

t-SNE + K-means

K-means clustering is one of the most frequently used cluster analysis methods. It iteratively computes the mean of all data points of each class as the center point of the class and assigns each data point into one of the k clusters whose mean is closest to the given data point. As a consequence, two data points that are closer to each other are more likely to be classified into the same category. The K-means algorithm and its variants have been applied in a number of fields. Within the area of single-cell technologies, many single-cell clustering algorithms use K-means, such as the SC3 [11] and pcaReduce [37]. However, one may argue that the most popular method is a two-step combination of the t-SNE method and K-means clustering: First, t-SNE is used to reduce the dimensionality of the data from a high-dimensional space to a 3d- or 2d-plot. Next, the K-means method is used for clustering the processed dataset whose dimension is reduced.

SC3

Kiselev et al. developed a method called SC3 for determining cell types based on transcriptome profiles alone, achieving high accuracy and robustness by combining multiple clustering solutions through a consensus approach [11]. The workflow of SC3 can be grouped into the following five steps: (1) filter out rare genes and common genes to reduce the dimensionality of the data; (2) construct three distance matrices between cells using the Euclidean, Pearson, and Spearman metrics; (3) transform all distance matrices using either PCA or by calculating the eigenvectors of the associated Laplacian; the columns of the resulting matrices are then sorted in

ascending order by their corresponding eigenvalues; (4) perform K-means clustering on the first d eigenvectors of the transformed distance matrices; (5) obtain a hierarchical clustering from the consensus matrix, which is constructed using the cluster-based similarity partitioning algorithm (CSPA). Each clustering result is represented by a binary similarity matrix, and the consensus matrix is calculated by averaging all similarity matrices.

A major bottleneck of SC3 is its longer running time compared to other models. Furthermore, for dealing with even larger datasets, SC3 implements a hybrid approach that combines unsupervised and supervised methodologies, which have a possible limitation of rare cell types not being identified.

SSC

The SSC algorithm is a novel approach to the subspace clustering problem using sparse representation. It is particularly useful for clustering data drawn from multiple low-dimensional subspace embedded in a high-dimensional space, a feature common to many scRNA-seq datasets. The SSC algorithm solves the subspace clustering problem in two steps. The first one is to solve the following global sparse optimization problem:

$$min \|C\|_1 \ s.t. \ X = XC \ and \ diag(C) = 0, \quad (8)$$

where $X \in R^{D \times N}$ represents the input matrix.

The output matrix $C \in R^{N \times N}$ is a block diagonal matrix in which the nonzero block corresponding to data points in the same subspace. In the second step, this information about the membership of data points is utilized in the spectral clustering framework to obtain predicted labels.

Although SSC performs well in many applications, it ignores the constraint relationship between the coefficient matrix and the clustering result, which is a major shortcoming of the algorithm. To alleviate this problem, several improved SSC variants have been proposed, including the SSSC described below.

SSSC

As an improvement to SSC [38], the structural sparse subspace clustering (SSSC) algorithm proposes a unified joint optimization framework, which not only obtains the spectral clustering result through the sparse optimization step but also constrains the coefficient matrix by the clustering result in turn. To this end, an incidence matrix $Q = [q_1 \dots q_n] \in R^{N\times n}$ is constructed to associate each data point with the subspace that contains it using the clustering result from a previous iteration. Since each data point belongs to one subspace, we have $Q\mathbf{1}$=$\mathbf{1}$ and $rank(Q) = n$, where $\mathbf{1}$ is the vector of all ones of appropriate dimension. Consequently, the set of all feasible incidence matrices is:

$$\mathbb{Q} = \{Q \in \{0,1\}^{N\times n} : Q\mathbf{1} = \mathbf{1}, rank(Q) = n\}. \quad (9)$$

Using a subspace structure norm defined as:

$$\|C\|_Q = \sum_{i,j} |C_{ij}| \left(\frac{1}{2}\left\|q^i - q^j\right\|^2\right), \quad (10)$$

where q^i and q^j are the i-th and j-th rows of the matrix Q.

The subspace clustering problem is reformulated in SSSC into solving the following optimization problem:

$$\min_{C,Q} \|C\|_Q + \|C\|_1 \quad s.t.\ X = XC, diag(C) = 0, \text{and } Q \in \mathbb{Q}. \quad (11)$$

This optimization problem can be solved efficiently via a combination of an alternating direction method of multipliers (ADMM) with spectral clustering.

