Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research Article

scIALM: A method for sparse scRNA-seq expression matrix imputation using the Inexact Augmented Lagrange Multiplier with low error

Xiaohong Liu, Han Wang *, Jingyang Gao *

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, 100029, China

ARTICLE INFO

Keywords:
scRNA-seq
Inexact Augmented Lagrange Multiplier
Sparse matrix imputation
Dropout imputation

ABSTRACT

Single-cell RNA sequencing (scRNA-seq) is a high-throughput sequencing technology that quantifies gene expression profiles of specific cell populations at the single-cell level, providing a foundation for studying cellular heterogeneity and patient pathological characteristics. It is effective for developmental, fertility, and disease studies. However, the cell-gene expression matrix of single-cell sequencing data is often sparse and contains numerous zero values. Some of the zero values derive from noise, where dropout noise has a large impact on downstream analysis. In this paper, we propose a method named scIALM for imputation recovery of sparse single-cell RNA data expression matrices, which employs the Inexact Augmented Lagrange Multiplier method to use sparse but clean (accurate) data to recover unknown entries in the matrix. We perform experimental analysis on four datasets, calling the expression matrix after Quality Control (QC) as the original matrix, and comparing the performance of scIALM with six other methods using mean squared error (MSE), mean absolute error (MAE), Pearson correlation coefficient (PCC), and cosine similarity (CS). Our results demonstrate that scIALM accurately recovers the original data of the matrix with an error of 10e-4, and the mean value of the four metrics reaches 4.5072 (MSE), 0.765 (MAE), 0.8701 (PCC), 0.8896 (CS). In addition, at 10%-50% random masking noise, scIALM is the least sensitive to the masking ratio. For downstream analysis, this study uses adjusted rand index (ARI) and normalized mutual information (NMI) to evaluate the clustering effect, and the results are improved on three datasets containing real cluster labels.

1. Introduction

Single-cell sequencing technology [1] is an improvement based on next-generation sequencing technology [2] (also known as second-generation sequencing technology), which focuses on researching the genome [3], transcriptome [4], epigenome [5], and proteome [6] of individual cells. Single-cell RNA sequencing is one of its mainstream technologies, alongside extensive research on single-cell multi-omics analysis [7]. Single-cell transcriptome sequencing has also gradually entered the high-throughput era. In comparison to traditional sequencing technologies, scRNA-seq can process tens of thousands of single-cell data at the same time, which makes the gene sequencing analysis from macroscopic to microscopic and provides a foundation for further research on cellular heterogeneity [8,9]. It can also be used to analyze cell developmental trajectories as well as the study of diseases [10–12].

Although scRNA-seq provides gene expression at the single cell level, its expression matrix often contains a significant amount of zero-value noise, resulting in the sparse expression matrix, which is often referred to as "zero expression" [13]. Zero expression can arise from two main factors [14]: 1. biological phenomenon, where certain genes are indeed not expressed in the corresponding cells; 2. technical reasons, causing low-expression genes are not detected, known as dropout events [15]. Dropout events may occur due to low sequencing depth or unsuccessful reverse transcription of certain genes. Single-cell transcriptome data contain a variety of noise, and a particularly prominent source is dropout events. These dropout events can significantly impact the downstream analysis of the cells, including dimensionality reduction and clustering [16], developmental trajectory [17], gene differential expression [18], and gene regulatory network inference [19].

In recent years, many methods have emerged to recover dropout events in scRNA-seq data. DCA [20] (deep count autoencoder network) is a notable approach, which employs a zero-inflated negative binomial distribution model (ZINB) and an autoencoder to denoise the data. MAGIC [21] utilizes a Markov affinity-based cell graph, which shares information between similar cells through data diffusion to eliminate dropout noise. In addition, scVI [22], scImpute [23], SAVER [24] and

* Corresponding authors.
E-mail addresses: liuxiaohongexec@163.com (X. Liu), 2021200811@buct.edu.cn (H. Wang), gaojy@mail.buct.edu.cn (J. Gao).

https://doi.org/10.1016/j.csbj.2023.12.027
Received 25 October 2023; Received in revised form 21 December 2023; Accepted 22 December 2023
Available online 2 January 2024

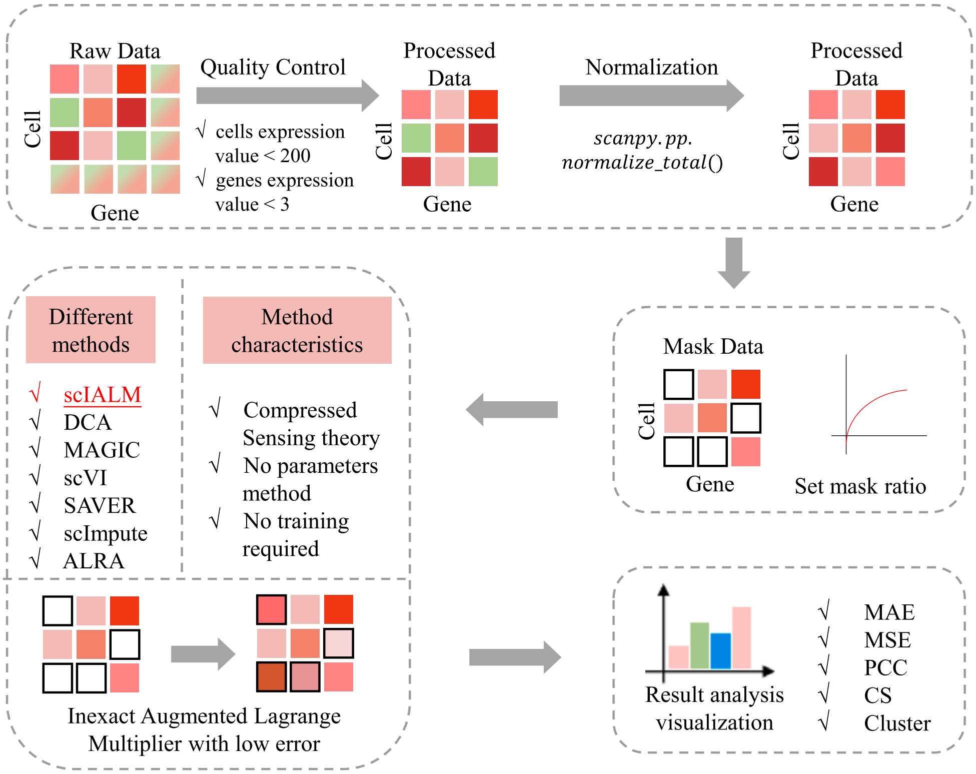

Fig. 1. The framework of scIALM.

ALRA [25] are also prominent imputation methods currently in the field. scVI also uses a deep learning approach. scImpute is a statistical method that learns the dropout probability of each gene in each cell based on a mixed model and then imputes it with information from the same gene in other similar cells. SAVER is an expression recovery method for scRNA-seq data based on the Unique Molecular Index (UMI) [26]. ALRA uses an adaptive threshold low-rank approximation, which takes advantage of the non-negativity and low-rank structure of the expression matrix to selectively impute technical zeros. These methods may introduce new biases into the real data, relying on similarities in cell or gene expression that may lose some of the global information. We analyzed scRNA-seq data, leaving aside the bioinformatics background, which is essentially a matrix completion problem [27]. In addition, the matrix has the characteristics of sparsity, so we have to recover unknown entries in the matrix using few real data.

