

RESEARCH

Open Access



scZiva: imputation method for single-cell RNA-seq data with zero-inflated variational autoencoder

Long Tuan Vo¹, Van Vinh Le^{1*}, Quoc Toan Ha¹ and Anh Quoc Nguyen¹

*Correspondence:

Van Vinh Le

vinhvl@hcmute.edu.vn

¹Ho Chi Minh City University of
Technology and Engineering, Vo

Van Ngan Street,

Ho Chi Minh City 700000, Vietnam

Abstract

Background Single-cell RNA sequencing (scRNA-seq) is considered a revolution in gene expression studies and offers significant benefits across various fields of biomedical or clinical applications. However, a high proportion of technical dropouts in scRNA-seq data leads to increased noise and reduces the performance of downstream analyses such as cell clustering, differential expression analysis, and cell trajectory inference. Numerous recent imputation methods utilize deep learning to recover missing gene expression values in scRNA-seq data. Despite the research efforts, existing methods have limitations in capturing local co-expression patterns in scRNA-seq data and handling the uncertainty in distinguishing technical zeros from true biological zeros.

Results This work proposes a novel imputation method for scRNA-seq data, called scZiva, based on Variational Autoencoder (VAE). It introduces a structured probabilistic framework that jointly models dropout uncertainty and statistically induced local gene dependencies. scZiva also adopts a probability-guided selective imputation mechanism to recover likely technical dropouts while preserving biologically meaningful zeros. The framework is implemented using a Zero-Inflated Negative Binomial (ZINB) likelihood with a convolution-enhanced encoder architecture. Comprehensive experiments conducted on both simulated and real datasets demonstrate the strength of scZiva compared with other baseline methods.

Conclusion The proposed method demonstrates strong and stable performance in most evaluation settings compared to five baseline methods, particularly in recovering missing gene expression values and supporting downstream analyses. It is a promising approach for analyzing scRNA-seq data.

Keywords scRNA-seq, Imputation, Missing data, Variational autoencoder

Introduction

Gene expression studies have been significantly advanced by single-cell RNA sequencing (scRNA-seq) technologies [1], and have a meaningful impact on biomedical and clinical applications [2]. While traditional bulk RNA-seq technology only provides analysis of gene expression at the population level, scRNA-seq allows for investigating expression



profiles at a single-cell resolution. Despite its advantages, scRNA-seq data suffer from some limitations due to their unique characteristics. One of the significant obstacles is a phenomenon called dropout events, which is the state of missing values in data due to technical noises, *i.e.*, amplification bias, low-input cell library, or inefficient mRNA capture. The dropout events can result in poor performance of downstream analysis for scRNA-seq data [3], such as cell clustering, differential expression analysis, and cell trajectory inference. Thus, it poses research challenges for developing effective scRNA-seq imputation methods to recover missing gene expression values.

Most early imputation approaches for scRNA-seq data can be classified into model-based, smoothing-based, and low-ranked matrix-based methods. SAVER [4] and scImpute [5] are typical algorithms of the model-based group, which assume the count of genes in cells follows statistical models. They estimate the parameters of the models using observed data and then infer missing values. While SAVER uses the Poisson-Gamma model and calculates the posterior mean to recover expression values, scImpute is based on a mixture model of Gamma and Normal distributions to represent dropout values and the actual gene expression levels, respectively. Smoothing-based imputation methods, such as MAGIC [6] and AcImpute [7], utilize the similarity between cells when they estimate missing values via neighboring cells. MAGIC firstly constructs an exponential Markov matrix and then achieves an imputed matrix by multiplying the matrix by the original gene expression data. AcImpute constrains smoothing weights across cells for genes that have different expression levels to enhance imputation performance.

Additionally, some low-ranked matrix-based methods, such as CMF-Impute [8], scRMD [9], and netNMF-sc [10], are based on matrix decomposition techniques to map gene expression data into a low-rank representation to support reconstructing missing values in the original matrix. CMF-Impute applies a collaborative matrix decomposition method for two matrices of cell features and gene features in its data recovery task, while scRMD uses a robust matrix decomposition to impute data. Different from the two methods, netNMF-sc additionally uses available gene–gene interactions information to guide its parameter estimation process of the imputation model.

Recent imputation methods for scRNA-seq data primarily utilize the advantages of deep learning techniques. A group of methods, such as sc-fGAIN [11], scMultiGAN [12], Scmaskgan [13], and scGGAN [14], rely on generative adversarial networks (GANs) in their imputation process. In those approaches, two networks of Generator and Discriminator are trained together to help the Generator return a high-quality imputed matrix. sc-fGAIN applies f -GAN framework, which utilizes f -divergence functions to optimize the imputation model, while both scMultiGAN and Scmaskgan transform gene expression matrices into images before performing GAN-based imputation. scGGAN is a graph-based generative adversarial network and incorporates bulk RNA-seq and gene sequencing data for training its model.

Another category of deep learning-based methods for scRNA-seq data imputation, *e.g.*, IGSimpute [15], Bubble [16], DCA [17], and SAE-Impute [18] aim to impute data using autoencoder structures where the decoder is responsible for reconstructing completed data. IGSimpute and Bubble adopt denoising autoencoders, whereas the Bubble model is further constrained by bulk RNA-seq data. While DCA proposes a variant of an autoencoder to denoise scRNA-seq data, supporting the recovery of missing values, SAE-Impute combines autoencoders and subspace regression in its imputation process.

Besides, the study of Widad [19] applies a standard Variational Autoencoder to impute scRNA-seq data.

Despite their success, existing deep learning methods still have main limitations. First, most approaches are difficult to explicitly handle the uncertainty in distinguishing technical zeros from true biological zeros. Second, gene expression values are often treated as independent input features, ignoring local co-expression patterns. These limitations are especially evident in current VAE-based frameworks, where reconstruction is typically performed globally without selective treatment of likely dropout events. Furthermore, there is a fundamental trade-off between recovering missing expression signals and preserving true biological silence.

Motivated by these challenges, we propose scZiva, a structured probabilistic imputation framework built on a VAE for scRNA-seq data. The novelty of scZiva lies in a joint probabilistic-structural design with three major characteristics: (i) a ZINB-based likelihood to model overdispersion and zero inflation; (ii) convolutional encoding combined with gene reordering to exploit local co-expression structures; and (iii) a probability-guided selective imputation mechanism that adaptively reconstructs likely technical dropouts while preserving biologically meaningful zeros. Experimental results on both simulated and real datasets show the strength of scZiva compared to baseline methods in various tasks, including the recovery of dropout values, cell clustering, differential expression analysis, and cell trajectory inference.