Seurat

Butler et al. developed a comprehensive R package, which is an indispensable tool in the field of single cell RNA-seq analysis. [39]. This toolkit provides a number of functions including t-SNE dimensionality reduction analysis, cluster analysis, differential expression, construction of developmental trajectories, mark gene recognition and so on. For this work, we are mainly interested in the cluster analysis module, which is used to identify cell subtypes. Instead of a direct cluster analysis applied to all cells, Seurat first performs a PCA to select the principal components with the largest contribution, and then uses the selected principal components to

perform cluster analysis. The clustering algorithm includes original Louvain algorithm (The default), Louvain algorithm with multilevel refinement, SLM, and Leiden. The t-SNE dimensionality reduction technique is also employed to display expression distributions of cells in a 2d-plot, where cells in the same cluster are coded by the same color.

SIMLR

Wang et al. developed a novel similarity-learning framework called SIMLR [17], which learns an appropriate distance metric from combining multiple types of distances between cells. SIMLR assumes that cells in the same subpopulation are more similar, and the similarity matrix should have an approximate block diagonal structure in which the number of blocks is determined by the number of separable subpopulations of the input cells. In the default implementation of SIMLR, Gaussian kernels, which generate the best empirical performance among a number of candidate kernels, take the form:

$$\mathrm{K}(c_i, c_j) = \frac{1}{\epsilon_{ij}\sqrt{2\pi}} \exp\left(-\frac{\|c_i - c_j\|_2^2}{2\epsilon_{ij}^2}\right). \quad (12)$$

Here $\| c_i - c_j \|_2$ is the Euclidean distance between cell i and cell j, and the variance ϵ_{ij} can be calculated with different scales:

$$\mu_i = \frac{\sum_{l \in KNN(c_i)} \|c_i - c_j\|_2}{k}, \ \epsilon_{ij} = \frac{\sigma(\mu_i + \mu_j)}{2}, \quad (13)$$

where $KNN(c_i)$ represents cells that are top k neighbors of cell i.

SIMLR uses learned similarities to visualize cells, reduce the dimensionality of the input data, and cluster cells into subgroups, giving priority to genes with the highest variability that can explain differences in the entire population. Since the implementation of SIMLR needs the number of clusters as an input, it is not suitable for analyzing data with unknown structure.

SinNLRR

Based on similarity learning, Zheng et al. proposed a scRNA-seq cell-type detection method called SinNLRR [40], where non-negative and low-rank structures on the

similarity matrix are imposed. This leads to an optimization problem with the form:

$$min\frac{1}{2}\|X - XC\|_F + \lambda\|C\|_* \quad s.t.\ C \geq 0, \qquad (14)$$

where X is the input matrix, and C is a coefficient matrix in which the entry $C_{i,j}$ denotes the confidence of cells i and j in the same subpopulation.

SinNLRR applies the alternating direction method of the multiplier (ADMM) to solve the optimization problem and proposes an adaptive penalty selection method to avoid sensitivity to parameters. The learned similarity matrix can be visualized, and Laplace scores can be used to prioritize gene markers. SinNLRR is benchmarked with ten human and mouse scRNA-seq datasets, whose sizes range from dozens to thousands of cells. SinNLRR obtained stronger robustness and more accurate results using different datasets. At present, the main goal of SinNLRR is to reduce the running time on large-scale scRNA-seq data.

Corr

By introducing a new similarity measure named differentiability correlation, Jiang et al. proposed a hierarchical clustering-based algorithm called Corr to predict cell types [32]. Differentiability correlation evaluates the similarity between any two cells by using the correlation between the gene expression profiles of two cells and incorporating information from all other cells. Since the relationship of cell-specific gene expression patterns over the whole cell population is considered, this novel measure turns out to be more robust against cell heterogeneity and data noise. Using the framework of hierarchical clustering, Corr incorporates factorial ANOVA in optimal cluster number determination, which allows the number of clusters to be automatically determined. Corr is benchmarked with several real scRNA-seq datasets, with outstanding performance and a correct cluster number obtained for each dataset.