The concept of sparsity originates from compressed sensing theory [28], which was applied in image processing initially, such as image denoising [29,30] and medical image MRI [31,32], and gradually extended to many fields other than images. Compressed sensing theory proposes that a one-dimensional signal y can be reconstructed from far fewer samples than the traditional sampling theorem (Nyquist Sampling Theorem). It means that the original data can be reconstructed with high probability using a small number of observations [33]. The three core issues of compressed sensing theory are sparse representation of signals [34,35], design of observation matrices, and reconstruction algorithms. In many practical problems, the data are two-dimensional matrices. In such cases, the low rank of the matrix space (i.e., the singular value of the matrix is sparse) is considered as sparsity of the vector space, and the theory of compressed sensing can be extended to the matrix completion theory. Matrix completion theory focuses on recovering the entire matrix from partially known elements. The problem of imputing dropout events in scRAN-seq data aligns with the theory of matrix completion. It assumes that the original matrix is low-rank [25,36], and its three core issues are the low-rank property of the matrix, incoherent characteristics, and reconstruction algorithms. At present, the reconstruction algorithms for compressed sensing and matrix completion have been extensively studied and we can transform this problem into convex optimization problems to solve. The augmented Lagrange multiplier (ALM) has been shown to converge more accurately and efficiently than the singular value threshold (SVT) algorithm and the accelerated proximal gradient (APG) algorithm [37].

The key of scIALM is to make a reasonable assumption that the expression matrix of scRNA-seq is low-rank. The critical step of the algorithm is to calculate the leading singular values and singular value vectors of the matrix through singular value decomposition (SVD) [38] to obtain a low-rank approximate representation of the matrix. By augmented Lagrange multiplier method, we can transform the imputation of the expression matrix into a convex optimization problem. This paper improves an inexact augmented Lagrange multiplier algorithm (IALM) [36]: 1. No longer uniformly sample the sparse original matrix, but directly use the non-uniform matrix as input data; 2. By calculating the ratio between continuous singular values $(svd_i/svd_i + 1)$ in the matrix, we can determine the upper bound of the prediction dimension by dividing the singular values into two groups. The singular values of the scRNA-seq matrix are not grouped, so this ratio is no longer used as the basis for updating matrix rank. These improvements make IALM suitable for scRNA-seq, and scIALM can reconstruct the original data with a high probability and effectively recover unknown data from dropout events. The framework of scIALM is shown in Fig. 1.

2. Materials and methods

The data of scRNA-seq can be converted into a cell-gene expression matrix after quality control and mapping. By mapping the position of each read in the file (SAM/BAM file), we obtain the expression of genes. The preprocessed expression matrix, which is numerical, serves as the input data for this method (scIALM) [39]. Although we cannot definitively verify the low-rank nature of the gene expression matrix, we can reasonably assume that the matrix to be recovered is low-rank [25]. One of the central issues in matrix completion is low rank, and scIALM will recover the original matrix with the lowest possible rank. Under this assumption, the rank of the matrix is predicted by iteration, and the purpose of iteration is to impute unknown entries without introducing new biases.

2.1. Basic knowledge

When dealing with high-dimensional data, it is commonly assumed that the data lies in proximity to a low-dimensional linear subspace. We usually use principal component analysis (PCA) to estimate this low-dimensional subspace, i.e., dimension reduction [40]. Given a matrix $D \in R^{m*n}$ with $D = A + E$, the mathematical model for estimating the low-dimensional subspace is described as finding a low-rank matrix A and a noise matrix E such that the difference between A and D is minimized.

However, when the noise within the data is large and sparse, classical PCA is no longer suitable. In such instances, the problem can be solved by modeling the following convex optimization problem:

$$\underset{A,E}{min} ||A||_* + \lambda ||E||_1, \ subject\ to\ D = A + E \tag{1}$$

where $|| \cdot ||_*$ is the kernel norm of the matrix, $|| \cdot ||_1$ is the sum of the absolute values of the matrix terms, and λ is a positive weighted parameter. The method for solving such an optimization problem is called Robust PCA (RPCA) [41], which has been applied in problems such as background modeling and image recovery. Subsequently, the application of iterative thresholding (IT) to solve the optimization problem (1) overcame its limitations in practical applications, and the accelerated proximal gradient (APG) algorithm further improved the speed of iteration [42,43].

2.2. Augmented Lagrange multiplier

In general, the Lagrange multiplier method is commonly used to solve constrained optimization problems of the following type:

$$min\ f(X), \ subject\ to\ h(X) = 0 \tag{2}$$

where $f : R^n \rightarrow R$, $h : R^n \rightarrow R^m$. For the above optimization problem with equality constraints, the Lagrange multiplier method can be used to transform it into an unconstrained optimization problem, and the main idea is to introduce the Lagrange multiplier Y and transform it into the following problem to solve:

$$L(X, Y) = f(X) + < Y, h(x) > \tag{3}$$

where $< \cdot, \cdot >$ is the inner product of a matrix or a vector. On the basis of the Lagrange multiplier method, we add the penalty term μ, and the convergence speed of the algorithm will increase. According to [19] its augmented Lagrange function is defined as follows:

$$L(X, Y, \mu) = f(X) + < Y, h(x) > + \frac{\mu}{2} ||h(X)||_F^2 \tag{4}$$

where μ is a positive scalar, called the penalty parameter, $\frac{\mu}{2}||h(X)||_F^2$ is the penalty term, and $|| \cdot ||_F$ is the Frobenius norm. Under general conditions, when $\{\mu_k\}$ is an increasing sequence and f and h are continuous differentiable functions, it has been proven in [44] that the Lagrange multiplier Y_k converges linearly to the optimal solution. Therefore, the above optimization problem (2) can be solved by the augmented Lagrange multiplier method, and the general approach is summarized in Algorithm 1.

For equation (1), we apply the Lagrange multiplier method to solve the RPCA, which is transformed into the following equation (5). The solving process is called the exact augmented Lagrange multiplier method (EALM), and the general method is outlined in Algorithm 2.

$$L(A, E, Y, \mu) = ||A||_* + \lambda ||E||_1 + < Y, D - A - E > + \frac{\mu}{2} ||D - A - E||_F^2 \tag{5}$$

Algorithm 1 General Method of Augmented Lagrange Multiplier

```
$\rho \geq 1$.
while not converged do
    $X_{k+1} = arg\ \underset{x}{min}\ L(X, Y_k, \mu_k)$;
    $Y_{k+1} = Y_k + \mu_k h(X_{k+1})$;
    $\mu_{k+1} = \rho \mu_k$.
end while
Output: $X_k$.
```

Algorithm 2 requires solving the sub-problem $(A_{k+1}^*, E_{k+1}^* = arg\ \underset{A,E}{min}\ L(A, E, Y_k^*, \mu_k)$, which is a slower convergence process. It turns out that we don't have to solve the sub-problem exactly, but just updating A_k and E_k once is sufficient, and it is the inexact ALM method (IALM).

Algorithm 2 Exact Augmented Lagrange Multiplier (EALM)

```
Input: $D \in R^{m*n}$, $\lambda$.
Initialize : $Y_0^*; \mu_0 > 0; \rho > 1; k = 0$
while not converged do
    $(A_{k+1}^*, E_{k+1}^*) = arg\ \underset{A,E}{min}\ L(A, E, Y_k^*, \mu_k)$;
    $Y_{k+1}^* = Y_k^* + \mu_k(D - A_{k+1}^* - E_{k+1}^*)$;
    $\mu_{k+1} = \rho \mu_k$;
    $k \leftarrow k + 1$.
end while
Output: $(A_k^*, E_k^*)$.
```

For low-rank matrix recovery [45], there is the following formulation: given an $m * n$ matrix M,

$$\underset{x}{min}\ rank(X), \ subject\ to\ X_{i,j} = M_{i,j}, \ \forall (i,j) \in \Omega \tag{6}$$

where Ω is the index set of the matrix M, i, j are the index coordinates, and the resulting matrix X (called the completion of M) has a rank no more than the prescribed integer. The optimization problem in (6) is non-convex, and a common method is to use the kernel norm to approximate the rank of the matrix. The literature [46] shows that the kernel norm is the optimal convex relaxation of rank in a certain sense. Thus, the minimized rank problem ($\underset{x}{min}\ rank(X)$) can be transformed into a minimized matrix kernel norm ($\underset{A}{min}\ ||A||_*$).