Methods

Given a raw scRNA-seq count matrix $X \in \mathbb{R}^{N \times G}$, where N and G denote the number of cells and genes, respectively. The objective of scZiva is to recover the expression levels of genes that were not observed due to dropout events. A preprocessing task is applied to discard genes with zero expression across all cells of X . The main architecture of scZiva is illustrated in Fig. 1, consisting of an encoder and a decoder. The encoder employs a one-dimensional convolutional layer (Conv1D) to capture those local relationships while the decoder reconstructs the gene expression distribution using the Zero-Inflated Negative Binomial (ZINB [20]) parameterization.

Encoder: Conv1D for discovering gene relationships

Conv1D layer applies sliding filters along one gene vector dimension to extract local patterns and is widely adopted as a fundamental component in hybrid deep learning

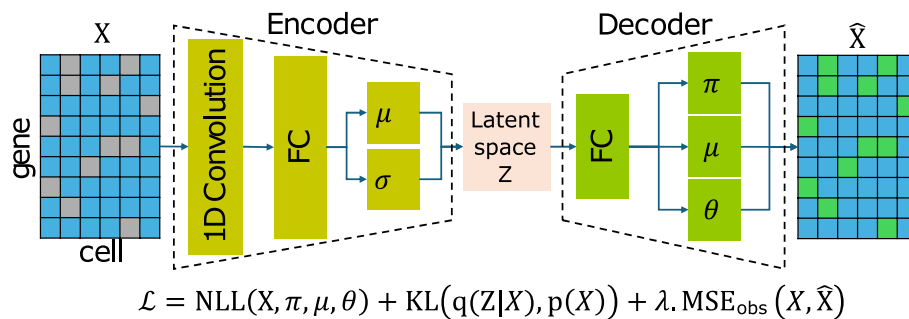


Fig. 1 Overall architecture of scZiva. The encoder maps the raw input X to latent parameters (μ_Z, σ_Z^2) , and the decoder produces ZINB parameters (π, μ, θ) to reconstruct $\hat{X}$. Training minimizes a loss combining ZINB negative log-likelihood, KL divergence, and an observed-entry mean squared error term

architectures [21–23]. Motivated by these successes, we employ Conv1D to explore local gene-gene dependencies that potentially reflect biological patterns. Moreover, to better exploit these local correlations, we first reorder genes using covariance-based hierarchical clustering, ensuring that highly co-expressed genes are positioned adjacently in the input matrix. This preprocessing step introduces structured proximity among statistically related genes, allowing Conv1D to model local co-expression patterns. Noticed that such locality is induced by statistical structure instead of assumed to reflect inherent biological contiguity. Technically, a Conv1D layer with kernel size k , stride s , and no padding, parameterized by weights $w \in \mathbb{R}^k$ and bias $b \in \mathbb{R}$ shared across all cells, transforms the input as:

$$H_{i,j} = \sum_{m=0}^{k-1} w_m X_{i,j s+m} + b, \quad (1)$$

where $H \in \mathbb{R}^{N \times L}$ is the resulting feature map, i indexes the cell ($1 \leq i \leq N$), j indexes the convolutional window, and $L = \lfloor \frac{G-k}{s} \rfloor + 1$ is the output length along the gene dimension.

Let $x_i \in \mathbb{R}^G$ denote the zero-filled expression profile of cell i . The Conv1D output $h_i \in \mathbb{R}^L$ is passed through a fully connected layer to produce a hidden representation, which is then linearly projected into the mean vector $\mu_i \in \mathbb{R}^{d_z}$ and log-variance vector $\log \sigma_i^2 \in \mathbb{R}^{d_z}$ of the approximate posterior:

$$q(Z_i | X_i) = \mathcal{N}(\mu_i, \text{diag}(\exp(\log \sigma_i^2))). \quad (2)$$

Sampling from this posterior is performed via the reparameterization trick:

$$Z_i = \mu_i + \sigma_i \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (3)$$

Decoder: ZINB parameterization for gene expression reconstruction

Given a latent representation $Z_i \in \mathbb{R}^{d_z}$ from the encoder, the decoder maps it back to the gene expression space by predicting the parameters of a ZINB distribution for each gene [20]. Specifically, the decoder projects Z_i through a fully connected layer with a rectified linear activation, producing a hidden representation. This is followed by another linear transformation whose output dimension is $3G$, yielding a vector $o_i \in \mathbb{R}^{3G}$. We then partition o_i into three G -dimensional components corresponding to dropout probability, mean, and dispersion:

$$o_i = [p_i^{(\pi)} \parallel p_i^{(\mu)} \parallel p_i^{(\theta)}],$$

where $\parallel$ denotes concatenation. To ensure valid parameter domains, element-wise nonlinearities are applied:

$$\pi_i = \sigma(p_i^{(\pi)}), (\mu_i, \theta_i) = \text{softplus}(p_i^{(\mu)}, p_i^{(\theta)}), \quad (4)$$

with $\sigma(\cdot)$ the logistic sigmoid mapping to $(0, 1)$ and $\text{softplus}(\cdot) = \log(1 + e^{(\cdot)})$ ensuring strictly positive outputs.

Thus, for each cell i and gene j , the decoder outputs a triplet $(\pi_{ij}, \mu_{ij}, \theta_{ij})$ parameterizing the ZINB distribution:

$$X_{ij} \sim \text{ZINB}(\pi_{ij}, \mu_{ij}, \theta_{ij}), \quad (5)$$

where π_{ij} models the dropout probability, μ_{ij} represents the mean expression, and θ_{ij} controls the inverse dispersion. Details about formula (5) are provided in the Supplementary Section 1. During training, these three parameters are optimized jointly with the encoder via a composite loss that includes the ZINB negative log-likelihood, Kullback–Leibler divergence in the latent space, and a small mean squared error term for reconstruction.

Training phase

The training objective of scZiva extends the standard VAE framework by combining probabilistic modeling of count data with an auxiliary reconstruction criterion. Formally, the loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{ZINB}} + \mathcal{L}_{\text{KL}} + \lambda \cdot \mathcal{L}_{\text{MSE}}, \quad (6)$$

where each term is described below.