Evaluations of clustering

To benchmark the performance of various clusters through the imputation algorithm used, four clustering evaluation indicators are chosen to quantify the clustering

performance on each scRNA-seq dataset. Formally, let $P = \{p_1, p_2, \ldots, p_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ represent the real cell type from the dataset and the cell type generated by clustering algorithm, respectively. The dissimilarity between P and T can then be measured by one of the following four indicators: normalized mutual information (NMI) [41], adjusted Rand index (ARI) [42], homogeneity (HOM) [43], and completeness (COM) [43].

NMI

The NMI score between P and T is defined as:

$$NMI(P,T) = \frac{MI(P,T)}{\sqrt{H(P)H(T)}}, \quad (15)$$

where $H(P)$ and $H(T)$ denote the entropy of P and T, respectively, and $MI(P,T)$ represents the mutual information between them. It is well known that NMI has an upper bound of 1 and lower bound of 0.

ARI

To define the ARI score, cell pairs in the dataset are classified into one of the following four types: the number of cell pairs that are in the same cluster in both P and T denoted by N_{11}; the number of cell pairs that are in different clusters in both P and T by N_{00}; the number of cell pairs that are in the same cluster in P but in different clusters in T by N_{10}; the number of cell pairs that are in different clusters in P but in the same cluster in T by N_{01}. The ARI score between P and T is then defined as:

$$ARI(P,T) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00}+N_{01})(N_{01}+N_{11}) + (N_{00}+N_{10})(N_{10}+N_{11})}. \quad (16)$$

The ARI score is bounded above by 1 and equals 0 when the Rand index is the same as its expected value (under the generalized hypergeometric distribution for randomness).

HOM

One desired property for the output clustering T is that it satisfies the homogeneity

criteria; that is, each cluster in T contains only cells from a single cluster in P. To measure how close the clustering T to this ideal situation, the homogeneity score (HOM) is defined as:

$$HOM_P(T) = 1 - \frac{H(P|T)}{H(P)}. \qquad (17)$$

Here $H(P|T)$ is the entropy of P conditioned on T. Note that for a perfectly homogeneous clustering T, we have $H(P|T) = 0$, and hence its homogeneity score is 1. We also use the convention that when P contains only one cluster, that is, $H(P) = 0$, the homogeneity score is always 1. Then the HOM score is between 0 and 1, with close to 1 being desirable. However, being homogeneous alone is not sufficient for good clustering. For instance, the trivial clustering T in which each cluster contains only one cell always has a homogeneity score of 1. To deal with such cases, we will consider one additional score based on completeness.

COM

To some extent, completeness is a property that is symmetrical to homogeneity. That is, if each cluster in P contains only units from a single cluster in T, then cluster T is complete. To measure how close the clustering T to this ideal situation, the completeness score (COM) is defined as:

$$COM_P(T) = 1 - \frac{H(T|P)}{H(T)}. \qquad (18)$$

Similar to the homogeneity score, here we also use the convention that the completeness score is 1 when T contains only one cluster. Furthermore, the completeness score is between 0 and 1, with 1 being desirable. Note that the homogeneity score and the completeness score run roughly in opposition: a high homogeneity score often means a low completeness score. Hence, a clustering that is high on both the homogeneity and the completeness scores is truly desirable because it indicates that the clustering is indeed rather consistent with the golden standard.

Authors' contributions

Junlin Xu: Writing - original draft, Conceptualization, Data curation, Methodology, Visualization. Lingyu Cui: Writing - review & editing, Investigation, Formal analysis. Jujuan Zhuang: Investigation, Data curation. Yajie Meng: Visualization, Writing - review & editing. Pingping Bing: Investigation, Data curation. Bingsheng He: Investigation, Data curation. Geng Tian: Data curation, Validation. Taoyang Wu: Investigation, Writing - review & editing. Jialiang Yang: Writing - review & editing, Conceptualization, Supervision. All authors read and approved the final manuscript.

Competing interests

JY and GT are currently employed by Geneis (Beijing) Co. Ltd.; All other authors have declared no competing interests.