For scRNA-seq data with dropout noise, using sparse but clean (accurate) known data to recover the rest of the matrix entries, we formulate this MC problem as follows:

$$\underset{A}{min}\ ||A||_*, \ subject\ to\ A + E = D, \ \pi_\Omega(E) = 0 \tag{7}$$

where $\pi_\Omega : R^{m*n} \rightarrow R^{m*n}$, Ω is the index set of matrix entries in D. D is the input matrix, i.e., the preprocessed cell-gene expression matrix, and its unknown entry data is 0. A is the output matrix, and E represents noise.

There is an equality constraint in (7), so according to (2) and (4), its augmented Lagrangian function is:

$$L(A, E, Y, \mu) = ||A||_* + < Y, D - A - E > + \frac{\mu}{2} ||D - A - E||_F^2 \tag{8}$$

For MC problems, the inexact augmented Lagrange multiplier method is described in Algorithm 3. The flowsheet of scIALM is shown in Fig. 2. The code of scIALM has been given in Github: https://github.com/lxh07/scIALM.

Where ρ is the hyperparameter associated with the μ update, and their initialization is not the focus of this study. μ is a positive number, and the value of ρ makes $\{\mu_k\}$ an increasing sequence. sv is the predicted rank of the matrix A, and R is the increasing value of sv during each iteration. The convergence criterion is $\frac{||D - A_k - E_k||_F}{||D||_F} < \varepsilon$.

Since there is an equality constraint in this problem: $D - A - E = 0$, this paper uses the augmented Lagrange multiplier method to transform it into an unconstrained problem and finally outputs a low-rank matrix. This low-rank matrix does not have an exact rank but arrives at

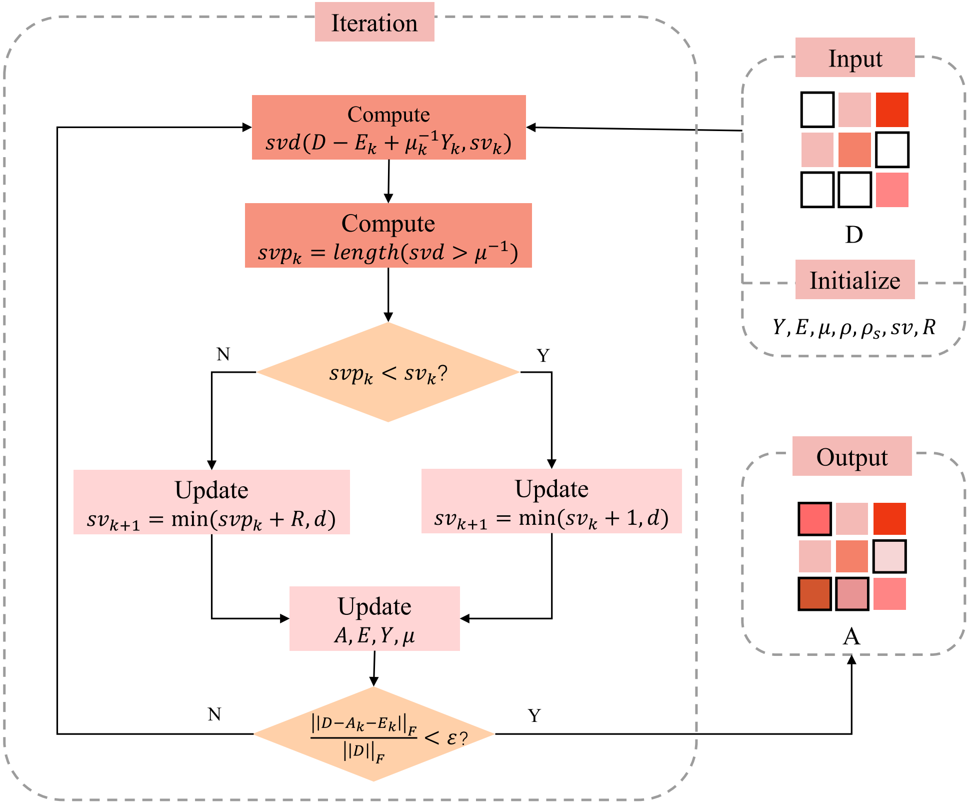
Iteration
Compute
$svd(D - E_k + \mu_k^{-1}Y_k, sv_k)$
Compute
$svp_k = length(svd > \mu^{-1})$
N
$svp_k < sv_k?$
Y
Update
$sv_{k+1} = \min(svp_k + R, d)$
Update
$sv_{k+1} = \min(sv_k + 1, d)$
Update
A, E, Y, μ
N
$\frac{||D-A_k-E_k||_F}{||D||_F} < \varepsilon?$
Y
Input
D
Initialize
$Y, E, \mu, \rho, \rho_s, sv, R$
Output
A

Fig. 2. The flowsheet of scIALM.

Algorithm 3 Inexact Augmented Lagrange Multiplier (IALM) for MC

```
Input: $D \in R^{m*n}$.
 1: Initialize : $Y_0^*; E; \mu_0 > 0; \rho > 1; k = 0; sv_0; R$
 2: while not converged do
 3:     $//Lines\ 4-6\ solve\ A_{k+1} = arg \min_A L(A, E_k, Y_k, \mu_k)$.
 4:     $Update\ sv_k$;
 5:     $(U, S, V) = svd(D - E_k + \mu_k^{-1}Y_k)$;
 6:     $A_{k+1} = US_{\mu_k^{-1}}[S]V^T$.
 7:     $//Lines\ 8\ solve\ E_{k+1} = arg \min_{\pi_\Omega(E)=0} L(A_{k+1}, E, Y_k, \mu_k)$.
 8:     $E_{k+1} = \pi_{\bar{\Omega}}(D - A_{k+1} + \mu_k^{-1}Y_k)$.
 9:     $Y_{k+1} = Y_k + \mu_k(D - A_{k+1} - E_{k+1})$.
10:     $\mu_{k+1} = \rho\mu_k$.
11:     $k \leftarrow k + 1$.
12: end while
Output: $(A_k, E_k)$.
```

the lowest possible rank by prediction. The only condition for the algorithm to stop iteration is that the original data in matrix D is accurately recovered (the recovery error ε is set to 1e-4). The unknown items are imputed during the original data recovery process.

The descriptions and values of the parameters and formulas in Fig. 2 are shown in Table 1.

2.3. Metrics

Due to the random and unpredictable occurrence of dropout events, it is impossible to provide a real data benchmark. So to evaluate the experimental effect, we randomly mask (using standard normal distribution) the expression matrix at different ratios to simulate dropout events. This paper uses the following four metrics to evaluate the imputation effect of the masking position:

① Mean square error (MSE):

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(x_i - y_i)^2 \tag{9}$$

Table 1
The items in Fig. 2.