ZINB negative log-likelihood The primary objective here is to minimize the negative log-likelihood of the observed counts under the ZINB distribution parameterized by $(\pi_{ij}, \mu_{ij}, \theta_{ij})$. Formally, for each entry x_{ij} , the log-likelihood is defined as:

$$\ell(X_{ij}) = \begin{cases} \log(\pi_{ij} + (1 - \pi_{ij}) \cdot \text{NB}(0; \mu_{ij}, \theta_{ij})), & X_{ij} = 0, \\ \log(1 - \pi_{ij}) + \log \text{NB}(X_{ij}; \mu_{ij}, \theta_{ij}), & X_{ij} > 0, \end{cases} \quad (7)$$

where $\text{NB}(X; \mu, \theta)$ denotes the probability mass function of the Negative Binomial distribution. The ZINB loss is then the negative sum of log-likelihoods across all cells and genes:

$$\mathcal{L}_{\text{ZINB}} = - \sum_{i=1}^N \sum_{j=1}^G \ell(X_{ij}). \quad (8)$$

This term ensures the model can learn to distinguish biological zeros from technical dropouts (via π_{ij}), as well as the overdispersed distribution of expression counts (via μ_{ij} and θ_{ij}).

Kullback–Leibler divergence $\mathcal{L}_{\text{KL}}$ is the Kullback–Leibler divergence between the approximate posterior $q(Z_i | X_i)$ and the standard Gaussian prior $\mathcal{N}(0, I)$. This term makes the latent space smoother and prevents overfitting:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_{k=1}^{d_z} (\mu_{ik}^2 + \sigma_{ik}^2 - \log \sigma_{ik}^2 - 1). \quad (9)$$

Auxiliary MSE loss In addition to the probabilistic objective, we introduce a mean squared error (MSE) term that directly compares the expected value of the ZINB reconstruction with the observed counts on non-dropout event entries:

$$\mathcal{L}_{\text{MSE}} = \sum_{(i,j) \in \Omega} [(1 - \pi_{ij}) \cdot \mu_{ij} - x_{ij}]^2, \quad (10)$$

where Ω denotes the set of observed (non-dropout) indices. This auxiliary loss provides an explicit reconstruction signal that complements the statistical likelihood, encouraging the latent representation to retain richer semantics beyond what is strictly required by maximum likelihood. Importantly, this auxiliary term does not replace the ZINB likelihood as the primary training objective, but serves as a complementary regularizer applied only to observed (non-dropout) entries. Since the MSE is restricted to Ω and weighted adaptively, it does not enforce global smoothing across all zero entries. Such auxiliary objectives have been shown to improve representation quality and generalization [24, 25], similar to the role of dual self-supervised objectives in approaches such as MVEB [26].

Adaptive weighting Instead of fixing the coefficient λ manually, scZiva introduces a learnable weight λ that adaptively balances the auxiliary MSE loss during training. This weight is jointly optimized with the model parameters and regularized by a small quadratic penalty $\lambda_{\text{reg}}(\lambda - 1)^2$ to prevent divergence from its nominal value.

Imputation strategy

After training, scZiva leverages the learned parameters of the ZINB decoder to recover missing or dropout values in the expression matrix. Let π and μ denote the estimated dropout probability and mean expression matrices, respectively. Here, π_{ij} represents the posterior probability under the model that entry (i, j) arises from technical dropout. For each cell i and gene j , a zero entry in the input matrix is considered a potential dropout. Imputation is then performed selectively according to a threshold τ on the dropout probability:

$$\hat{X}_{ij} = \begin{cases} (1 - \pi_{ij}) \cdot \mu_{ij}, & \text{if } X_{ij} = 0 \text{ and } \pi_{ij} > \tau, \\ X_{ij}, & \text{otherwise.} \end{cases} \quad (11)$$

The threshold τ acts as a decision boundary that controls how many zero entries are imputed; in other words, it preserves statistically meaningful zero expressions. By setting τ to a very small value, scZiva restricts the zero-imputations that are assigned by the ZINB decoder. Notably, this mechanism does not assert biological truth for any specific zero entry; instead, it provides a probabilistically guided selection strategy under the model assumptions. Additionally, the scZiva algorithm is presented in detail in Supplementary Algorithm 1.

Evaluation metrics

This work utilizes three commonly used matrices of Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Pearson Correlation Coefficient (PCC) for evaluating the imputation quality of scZiva and baseline methods. While RMSE measures the average magnitude of the squared differences between predicted values and true counts, MAE calculates the average absolute difference between the predicted and true ones. Otherwise, PCC assesses the linear correlation between predicted and true values. More specifically, given the gene expression matrix $X \in \mathbb{R}^{N \times G}$, and an imputed matrix $\hat{X} \in \mathbb{R}^{N \times G}$ with N cells and G genes, the three metrics can be calculated as the following formulations [27, 28].

$$\text{RMSE} = \sqrt{\frac{1}{N \cdot G} \sum_{i=1}^N \sum_{j=1}^G (X_{ij} - \hat{X}_{ij})^2}, \quad (12)$$

$$\text{MAE} = \frac{1}{N \cdot G} \sum_{i=1}^N \sum_{j=1}^G |X_{ij} - \hat{X}_{ij}|, \quad (13)$$

$$\text{PCC} = \frac{\sum_{i=1}^N \sum_{j=1}^G (X_{ij} - \bar{X})(\hat{X}_{ij} - \bar{\hat{X}})}{\sqrt{\sum_{i=1}^N \sum_{j=1}^G (X_{ij} - \bar{X})^2} \sqrt{\sum_{i=1}^N \sum_{j=1}^G (\hat{X}_{ij} - \bar{\hat{X}})^2}}. \quad (14)$$

In downstream analyses, this work uses Adjusted Rand Index (ARI) [29] and Normalized Mutual Information (NMI) [30] to evaluate the performance of clustering methods on raw and imputed gene expression data. In addition, Kendall's rank correlation score [31] is used to assess the cell trajectory inference, while the overlap proportion of genes is utilized to evaluate the results of differential expression analysis.