Acknowledgements

This work was supported by the Hunan Provincial Innovation Foundation for Postgraduate of China (Grant No. CX20200434) awarded to Junlin Xu, and the National Natural Science Foundation of China (Grant No. 62172004) to Bing Wang. We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

References

1. Jaitin DA, Kenigsberg E, Keren-Shaul H et al. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types, Science 2014;343:776-779.
2. Kolodziejczyk AA, Kim JK, Svensson V et al. The technology and biology of single-cell RNA sequencing, Molecular cell 2015;58:610-620.
3. Stegle O, Teichmann SA, Marioni JC. Computational and analytical challenges in single-cell transcriptomics, Nature Reviews Genetics 2015;16:133-145.
4. Zappia L, Phipson B, Oshlack A. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database, PLoS computational biology 2018;14:e1006245.
5. Yang J, Liao B, Zhang T et al. Editorial: Bioinformatics Analysis of Single Cell Sequencing Data and Applications in Precision Medicine, Front Genet 2019;10:1358.

6. Zhang L, Zhang S. Comparison of computational methods for imputing single-cell RNA-sequencing data, IEEE/ACM transactions on computational biology and bioinformatics 2018.
7. Grün D, Muraro MJ, Boisset J-C et al. De novo prediction of stem cell identity using single-cell transcriptome data, Cell stem cell 2016;19:266-277.
8. Buettner F, Natarajan KN, Casale FP et al. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells, Nature biotechnology 2015;33:155-160.
9. Peng L, Tian X, Tian G et al. Single-cell RNA-seq clustering: datasets, models, and algorithms, RNA Biol 2020;17:765-783.
10. Tasic B, Menon V, Nguyen TN et al. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics, Nature neuroscience 2016;19:335-346.
11. Kiselev VY, Kirschner K, Schaub MT et al. SC3: consensus clustering of single-cell RNA-seq data, Nature methods 2017;14:483-486.
12. Lin P, Troup M, Ho JW. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data, Genome biology 2017;18:59.
13. Dey KK, Hsiao CJ, Stephens M. Visualizing the structure of RNA-seq expression data using grade of membership models, PLoS genetics 2017;13:e1006599.
14. Zhuang J, Cui L, Qu T et al. A streamlined scRNA-Seq data analysis framework based on improved sparse subspace clustering, IEEE Access 2021;PP:1-1.
15. Kim DH, Marinov GK, Pepke S et al. Single-cell transcriptome analysis reveals dynamic changes in lncRNA expression during reprogramming, Cell stem cell 2015;16:88-101.
16. Macosko EZ, Basu A, Satija R et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets, Cell 2015;161:1202-1214.
17. Wang B, Zhu J, Pierson E et al. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning, Nature methods 2017;14:414-416.
18. Mongia A, Sengupta D, Majumdar A. McImpute: Matrix completion based imputation for single cell RNA-seq data, Frontiers in genetics 2019;10:9.
19. Badsha MB, Li R, Liu B et al. Imputation of single-cell gene expression with an autoencoder neural network, Quantitative Biology 2020:1-17.
20. Chen C, Wu C, Wu L et al. scRMD: Imputation for single cell RNA-seq data via robust matrix decomposition, Bioinformatics 2020;36:3156-3161.
21. Xu J, Cai L, Liao B et al. CMF-Impute: an accurate imputation tool for single-cell RNA-seq data, Bioinformatics 2020;36:3139-3147.
22. Linderman GC, Zhao J, Kluger Y. Zero-preserving imputation of scRNA-seq data using low-rank approximation, bioRxiv 2018:397588.
23. Biase FH, Cao X, Zhong S. Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing, Genome research 2014;24:1787-1796.
24. Zeisel A, Muñoz-Manchado AB, Codeluppi S et al. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq, Science 2015;347:1138-1142.
25. Pollen AA, Nowakowski TJ, Shuga J et al. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex, Nature biotechnology 2014;32:1053.
26. Yan L, Yang M, Guo H et al. Single-cell RNA-Seq profiling of human preimplantation