Item	Description	Value
Y	Lagrange multiplier	Y_0 = zero matrix
E	Noise matrix	E_0 = zero matrix
μ	Parameter of the penalty item	$\mu_0 = \frac{0.3}{max(singular\ values)}$
ρ_s	Non-zero ratio of the matrix	Computation
ρ	$\mu_{k+1} = \rho\mu_k$	$\rho = 1.1 + 2.5\rho_s$
svp_k	Number of singular values greater than μ_k^{-1}	Computation
sv_k	Predicted rank of A	$sv_0 = 5$
R	Increasing value of sv	200
d	Dimension of D	-
ε	Stopcriterion	1e-4
$svd(D - E_k + \mu_k^{-1}Y_k, sv_k)$	Calculate the first sv_k singular values	-
$length(svd > \mu_k^{-1})$	Calculate svp_k	-

② Mean absolute error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{10}$$

③ Pearson correlation coefficient (PCC):

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{11}$$

④ Cosine similarity (CS):

$$cos(\theta) = \frac{A \bullet B}{||A||||B||} = \frac{\sum_{i=1}^{N} A_i \times B_i}{\sqrt{\sum_{i=1}^{N}(A_i)^2} \times \sqrt{\sum_{i=1}^{N}(B_i)^2}} \tag{12}$$

where x_i represents the data after imputing by scIALM, y_i represents the real data in the expression matrix (i.e., the original data of the

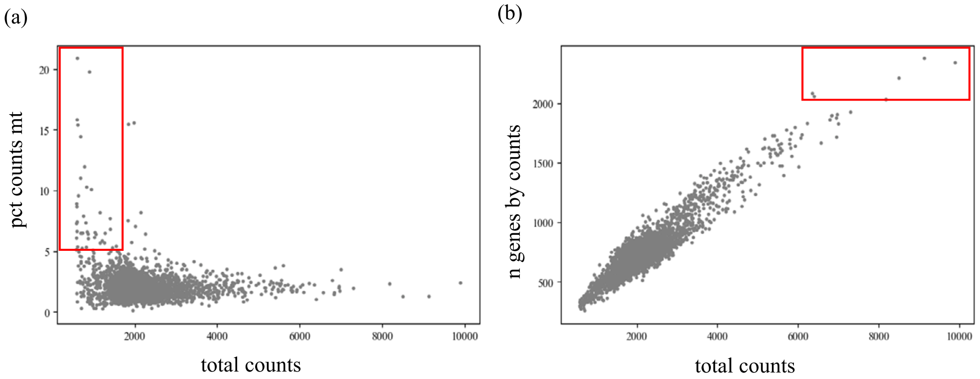

Fig. 3. (a) shows the mitochondrial gene counts and total expression counts of the PBMC dataset. (b) shows the gene counts and total expression counts of the PBMC dataset. The red boxes indicate outliers.

masking position); $\bar{x}$ and $\bar{y}$ represent the corresponding mean values, respectively; A and B represent the gene expression profile after imputing and the real gene expression profile, both of them are vectors, and A_i and B_i represent the corresponding i-th values. MSE and MAE reflect whether the gene expression value is the same before and after imputation, the value of both ranges between $[0, +\infty]$, the smaller the value is, the closer the expression value is. MSE and MAE are auxiliary evaluation indicators. The former reflects the degree of difference between the real and the imputed value but is sensitive to anomalies (outliers). The latter better reflects the reality of the errors. PCC and CS are used to measure whether the data expression trend is the same before and after imputation, the value of both ranges between $[-1, 1]$, and the closer to 1, the more consistent the expression trend. PCC and CS are the primary evaluation metrics.

In downstream analysis, we use the adjusted rand index (ARI) and normalized mutual information (NMI) to measure clustering results.

2.4. Datasets

The expression matrix is a $mCells * nGenes$ scale. Each row represents different cells, and each column represents different genes. Due to the presence of numerous zero values, the expression matrix exhibits sparsity, some of which originate from dropout events.

In this paper, we use the following four different real datasets for experiments, and the number of cells and genes contained before quality control was indicated.

(1) Human Frozen Peripheral Blood Mononuclear Cells (PBMCs) from 10X GENOMICS, containing 2900 cells and 32738 genes.

(2) MOUSE embryo cell analysis published by Klein (GSE6-5525), containing 2717 cells and 24021 genes. [47]

(3) MOUSE Brain cells published by Chen (GSE87544). It analyzes the mouse hypothalamic cell diversity, containing 14437 cells and 23284 genes. [48]

(4) Mouse Brain cells published by Campbell (GSE93374). It uses Drop-seq technology to perform single-cell analysis on adult mouse brain cells, containing 21,086 cells and 26,774 genes. [49]

2.5. Data preprocessing

The cell-gene expression matrix is very sparse, which means that it contains a significant number of zero values. To facilitate subsequent analysis, we have to perform Quality Control (QC) on the expression matrix [50]. There are three primary QC metrics [51]: the total number of transcript molecules measured, the total number of measured genes, and the percentage of transcripts originating from mitochondrial genes. The purpose of QC is to identify abnormal peaks in the matrix and set a threshold to remove them. Such peaks may correspond to dead cells, cells with broken cell membranes, or doublets (defined as the situation where a single droplet contains two or more cells during single-cell sequencing).

Table 2
Datasets size before and after QC.

Dataset	Before	After
PBMC	2900 cells*32738 genes	2843 cells*13003 genes
Klein	2717 cells*24021 genes	2713 cells*24021 genes
Chen	14437 cells*23284 genes	14197 cells*17752 genes
Campbell	21086 cells*26774 genes	11316 cells*21417 genes

QC focuses on filtering cells and genes with low counts, filtering genes with over-expressed mitochondria, filtering doublet cells with over-expressed counts, and ultimately normalizing the expression matrix to reduce the batch effect so that each row in the expression matrix has the same total expression values. In this paper, we use Scanpy [52] for QC: cells with an expression value of less than 200 and genes with an expression value of less than three are filtered out. In the case of PBMC, cells with a mitochondrial gene ratio of more than 5% and a total gene count exceeding 2000 are filtered (as shown in Fig. 3), and finally normalized. Table 2 shows the number of genes and cells contained in the four datasets before and after QC.

3. Results

3.1. Matrix dimensions prediction

Algorithm 3 involves singular value decomposition (SVD), and the singular values are arranged from largest to smallest. In many cases, we can approximate the matrix with the front singular values and the corresponding left and right singular value vectors. For a given matrix D, we do not have to calculate all the singular values each time but instead predict the matrix dimension (sv in Algorithm 3) incrementally. In this paper, Algorithm 3 adopts a fixed incremental value of R for updates. The experimental results show that: 1. A small R also recovers the original data of the matrix, but the large value has a certain improvement on the imputation performance of unknown entries. 2. sv determines the number of singular values that are retained during SVD calculation, but the later singular values are very small, which does not improve the experimental performance much. Therefore, it is worth investigating the choice of an appropriate R-value.