Benchmark settings

Baseline methods and hyperparameter settings

The proposed method is compared to five baseline methods, including AcImpute, SAVER, scRMD, CL-Impute, and SAE-Impute. These methods were selected because they are widely recognized state-of-the-art imputation tools across different methodological categories. In detail, AcImpute is a smoothing-based algorithm, while SAVER is a model-based technique, and scRMD is a low-rank matrix-based method. The remaining approaches, CL-Impute and SAE-Impute, are deep learning-based algorithms.

In the scZiva model, Conv1D is adopted with the kernel size set to $k = 64$, the hidden and latent dimensions are fixed to 128 and 64, respectively. The model is trained for 200 epochs using the Adam optimizer with a learning rate of 10^{-3} . An auxiliary reconstruction loss is incorporated, whose weight is learned during training and constrained within the range $[0.5, 1.5]$, with a regularization coefficient $\lambda_{\text{red}} = 10^{-3}$. For imputation, the dropout probability threshold is set to $\tau = 10^{-3}$ to selectively recover dropout events. An evaluation of τ is presented in Supplementary Section 10. All hyperparameters are kept identical across datasets.

Additionally, scZiva and all baseline imputation methods are run five times independently with different random seeds set to 1, 12, 123, 1234, and 12345, respectively, to ensure the reliability of experimental results. The mean values of RMSE, MAE, and PCC from the five runs are reported as the imputation performance. In the case of generating imputed data for downstream analysis, the random seed is set to 1 for all cases.

Datasets and data processing

It is a fact that the ground truth of dropout events in real datasets is not available. Therefore, simulated datasets generated by Splatter tool [32] are employed to provide reliable reference values. In addition, real scRNA-seq datasets are used to assess the effectiveness of imputation through downstream analyses. In particular, eight simulated datasets

are generated based on Tung dataset [33] with missing rates ranging from 20% to 90%. Tung dataset is a real dataset, provided by default with Splatter tool, has 564 cells and 13,106 genes. Furthermore, 14 real datasets (12 single-cell RNA datasets and 2 bulk RNA datasets), presented in detail in Supplementary Section 3, are used to evaluate the support of scZiva and baseline methods for downstream analysis tasks.

For a fair comparison, all baseline methods are applied to the same gene filtering step, where all genes not expressed in any cell are discarded. Besides, the gene or cell filtering (if any) in the baseline methods is skipped to ensure the dimensions of the imputed data generated by the different methods are the same. Furthermore, we also apply the tasks of log2-transform and data normalization to all imputation methods but respect their original procedure (same as in [34]). In particular, if these two tasks are included in the methods, they are performed without modification. In the cases where a method does not perform one or both of these tasks throughout its algorithms, the remaining unperformed tasks are applied to the imputed datasets.

Results

scZiva effectively recovers missing values

Our method and five baseline methods are tested on eight datasets corresponding to different dropout rates from 20% to 90%, where the ground truths are known for assessment. Table 1 presents MAE scores of the imputation methods on the datasets. It can be seen from the table that scZiva achieves the best MAE results among the methods for all cases of dropout rates. Particularly, in the case of 70% dropout rates, the proposed method gets better MAE scores from 0.02 to 1.66 compared to the baseline methods. In addition, RMSE values of the imputation methods for the datasets are presented in Supplementary Table S2. scZiva achieves the lowest RMSE values in 6 out of 8 cases (dropout rates from 30% to 80%), and ranks second in the remaining cases (dropout rates 20% and 90%).

Figure 2 shows that the proposed method has the highest PCC values for all dropout rates. The performance of all methods decreases when dropout rates increase. It can be understood that higher dropout rates make the imputation more difficult. SAE-Impute and AcImpute have the lowest performance among the methods. While the performance of SAE-Impute declines significantly as the dropout rate increases, AcImpute is less affected by the increase in dropout rates, but it has the lowest PCC values for most cases.

Table 1 MAE values of scZiva and five baseline methods under different dropout rates

Dropout rate								
Method	20%	30%	40%	50%	60%	70%	80%	90%
CL-Impute	0.0201	0.0322	0.0472	0.0645	0.0807	0.1012	0.1255	0.1727
scRMD	0.3260	0.4077	0.5172	0.6218	0.7938	1.1246	1.6408	2.6686
SAE-Impute	0.2653	0.2757	0.2868	0.2989	0.3110	0.3241	0.3342	0.3356
AcImpute	0.8853	1.0848	1.2202	1.4572	1.5003	1.7294	1.8394	1.8823
SAVER	0.0030	0.0071	0.0162	0.0329	0.0551	0.0914	0.1387	0.2145
scZiva	0.0029	0.0062	0.0126	0.0245	0.0409	0.0706	0.1099	0.1809

The best results are highlighted in bold

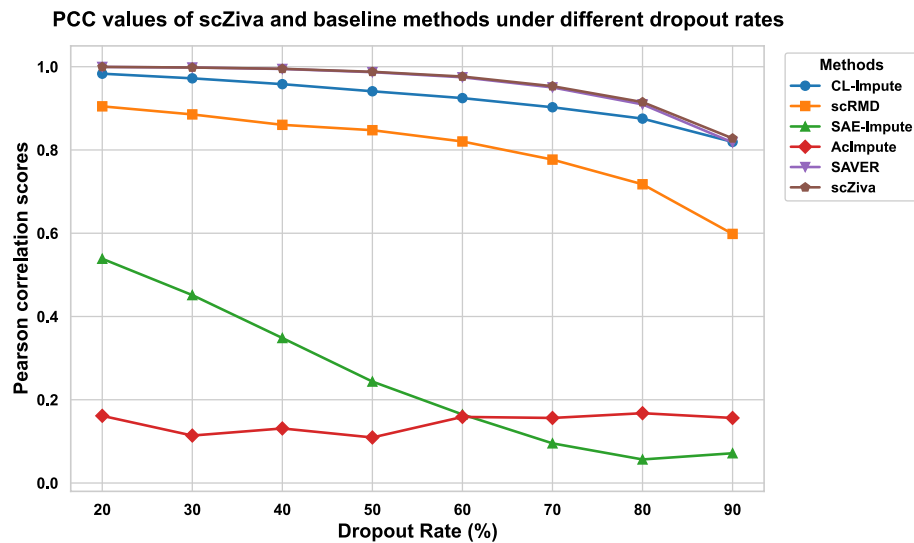


Fig. 2 PCC values of scZiva and five baseline methods under different dropout rates

scZiva improves cell type clustering

The clustering of cells is a considerable downstream task in scRNA-seq analysis, which aims to classify cells into groups of similar gene expression levels [35]. This experiment assesses the ability to support cell clustering of the imputation methods. scZiva and five baseline methods are firstly used to perform 10 real datasets (presented in Supplementary Table S1) to generate imputed expression matrices. The imputed matrices are then preprocessed by PCA [36] and clustered by k-means algorithm [37]. Besides, the same clustering task is also applied for the raw datasets, which are not imputed by any imputation methods.