722 embryos and embryonic stem cells, Nature structural & molecular biology 2013;20:1131.
723 27. Gong W, Kwak I-Y, Pota P et al. DrImpute: imputing dropout events in single cell RNA
724 sequencing data, BMC bioinformatics 2018;19:1-10.
725 28. Zhang L, Zhang S. PBLR: an accurate single cell RNA-seq data imputation tool considering
726 cell heterogeneity and prior expression level of dropouts, bioRxiv 2018:379883.
727 29. Qiu P. Embracing the dropouts in single-cell RNA-seq analysis, Nature communications
728 2020;11:1-9.
729 30. Ye P, Ye W, Ye C et al. scHinter: imputing dropout events for single-cell RNA-seq data with
730 limited sample size, Bioinformatics 2020;36:789-797.
731 31. Li WV, Li JJ. An accurate and robust imputation method scImpute for single-cell RNA-seq
732 data, Nature communications 2018;9:1-9.
733 32. Jiang H, Sohn LL, Huang H et al. Single cell clustering based on cell-pair differentiability
734 correlation and variance analysis, Bioinformatics 2018;34:3684-3694.
735 33. van Dijk D, Nainys J, Sharma R et al. MAGIC: A diffusion-based imputation method reveals
736 gene-gene interactions in single-cell RNA-sequencing data, bioRxiv 2017:111591.
737 34. Huang M, Wang J, Torre E et al. SAVER: gene expression recovery for single-cell RNA
738 sequencing, Nature methods 2018;15:539-542.
739 35. Kim T, Chen IR, Lin Y et al. Impact of similarity metrics on single-cell RNA-seq data clustering,
740 Briefings in bioinformatics 2019;20:2316-2326.
741 36. Petegrosso R, Li Z, Kuang R. Machine learning and statistical methods for clustering
742 single-cell RNA-sequencing data, Briefings in bioinformatics 2020;21:1209-1223.
743 37. Yau C. pcaReduce: hierarchical clustering of single cell transcriptional profiles, BMC
744 bioinformatics 2016;17:140.
745 38. Elhamifar E, Vidal R. Sparse subspace clustering: Algorithm, theory, and applications, IEEE
746 transactions on pattern analysis and machine intelligence 2013;35:2765-2781.
747 39. Satija R, Butler A, Hoffman P. Seurat: Tools for Single Cell Genomics, R package version
748 2018;2.
749 40. Zheng R, Li M, Liang Z et al. SinNLRR: a robust subspace clustering method for cell type
750 detection by non-negative and low-rank representation, Bioinformatics 2019;35:3642-3650.
751 41. Strehl A, Ghosh J. Cluster ensembles---a knowledge reuse framework for combining
752 multiple partitions, Journal of machine learning research 2002;3:583-617.
753 42. Hubert L, Arabie P. Comparing partitions, Journal of classification 1985;2:193-218.
754 43. Rosenberg A, Hirschberg J. V-measure: A conditional entropy-based external cluster
755 evaluation measure. In: Proceedings of the 2007 joint conference on empirical methods in natural
756 language processing and computational natural language learning (EMNLP-CoNLL). 2007, p.
757 410-420.

758

759 Figure legends

760 Figure 1. Schematic workflow of the scRNA-seq dataset cluster evaluation
761 framework

762

Figure 2. T-SNE visualization of cells from the Biase scRNA-seq dataset

Note: Cells are color-coded by the cell type annotation of the original study.

Figure 3. Benchmark of imputation algorithms on the t-SNE+k-means clustering of scRNA-seq dataset

(A) The use of different imputation algorithms to compare the four evaluation indicators obtained on the Biase scRNA-seq dataset through t-SNE+K-means clustering. **(B)** The use of different imputation algorithms to compare the four evaluation indicators obtained on the Zeisel scRNA-seq dataset through t-SNE+K-means clustering.

Figure 4. T-SNE visualization of cells from the Zeisel scRNA-seq dataset

Note: Cells are color-coded by the cell type annotation of the original study.

Figure 5. Comparison of the impact of different imputation algorithms on clustering performance on the Biase scRNA-seq dataset

Figure 6. Comparison of the impact of imputation algorithms on clustering performance on the Pollen scRNA-seq dataset

Figure 7. Comparison of the impact of imputation algorithms on clustering performance on the Zeisel scRNA-seq dataset

Supplementary material

Supplementary Table S1. Evaluate T-SNE visualization performance of different impute algorithms on 10 scRNA-seq datasets

Supplementary Table S2. Compare the performance of clustering algorithms on 10 scRNA-seq datasets

Supplementary Table S3. Compare the effects of ARLA on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S4. Compare the effects of CMF-Impute on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S5. Compare the effects of DrImpute on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S6. Compare the effects of MAGIC on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S7. Compare the effects of mcImpute on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S8. Compare the effects of SAVER on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S9. Compare the effects of scImpute on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S10. Compare the effects of scRMD on various clustering performances on 10 scRNA-seq datasets

Supplementary Table S11. Compare the effects of VIPER on various clustering performances on 10 scRNA-seq datasets