We use the Klein dataset with a 10% masking ratio, the Klein dataset with a 50% masking ratio, the PBMC dataset with a 10% masking ratio and the Chen dataset with a 10% masking ratio for experimental analysis to investigate the influence of the R-value on the four evaluation metrics. The corresponding results are shown in Fig. 4a-d, respectively. We use different R ($R_0 = 10$, $R_1 = 50$, and then incremented by 50, with a maximum value of 400) on these four datasets. The left figure in Fig. 4a-d shows the curves of MSE and MAE with R, and the right fig-

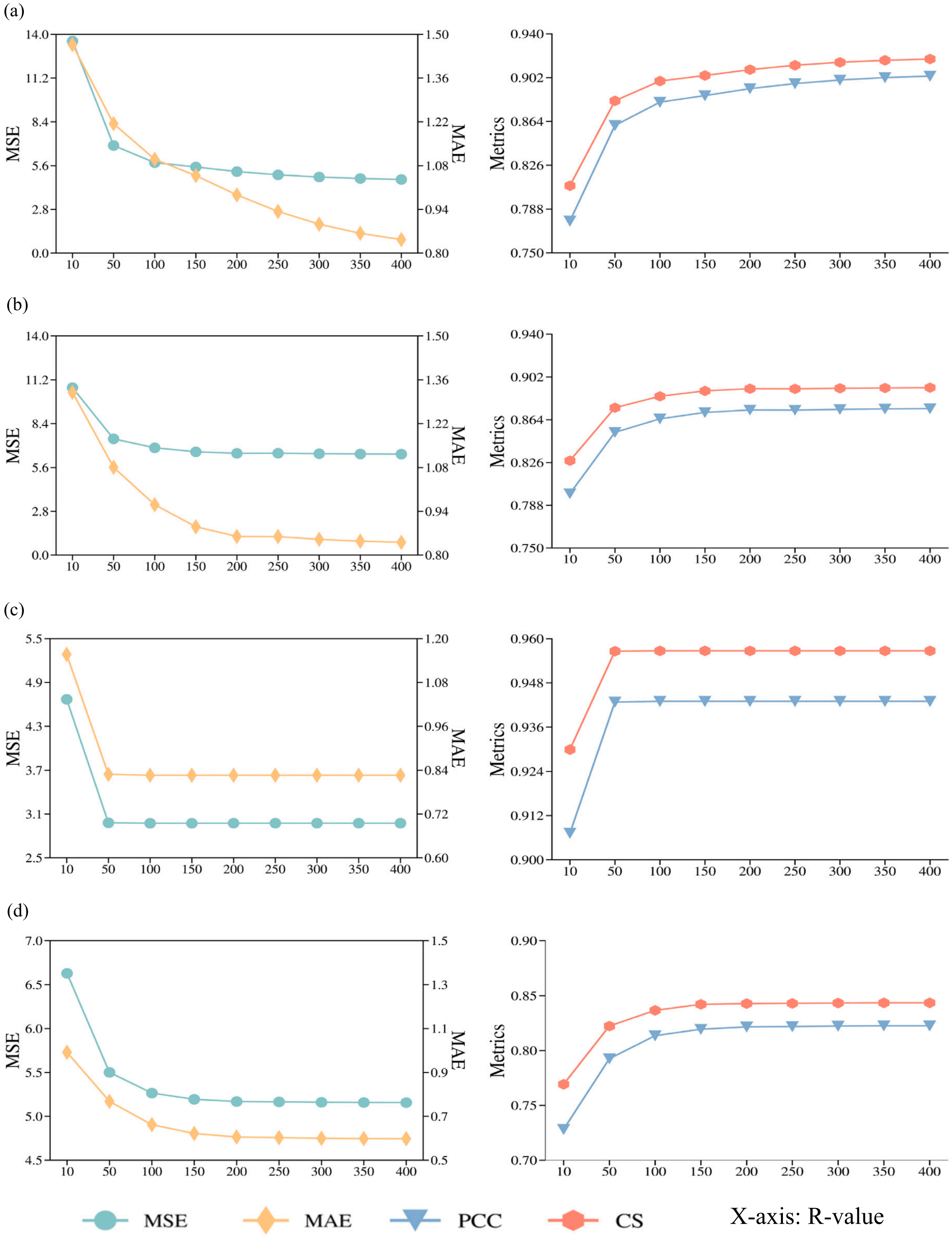

Fig. 4. (a) represents the curves of MSE (left), MAE (left), PCC (right), and CS (right) (dataset = Klein, mask = 10%). (b) shows the curves of MSE (left), MAE (left), PCC (right), and CS (right) (dataset = Klein, mask = 50%). (c) shows the curves of MSE (left), MAE (left), PCC (right), and CS (right) (dataset = PBMC, mask = 10%). (d) represents the curves of MSE (left), MAE (left), PCC (right), and CS (right) (dataset = Chen, mask = 10%).

ure shows the curves of PCC and CS with R. Fig. 4a and b use Klein, and we can observe that after $R = 250$, the trend of the curve slows down. This suggests that after R increases to a certain value, the experimental results are less affected by it. Comparing the two figures, it can be seen that on the same dataset, the changing trend of the four metrics with R is less affected by masking ratios, and they can finally obtain stable results. Fig. 4c shows that on PBMC, the curves tend to be stable after $R = 100$, or even almost a straight line, which is mainly due to the small size of PBMC. For the larger dataset Chen, the experimental results (as shown in Fig. 4d) show a similar trend. R is a parameter directly related to predicting dimension, further determining the number of singular values retained by SVD. For the k-th iteration, a larger R-

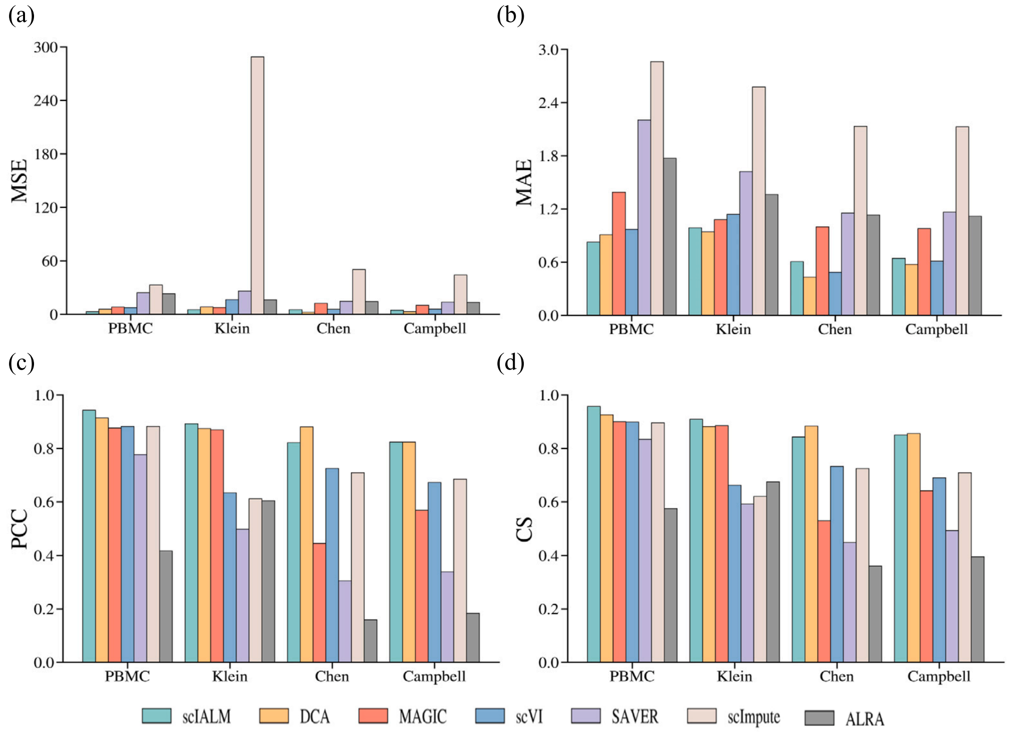

Fig. 5. (a) represents the MSE of gene expression values before and after imputation with different methods. (b) represents the MAE of gene expression values before and after imputation with different methods. (c) shows the PCC of gene expression values before and after imputation with different methods. (d) represents the CS of gene expression values before and after imputation with different methods.

value means that more singular values can be obtained in this iteration, which makes the rank of the matrix converge faster. So the experimental performance is significantly improved when R increases from 50 to 250. The singular value is decreasing, and the sum of the previous values usually accounts for 99% or more of the sum of all singular values. The later singular values are small and have little effect on the results, so the improvement of experimental performance is not obvious after $R = 250$.