Figure 3, Supplementary Table S3, and Table S4 present ARI and NMI scores of k-means on the imputed and raw datasets. It can be observed from the heatmaps that scZiva, SAVER, SAE-Impute, and CL-Impute are the four methods that are able to give better both average ARI and NMI scores than the case of the raw dataset. SAVER is the best algorithm among the methods, while scZiva is in second place with 6 of the 10 ARI scores and 3 of the 10 NMI scores ranking within the top two. Moreover, the proposed method achieves the highest average ARI and the second-highest average NMI among the methods.

scZiva supports the differential expression analysis

As one of the common downstream analyses of scRNA-seq data, the analysis of differential gene expression provides meaningful information to deeply understand diseases or distinguish cell types. The task aims to reveal the differentially expressed genes (DEGs) that show different activity levels between two or more groups of cells or tissues. In this work, we evaluate the impact of scZiva and the five baseline imputation methods on identifying differentially expressed genes. Following the process in previous studies [34], the Wilcoxon rank-sum test algorithm [38] is used to calculate DEGs for raw and imputed single-cell datasets. In addition, DEGs identified from bulk RNA-seq samples by the *limma* package [39] are treated as "gold standard" to assess results on single-cell samples with adjusted P-value set to smaller than 0.05.

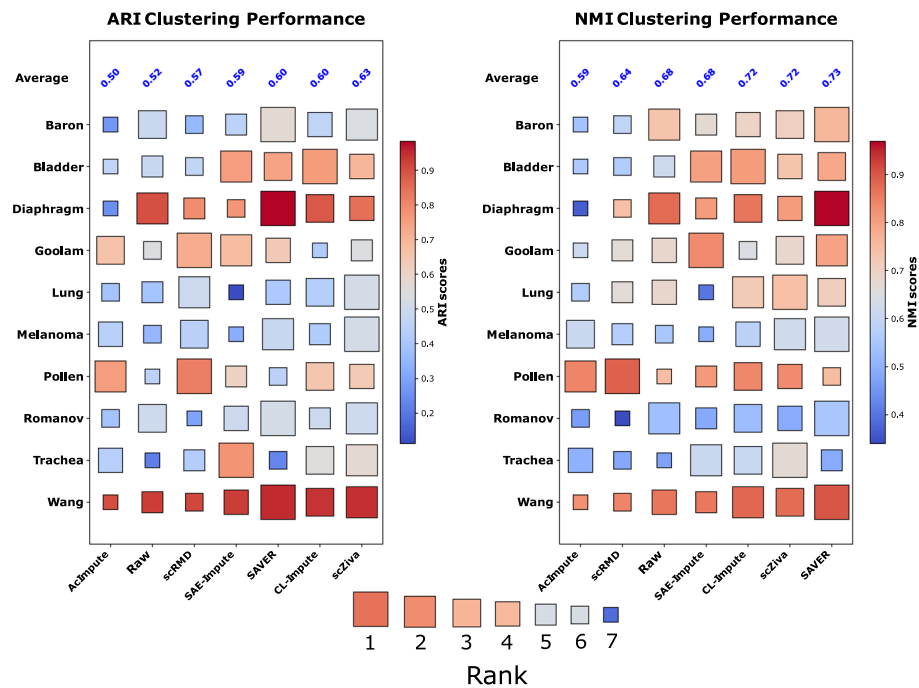


Fig. 3 The performance of different imputation methods in supporting the cell clustering

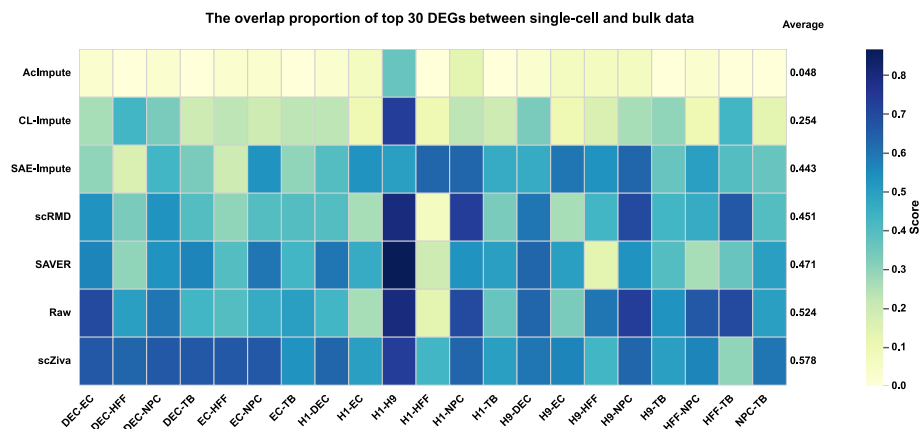


Fig. 4 The performance of scZiva and five baseline methods in supporting the identification of differentially expressed genes in GSE75748 dataset. Heatmap showing the value of overlap proportion for the top 30 DEGs between single-cell and bulk data. Each row shows imputation methods, while each column presents pairs of cell types

This experiment is performed on a real dataset available from the GEO database under the Series accession number GSE75748 [40]. The dataset, presented in Supplementary Section 3.2, consists of 1018 cells belonging to 7 cell types (H1, H9, EC, NPC, DEC, HFF, and TB), and has both single-cell and bulk samples. scZiva and five baseline methods are first used to generate six imputed scRNA-seq datasets. The DEGs in a raw scRNA-seq dataset, six imputed datasets, and a bulk sample are then calculated by the corresponding algorithms. We select the top 30 DEGs from scRNA-seq and bulk data to compare the overlapping levels.