Based on the above results, combined with the running time, and datasets with different sizes and sparsity, we all select the parameter $R = 200$ as the increasing value of rank during each iteration for Algorithm 3, but it is an adjustable parameter. For most datasets, we recommend starting from $R = 200$. We find the rank tends to fluctuate during iteration and does not always increase by a fixed value. To prevent large deviations, we update the parameter sv in the following strategy similar to that in [43]: svp denotes the number of singular values greater than $\mu_k{}^{-1}$. If svp_k is smaller than sv_k after iteration, sv_{k+1} is updated to $svp_k + 1$, otherwise it is incremented by R, as shown in equation (13):

$$sv_{k+1} = \begin{cases} svp_k + 1 & if\ \ svp_k < sv_k \\ min(svp_k + R,\ d) & if\ \ svp_k = sv_k \end{cases} \tag{13}$$

where $d = min(m, n)$, $sv_0 = 5$.

3.2. Results of different methods

In this paper, we compare the imputation performance of scIALM with six mainstream methods for experiments on four real datasets: DCA, MAGIC, scVI, scImpute, SAVER and ALRA. The specific experimental parameters employed in this paper are as follows: $\varepsilon = 0.0001$, $\mu = 0.3/d\_norm$, $\rho = 1.1 + 2.5\rho_s$, d_norm is the maximum singular value of the input matrix D, and ρ_s is the non-zero ratio of the data, which is calculated. In this section, the masking ratio of four datasets is 10%. In Fig. 5, a-d represent MSE, MAE, PCC, and CS in turn, and we can see scIALM has improved on MSE and MAE. For PCC, scIALM improved by 3.2% on PBMC, 2.1% on Klein, and performed flat on the Campbell compared to DCA. For CS, compared to DCA, scIALM improves 3.5% and 6.78% in PBMC and Klein, respectively, and Campbell results are close. Overall, the results of MAGIC, scVI, SAVER, and scImpute are not stable, varying widely across different datasets. For example, the MSE of scImpute is 32.9176 on PBMC but as high as 288.8302 on Klein. PCC of SAVER is 0.7768 on PBMC but as low as 0.3389 on Campbell. Whereas scIALM does not show such a large difference on different datasets. MSE and MAE of ALRA improve on large-scale datasets (Chen and Campbell), but PCC and CS decrease significantly.

In addition to PBMC, the other three datasets provided cluster labels, with Klein containing 4 clusters, Chen containing 47 clusters, and Campbell containing 21 clusters. We use ARI and NMI to assess the clustering effect of different methods. In Fig. 6, a and b represent ARI and NMI, respectively. Among the above four indicators, DCA is sometimes better than scIALM, and in terms of clustering effect, scIALM performs better than DCA on the three datasets. For datasets with many cluster labels, the clustering effectiveness of each method needs to be improved. The imputation principle of ALRA is also based on the low-rank of the matrix, and its clustering performance is slightly better than scIALM. Both of them use SVD to compute a low-rank approximation of a matrix, and the matrix resulting from this low-rank approximation contains very few zeros. The final step of the ALRA is to recover the biological zeros in the matrix by thresholding its entries, which leads to an increase in its performance. But scIALM does not have this step. Klein analyzed mouse embryonic stem cells, revealing in detail the population structure and the heterogeneous onset of differentiation after leukemia inhibitory factor (LIF) withdrawal. The cluster labels are determined by the intervals of LIF withdrawal (d0, d2, d4, d7 days). The uniform manifold approximation and projection (UMAP) algorithm used to reduce the dimension of the expression matrix. And it can realize the visual analysis of clustering. In Fig. 6c, we show the figures of the raw matrix, noised matrix and the imputed matrix after scIALM. The expression matrix imputed by scIALM can make the same clusters tighter and different clusters more spread.

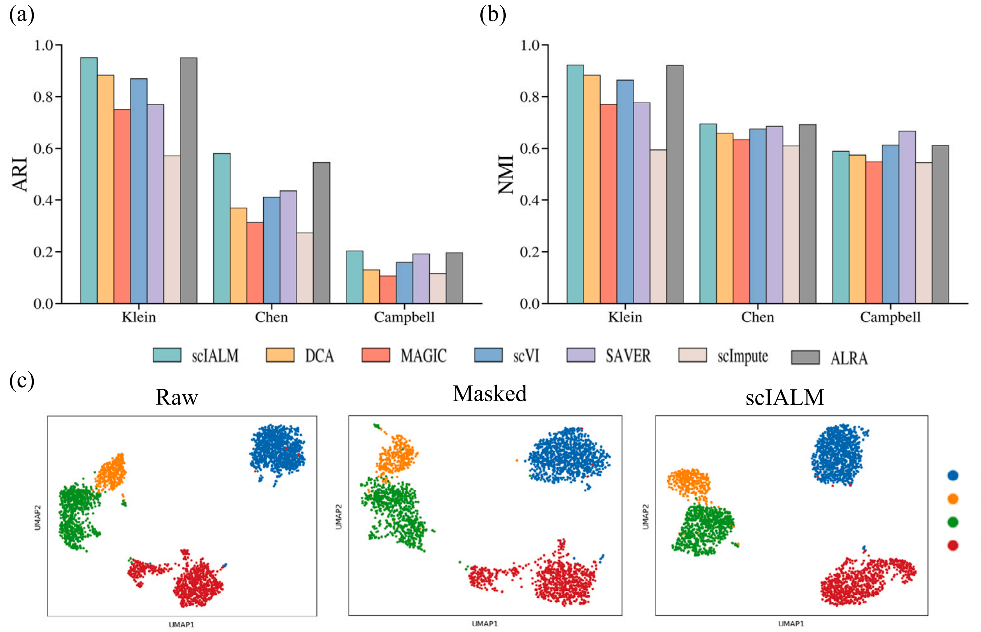

Fig. 6. (a) shows the ARI of different methods. (b) shows the NMI of different methods. (c) shows the visualizations of the raw matrix, masked matrix and imputed matrix after scIALM.

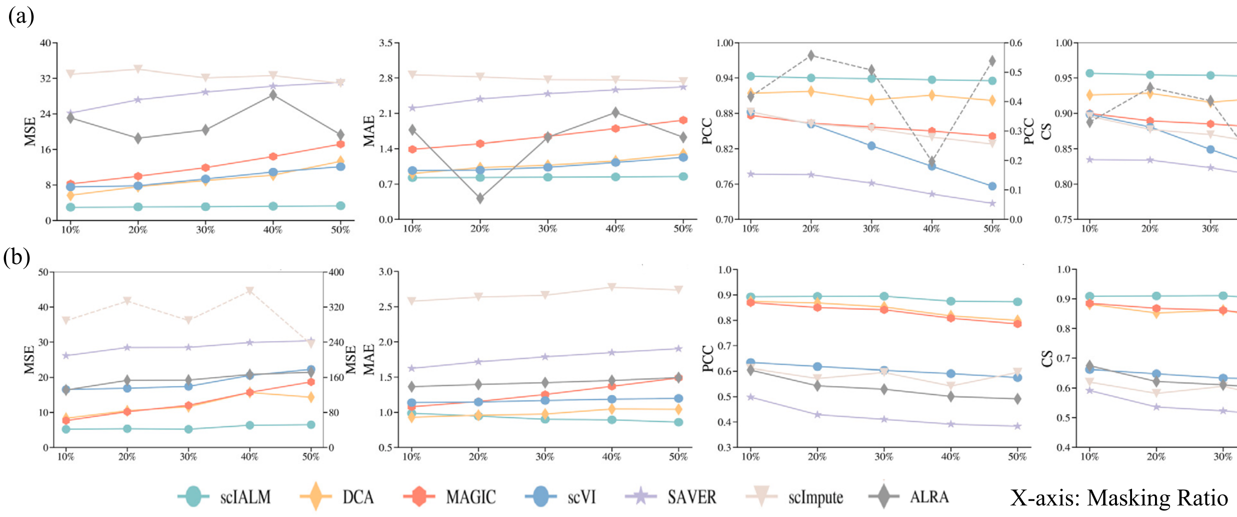

Fig. 7. (a) shows the MSE, MAE, PCC and CS of seven methods at different masking ratios from left to right. (dataset = PBMC) (b) shows the MSE, MAE, PCC and CS of seven methods at different masking ratios from left to right. (dataset = Klein).

3.3. Results under different masking ratios

To compare the results of scIALM and the other six methods under different masking ratios, we randomly mask the data with 10%, 20%, 30%, 40%, and 50% on PBMC and Klein to simulate different dropout rates. Draw a line chart based on the experimental results to analyze the effects of distinct methods, as shown in Fig. 7.

Fig. 7a shows the MSE, MAE, PCC, and CS of different methods on PBMC, and Fig. 7b shows the results on Klein. The values of the dashed lines in Fig. 7 are on the right axis of the corresponding subplots. We can see that scIALM is not sensitive to the masking rates, and the four metrics are more stable than the other six methods. Fig. 7a shows that DCA and scVI perform closely, followed by MAGIC and SAVER. As the masking ratio increases, the MSE and MAE of SAVER rise the fastest. Although MSE and MAE of scImpute show a downward trend, their values are always higher than other methods. The PCC and CS of scVI and SAVER decrease rapidly with the increase of masking rate, indicating that they are sensitive to masking rate. The indicators of ALRA are fluctuating, and it is difficult to assess the impact of masking ratios on it. The overall performance of other methods is: as the masking rate increased, the performance decreased. In Fig. 7b, we can see that scIALM is superior to other methods in two ways: 1. scIALM is almost the best in four metrics; 2. the performance degradation of scIALM is minimized as the masking ratio increases. The amount of data affects the model effect to some extent, so when the masking rate rises, the DCA and scVI performance decreases. The other methods used the expression of individual cells and individual genes for imputation. But scIALM does not rely on a single row or column information, the specific masking position has less impact on it, making the method robust.

4. Discussion

Single-cell RNA sequencing can provide gene expression at the single-cell level, allowing the study of cellular heterogeneity. However, the expression matrix contains many zero values, which complicates scRNA-seq data analysis. Therefore, how to identify and impute false zero values becomes one of the research in this field. The emergence of compressed sensing breaks through the traditional Nyquist sampling theorem and the original signal can be recovered by sparse signal. scRNA-seq possesses the natural sparsity and carries out the low-rank assumption. Therefore, this paper proposes scIALM to impute the matrix, and the dropout events are effectively recovered.

scIALM uses an inexact augmented Lagrange multiplier method to impute the gene expression matrix, assuming that the matrix is low-rank. The original data is recovered accurately with the lowest possible rank while imputing unknown entries. Compared with MAGIC, scVI, SAVER, scImpute and ALRA, scIALM has a significant improvement in four metrics. From the results, on Chen, the PCC and CS of scIALM are slightly worse than DCA, which may be due to the fact that Chen contains $14197 genes * 17752 cells$, and the whole matrix only contains 8.34% of non-zero values, which has relatively little global information. We admit that the dataset and the amount of data are not the only reasons that affect the results, but it is also one of the main reasons, and the experimental results are also affected by the characteristics of dataset itself. Another possible reason is that DCA uses ZINB as the loss function, and there has been extensive work demonstrating that negative binomial distributions can characterize scRNA-seq data well. Compared to other datasets, Chen can be better characterized and therefore performs better on DCA than scIALM. We will further explore and verify this conjecture in the future.

It is worth noting that scIALM does not need to learn any parameters during iteration, and can achieve better experimental results by calculation. In contrast to other methods, this method focuses on preserving the global information of the matrix, which is achieved by calculating the singular values. From Fig. 7, we can see that when we mask half of the real data, scIALM performs better and has the lowest performance degradation rate compared to $masking\ ratio = 10\%$, which suggests that it can better cope with less amount of data and large noise.

For the choice of R-value, according to Fig. 4a, it can be seen that on Klein, there is still a decreasing trend in MSE and MAE and an increasing trend in PCC and CS after $R = 200$. The reason for this is that Klein contains $2713 genes * 24021 cells$, and 34.39% non-zero values, which is a larger amount of real data than other datasets. Thus, for datasets with fewer cells and genes and low sparsity, we can start iterating with a larger R-value to achieve the best results as quickly as possible.

The problem with scIALM at this stage is that the rank of the matrix is updated with a value of 1 or a fixed value of R, which lacks flexibility. Future research focuses on finding more flexible strategies for updating the rank, perhaps combining singular values or expression matrices, with the aim of better adapting to the characteristics of different datasets. In addition, inspired by ALRA, further research will focus on how to recover the true biological zeros in the expression matrix to improve the downstream analysis. In the future, it may be possible to combine with deep neural networks to better learn the similar expression information between genes to further improve the experimental results.

5. Conclusion

scRNA-seq provides gene expression profiles at the single-cell level, which makes the gene sequencing analysis from macroscopic to microscopic, and provides a basis for studying cellular heterogeneity.

Considering one of the main characteristics of scRAN-seq — sparsity, this paper uses an inexact augmented Lagrange multiplier (IALM) method to fill the sparse single-cell RNA sequencing expression matrix and recover unknown entries based on sparse but clean (accurate) data. The original data in the expression matrix were recovered with an error of 0.0001, and the masking data were evaluated using four indicators: MSE, MAE, PCC, and CS. Through experiments, we found scIALM improved on different datasets and was insensitive to sparsity changes, achieving better results than other methods even if it contains 50% noise. For downstream analysis, this paper uses ARI and NMI for evaluation, and there are improvements on scIALM compared to other methods. Furthermore, we investigated the impact of the sv parameter on the experimental results and provided a strategy for updating sv-value according to the results, speeding up the convergence of the algorithm.

In this paper, we applied the augmented Lagrange multiplier method to single-cell data for the first time, which effectively recovered the dropout noise and was better able to cope with large noise. Since scRNA-seq data are numerical, we can apply mathematical methods to the field of bioinformatics, focusing on interpreting bioinformatics problems from a mathematical point of view, which provides new research ideas in the field. We will explore more flexible strategies to update sv to better fit different datasets and investigate methods to recover real biological zeros in the matrix to further improve downstream analysis results.

Funding

This work was supported by Ministry of Science and Technology of the People's Republic of China (2022FY101104).

CRediT authorship contribution statement

Xiaohong Liu: Methodology, Investigation, Data curation, Writing – original draft. Han Wang: Writing – review & editing, Visualization. Jingyang Gao: Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Kolodziejczyk AA, Kim JK, Svensson V, Marioni JC, Teichmann SA. The technology and biology of single-cell rna sequencing. Mol Cell 2015;58(4):610–20.

[2] Slatko BE, Gardner AF, Ausubel FM. Overview of next-generation sequencing technologies. Curr Protoc Mol Biol 2018;122(1):e59.

[3] Yilmaz S, Singh AK. Single cell genome sequencing. Curr Opin Biotechnol 2012;23(3):437–43.

[4] Tang F, Lao K, Surani MA. Development and applications of single-cell transcriptome analysis. Nat Methods 2011;8(Suppl 4):S6–11.

[5] Wen L, Tang F. Single cell epigenome sequencing technologies. Mol Asp Med 2018;59:62–9.