The top 30 DEGs overlap proportion of all pairs of cell types from the results on single-cell and bulk datasets is presented in Fig. 4. It can be observed that scZiva achieves

the best results for 11/21 pairs. Separately compared with the case of raw dataset (without imputation), scZiva gets 13/21 better cases while SAE-Impute has 2/21 higher cases. SAVER and scRMD have only one pair, which has better overlap values compared with the case of raw data. On average, our method gets the highest overlap proportion, and improves 5.4% overlap values compared with the case of not using imputation. Additional experiments for the cases of different top DEGs, presented in Supplementary Figure S1, also illustrate the strength of the proposed methods for the differential expression analysis.

scZiva significantly enhances the cell trajectory inference

The cell trajectory inference task allows us to understand the dynamic development of cells, as well as aids in identifying new cell subpopulations [41]. In order to assess the support of scZiva on the task, this work performs experiments on the time-course scRNA-seq dataset (GSE75748), which differentiates from human embryonic stem cells to definitive endoderm in a time-course manner at 0, 12, 24, 36, 72, and 96 h. We use Monocle2 package [42] to calculate and visualize the cell trajectory and infer pseudo-time for raw and imputed datasets. The Kendall's rank correlation metric is used to assess the rank-based associations between predicted pseudo-time and true-time labels.

The bar chart in Fig. 5 presents Kendall correlation scores of the cell trajectory supported by the methods. It can be seen that our method gets the highest scores and is one of three methods (others are CL-Impute, and SAVER) that are able to improve the cell trajectory inference ability compared with the case of not using imputation. scRMD, AcImpute, and SAE-Impute reduce the cell trajectory performance compared with the case of the raw dataset. The visualization of the trajectory results in Fig. 6 and Supplementary Figure S14 illustrates that the case of scZiva produces a predicted cell trajectory with clearer pseudotime ordering compared to the other cases.

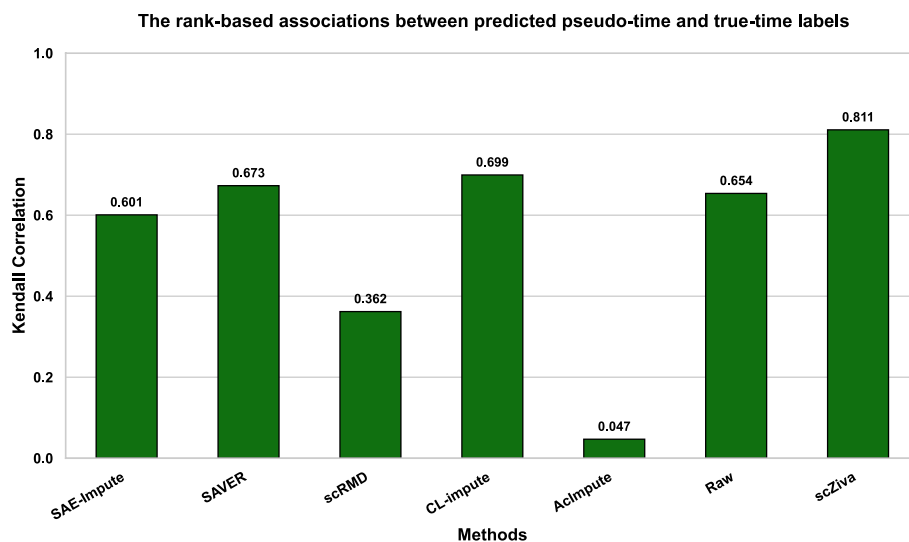


Fig. 5 The performance of imputation methods in the support of cell trajectory inference, and the case of raw GSE75748 dataset

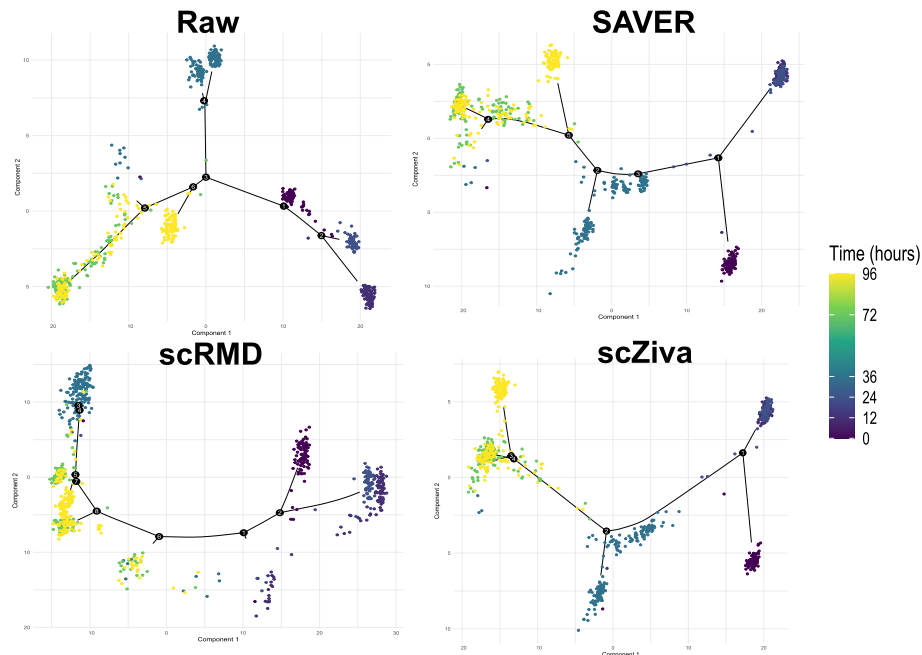


Fig. 6 The visualization of trajectory by time with the support of imputation methods (SAVER, scRMD, and scZiva), and the case of raw GSE75748 dataset

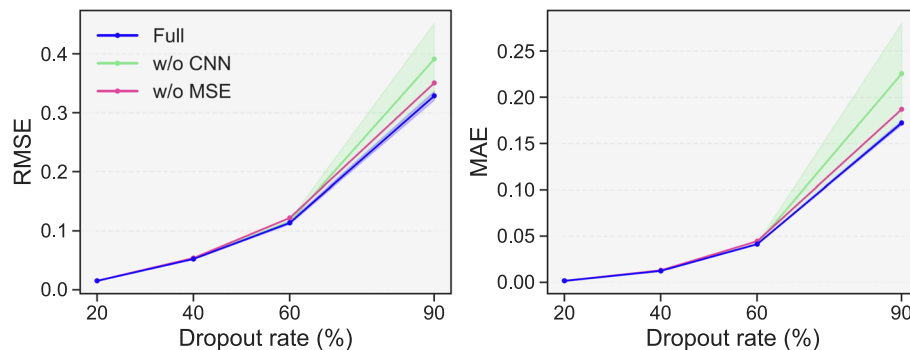


Fig. 7 The performance of the full scZiva model and its ablated variants under different dropout rates of Tung dataset. The full model generally achieves the best reconstruction accuracy

Ablation study

To validate the contribution of each component in scZiva, this work conducts an ablation study by removing key modules from the full model. Specifically, scZiva full model is compared with two variants: w/o CNN (removing the Conv1D module), and w/o MSE (removing the auxiliary MSE). All variants and scZiva are trained under identical settings on 20 random seeds across four dropout rates of Tung dataset. The average and standard deviation of RMSE and MAE returned by the methods are presented in Fig. 7.

The line charts show that scZiva consistently achieves the lowest RMSE and MAE across all dropout rates. At low dropout levels (20% to 40%), the performance advantage of the full model over its variants is less pronounced. However, this advantage becomes increasingly evident as the dropout rate increases. Removing the CNN-based module leads to higher errors and larger performance fluctuations, indicating that modeling local gene dependencies provides practical benefits for imputation. Similarly, removing

Table 2 Statistical significance testing between the full scZiva model and its ablated variants. Overall differences are assessed using the Friedman test, followed by one-sided Wilcoxon signed-rank tests with Bonferroni correction. Effect size is reported as *r*

Metric	Comparison	<i>p</i> _{raw}	<i>p</i> _{adj}	<i>r</i>
RMSE	Friedman	1.25×10^{-12}	–	–
	Full vs. w/o CNN	5.41×10^{-3}	1.62×10^{-2}	0.62
	Full vs. w/o Aux	9.54×10^{-7}	2.86×10^{-6}	1.10
MAE	Friedman	2.31×10^{-12}	–	–
	Full vs. w/o CNN	1.50×10^{-2}	4.50×10^{-2}	0.54
	Full vs. w/o Aux	9.54×10^{-7}	2.86×10^{-6}	1.10

the auxiliary MSE loss results in progressively larger reconstruction errors. We also show the statistical analysis using the Friedman test, followed by pairwise one-sided Wilcoxon signed-rank tests (with Bonferroni correction). The tests confirm that these differences are significant (*p* < 0.05) for all metrics, as shown in Table 2. These results demonstrate that each component of scZiva plays an important role in accurate imputation.

Discussion

Recovering gene expression values provides reliable biological information and supports the downstream analyses. The proposed method shows the strength in reconstructing missing values through different levels of dropout rates. This imputation is meaningful as it demonstrates the ability to improve the quality of downstream analytical tasks. In particular, compared with the case of performing on raw datasets, analyzing results on imputed datasets by scZiva return better performance for all cases of cell clustering, identifying the differentially expressed genes, and inferring cell trajectories. Compared with baseline methods, the proposed method shows competitive performance when being one of the two best methods for supporting the cell clustering task, and achieving the highest performance in the two remaining downstream analyses.

It is worth noting that the simulated datasets generated by Splatter are based on NB/ZINB-like assumptions, which are partially aligned with the decoder of scZiva. While such alignment helps us to control the quantitative evaluation under known ground truth, it may favor methods with a similar distribution. Therefore, the real scRNA-seq datasets analyzed in this study serve as a more comprehensive validation of the practical effectiveness of the proposed method.

The experiments for the cell trajectory inference illustrate that current imputation methods provide limited support to improve DEGs identification. These results are consistent with the observations reported in the benchmarking study [34]. Therefore, the improvement achieved by scZiva compared to the case of not using imputation is notable. However, further studies are needed to explore the reasons behind this limitation and to enhance the DEGs identification performance.

This work also shows that exploiting local gene-level features is beneficial in the analysis of scRNA-seq data. The integration of Conv1D layers into the VAE encoder significantly enhances the imputation quality. Moreover, although computational efficiency is not the primary concern of this study, experimental results shown in Supplementary Table S6 indicate that the computational aspect is also a strength of the proposed method.

Considering the effect of scZiva on widely studied genes, we further conducted an experimental analysis to examine how scZiva handles zero values in known housekeeping genes (*HPRT1*, *SDHA*) and cell type-specific marker genes (*CXCR4*, *CD34*, *SOX2*). *SOX2* is also a transcription factor. Supplementary Figures S3 to S13 in Supplementary Section 7 present that the proposed method imputes zero values without altering the biological characteristics of the genes. While the constitutive expression of the observed housekeeping genes is preserved, the analyzed marker genes remain highly enriched in their respective cell types. Thus, the biological meaning of gene expression is preserved after imputation. Furthermore, the expression level of the genes in the imputed data appears more consistent with bulk data compared to the raw data. However, further investigation into the impact of imputation on lowly expressed yet biologically important genes, such as transcription factors, may provide valuable insights for the improvement of imputation approaches.

While scZiva demonstrates strong performance across multiple benchmarks, it should be viewed as a methodological improvement within the current landscape of imputation approaches to scRNA-seq data. The complex biological and technical variability of single-cell transcriptomics remains difficult to fully capture within a single modeling framework. Future developments may integrate additional biological or multi-modal information to further enhance interpretability and robustness.

Conclusion

Dropout events in scRNA-seq caused by technical noise require effective methods to recover true expression values for enhancing the quality of downstream data analysis tasks. This study proposes an accurate scRNA-seq imputation method using a zero-inflated variational autoencoder. The main contributions of the proposed method includes leveraging ZINB distribution augmented with an auxiliary MSE term, as well as exploiting local gene-gene relationships to better preserve meaningful co-expression patterns using Conv1D layers. Furthermore, a probability-driven imputation strategy is also applied to enable an adaptive reconstruction of missing values in scRNA-seq data. The proposed method demonstrates competitive and robust performance compared to five baselines in recovering missing gene expression values and downstream analysis tasks.

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s12859-026-06422-2>.

Supplementary file 1 (pdf 2009 KB)

Acknowledgements

The authors would like to thank Ho Chi Minh City University of Technology and Engineering, the Ministry of Education and Training, Vietnam, for providing institutional and hosting support during this research.