[6] Vistain LF, Tay S. Single-cell proteomics. Trends Biochem Sci 2021;46(8):661–72.

[7] Lee J, Hyeon DY, Hwang D. Single-cell multiomics: technologies and data analysis methods. Exp Mol Med 2020;52(9):1428–42.

[8] Zheng M, Hu Z, Mei X, Ouyang L, Song Y, et al. Single-cell sequencing shows cellular heterogeneity of cutaneous lesions in lupus erythematosus. Nat Commun 2022;13(1):7489.

[9] Zhang C, Han X, Liu J, Chen L, Lei Y, et al. Single-cell transcriptomic analysis reveals the cellular heterogeneity of mesenchymal stem cells. Genomics Proteomics Bioinform 2022;20(1):70–86.

[10] Fortunato S, Barthelemy M. Resolution limit in community detection. Proc Natl Acad Sci USA 2007;104:36–41.

[11] Gohil SH, Iorgulescu JB, Braun DA, Keskin DB, Livak KJ. Applying high-dimensional single-cell technologies to the analysis of cancer immunotherapy. Nat Rev Clin Oncol 2021;18(4):244–56.

[12] Zhu Y, Huang Y, Tan Y, Zhao W, Tian Q. Single-cell rna sequencing in hematological diseases. Proteomics 2020;20(13):1900228.

[13] Li G, Yang Y, Van Buren E, Li Y. Dropout imputation and batch effect correction for single-cell rna sequencing data. J Bio-X Res 2019;2(04):169–77.

[14] Hicks SC, Townes FW, Teng M, Irizarry RA. Missing data and technical variability in single-cell rna-sequencing experiments. Biostatistics 2018;19(4):562–78.

[15] Qi R, Ma A, Ma Q, Zou Q. Clustering and classification methods for single-cell rna-sequencing data. Brief Bioinform 2020;21(4):1196–208.

[16] Luecken MD, Theis FJ. Current best practices in single-cell rna-seq analysis: a tutorial. Mol Syst Biol 2019;15(6):e8746.

[17] Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol 2019;37:547–54.

[18] Das S, Rai A, Rai SN. Differential expression analysis of single-cell rna-seq data: current statistical approaches and outstanding challenges. Entropy 2022;24(7):995.

[19] Mignone P, Pio G, D'Elia D, Ceci M. Exploiting transfer learning for the reconstruction of the human gene regulatory network. Bioinformatics 2020;36(5):1553–61.

[20] Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell rna-seq denoising using a deep count autoencoder. Nat Commun 2019;10(1):390.

[21] Van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, et al. Recovering gene interactions from single-cell data using data diffusion. Cell 2018;174(3):716–29.

[22] Ding J, Condon A, Shah SP. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nat Commun 2018;9(1):2002.

[23] Li WV, Li JJ. An accurate and robust imputation method scimpute for single-cell rna-seq data. Nat Commun 2018;9(1):997.

[24] Huang M, Wang J, Torre E, Dueck H, Shaffer S, et al. Saver: gene expression recovery for single-cell rna sequencing. Nat Methods 2018;15(7):539–42.

[25] Linderman G, Zhao J, Roulis M, Bielecki P, Flavell R, Nadler B, et al. Zero-preserving imputation of single-cell RNA-seq data. Nat Commun 2022;13:192.

[26] Karst S, Ziels R, Kirkegaard R, Sørensen E, McDonald D, et al. High-accuracy long-read amplicon sequences using unique molecular identifiers with Nanopore or PacBio sequencing. Nat Methods 2021;18:165–9.

[27] Candes E, Recht B. Exact matrix completion via convex optimization. Commun ACM 2012;55(6):111–9.

[28] Candès EJ, Wakin MB. An introduction to compressive sampling. IEEE Signal Process Mag 2008;25(2):21–30.

[29] Dong J, Ding Y, Kudo H. Nonlinear filtered compressed sensing applied on image denoising. In: 2021 6th international conference on multimedia and image processing; 2021. p. 1–6.

[30] Fan L, Zhang F, Fan H, Zhang C. Brief review of image denoising techniques. Vis Comput Ind Biomed Art 2019;2:1–12.

[31] Shangguan P, Jiang W, Wang J, Wu J, Cai C, et al. Multi-slice compressed sensing mri reconstruction based on deep fusion connection network. Magn Reson Imaging 2022;93:115–27.

[32] Ye JC. Compressed sensing mri: a review from signal processing perspective. BMC Biomed Eng 2019;1(1):1–17.

[33] Li Z, Xu W, Zhang X, Lin J. A survey on one-bit compressed sensing: theory and applications. Front Comput Sci 2018;12:217–30.

[34] Chen J, Yang S, Wang Z, Mao H. Efficient sparse representation for learning with high-dimensional data. IEEE Trans Neural Netw Learn Syst 2021.

[35] Han S, Wang N, Guo Y, Tang F, Xu L, et al. Application of sparse representation in bioinformatics. Front Genet 2021;12:810875.

[36] Lin Z, Chen M, Ma Y. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint. Available from: arXiv:1009.5055, 2010.

[37] Xie D, Woerdeman HJ, Xu A-B. Parametrized quasi-soft thresholding operator for compressed sensing and matrix completion. Comput Appl Math 2020;39:1–24.

[38] Jaradat Y, Masoud M, Jannoud I, Manasrah A, Alia M. A tutorial on singular value decomposition with applications on image compression and dimensionality reduction. In: 2021 international conference on information technology (ICIT). IEEE; 2021. p. 769–72.

[39] Slovin S, Carissimo A, Panariello F, Grimaldi A, Bouché V, et al. Single-cell rna sequencing analysis: a step-by-step overview. RNA Bioinform 2021:343–65.

[40] Ma J, Yuan Y. Dimension reduction of image deep feature using pca. J Vis Commun Image Represent 2019;63:102578.

[41] Chen Y, Fan J, Ma C, Yan Y. Bridging convex and nonconvex optimization in robust pca: noise, outliers, and missing data. Ann Stat 2021;49(5):2948.

[42] Tanabe H, Fukuda EH, Yamashita N. An accelerated proximal gradient method for multiobjective optimization. Comput Optim Appl 2023:1–35.

[43] Toh K-C, Yun S. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pac J Optim 2010;6(615–640):15.

[44] Bertsekas D. Constrained optimization and Lagrange multiplier methods. Academic Press; 2014.

[45] Zhang Y, Xia Y, Zhang H, Wang G, Dai L. On the generic low-rank matrix completion. arXiv preprint. Available from: arXiv:2102.11490, 2021.

[46] Borwein J, Lewis A. Convex analysis. Springer; 2006.

[47] Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 2015;161(5):1187–201.

[48] Chen R, Wu X, Jiang L, Zhang Y. Single-cell rna-seq reveals hypothalamic cell diversity. Cell Rep 2017;18(13):3227–41.

[49] Campbell JN, Macosko EZ, Fenselau H, Pers TH, Lyubetskaya A, et al. A molecular census of arcuate hypothalamus and median eminence cell types. Nat Neurosci 2017;20(3):484–96.

[50] Balzer MS, Ma Z, Zhou J, Abedini A, Susztak K. How to get started with single cell rna sequencing data analysis. J Am Soc Nephrol 2021;32(6):1279.

[51] Amezquita RA, Lun AT, Becht E, Carey VJ, Carpp LN, et al. Orchestrating single-cell analysis with bioconductor. Nat Methods 2020;17(2):137–45.

[52] Wolf FA, Angerer P, Theis FJ. Scanpy: large-scale single-cell gene expression data analysis. Genome Biol 2018;19:1–5.