Author contributions

LTV and VVL proposed the idea for the method, and equally contributed to the design of the method and experiments. LTV, VVL, QTH, and AQN developed the algorithm and conducted the experiments. All authors wrote and reviewed the manuscript.

Funding

This research was funded by the Ministry of Education and Training under project grant No. B2026-SPK-02, and was hosted by Ho Chi Minh City University of Technology and Engineering, Vietnam.

Data availability

All datasets used in this work can be directly downloaded from <https://zenodo.org/records/18263992>. The details and original references of the datasets also presented in Supplementary Data file. The source code of the proposed method are published in github page <https://github.com/tuanlvo293/scZiva>.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no conflict of interest.

Received: 21 January 2026 / Accepted: 9 March 2026

Published online: 19 March 2026

References

1. Tang F, Barbacioru C, Wang Y, Nordman E, Lee C, Xu N, et al. mrna-seq whole-transcriptome analysis of a single cell. *Nat Methods*. 2009;6(5):377–82.
2. Poirion OB, Zhu X, Ching T, Garmire L. Single-cell transcriptomics bioinformatics and computational challenges. *Front Genet*. 2016;7:163.
3. 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.
4. Huang M, Wang J, Torre E, Dueck H, Shaffer S, Bonasio R, et al. Saver: gene expression recovery for single-cell rna sequencing. *Nat Methods*. 2018;15(7):539–42.
5. Li WV, Li JJ. An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nat Commun*. 2018;9(1):997.
6. Van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*. 2018;174(3):716–29.
7. Zhang W, Liu T, Zhang H, Li Y. Acimpute: a constraint-enhancing smooth-based approach for imputing single-cell rna sequencing data. *Bioinformatics*. 2025;41(3):711.
8. Xu J, Cai L, Liao B, Zhu W, Yang J. Cmf-impute: an accurate imputation tool for single-cell rna-seq data. *Bioinformatics*. 2020;36(10):3139–47.
9. Chen C, Wu C, Wu L, Wang X, Deng M, Xi R. scrmr: imputation for single cell rna-seq data via robust matrix decomposition. *Bioinformatics*. 2020;36(10):3156–61.
10. Elyanow R, Dumitrescu B, Engelhardt BE, Raphael BJ. netnfm-sc: leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis. *Genome Res*. 2020;30(2):195–204.
11. Si T, Hopkins Z, Yanev J, Hou J, Gong H. A novel f-divergence based generative adversarial imputation method for scrna-seq data analysis. *PLoS ONE*. 2023;18(11):0292792.
12. Wang T, Zhao H, Xu Y, Wang Y, Shang X, Peng J, et al. scmultigan: cell-specific imputation for single-cell transcriptomes with multiple deep generative adversarial networks. *Brief Bioinform*. 2023;24(6):384.
13. Wu Y, Xu L, Cong X, Li H, Li Y. Scmaskgan: masked multi-scale cnn and attention-enhanced gan for scrna-seq dropout imputation. *BMC Bioinformatics*. 2025;26(1):130.
14. Huang Z, Wang J, Lu X, Mohd Zain A, Yu G. scggan: single-cell rna-seq imputation by graph-based generative adversarial network. *Brief Bioinform*. 2023;24(2):040.
15. Xu K, Cheong C, Veldsman WP, Lyu A, Cheung WK, Zhang L. Accurate and interpretable gene expression imputation on scrna-seq data using igsimpute. *Brief Bioinform*. 2023;24(3):124.
16. Chen S, Yan X, Zheng R, Li M. Bubble: a fast single-cell rna-seq imputation using an autoencoder constrained by bulk rna-seq data. *Brief Bioinform*. 2023;24(1):580.
17. 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.
18. Bai L, Ji B, Wang S. Sae-impute: imputation for single-cell data via subspace regression and auto-encoders. *BMC Bioinformatics*. 2024;25(1):317.
19. Alyassine W, Raju AS, Braytee A, Anaissi A, Naji, M: An efficient and reliable scrna-seq data imputation method using variational autoencoders. In: *The International Conference on Innovations in Computing Research*, pp. 84–97 (2024); Springer
20. Yau KK, Wang K, Lee AH. Zero-inflated negative binomial mixed regression modeling of over-dispersed count data with extra zeros. *Biometrical Journal journal of mathematical methods in biosciences*. 2003;45(4):437–52.
21. Gao C, Welch JD. Integrating single-cell multimodal epigenomic data using 1d convolutional neural networks. *Bioinformatics*. 2025;41(1):705.
22. Zhang W, Huckaby B, Talburt J, Weissman S, Yang MQ. cnnimpute: missing value recovery for single cell rna sequencing data. *Sci Rep*. 2024;14(1):3946.
23. Louridi N, El Ouahidi A, Benic C, Douzi S, El Ouahidi B. Vae-cnn for coronary artery disease prediction. In: *The International Workshop on Big Data and Business Intelligence*, pp. 155–164 (2024); Springer
24. Chen X, Ding M, Wang X, Xin Y, Mo S, Wang Y, et al. Context autoencoder for self-supervised representation learning. *Int J Comput Vision*. 2024;132(1):208–23.
25. Wang G, Liu W, He Y, Xu C, Ma L, Li H. Eegpt: Pretrained transformer for universal and reliable representation of eeg signals. *Adv Neural Inf Process Syst*. 2024;37:39249–80.

26. Wen L, Wang X, Liu J, Xu Z. Mveb: Self-supervised learning with multi-view entropy bottleneck. *IEEE Trans Pattern Anal Mach Intell.* 2024;46(9):6097–108.
27. Chai T, Draxler RR. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific model development discussions.* 2014;7(1):1525–34.
28. Shi Y, Wan J, Zhang X, Yin Y. Cl-impute: a contrastive learning-based imputation for dropout single-cell rna-seq data. *Comput Biol Med.* 2023;164:107263.
29. Hubert L, Arabie P. Comparing partitions *Journal of classification.* 1985;2:193–218.
30. Strehl A, Ghosh J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J Mach Learn Res.* 2002;3:583–617.
31. Abdi H. The kendall rank correlation coefficient. *Encyclopedia of measurement and statistics.* 2007;2:508–10.
32. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell rna sequencing data. *Genome Biol.* 2017;18(1):174.
33. Tung P-Y, Blischak JD, Hsiao CJ, Knowles DA, Burnett JE, Pritchard JK, et al. Batch effects and the effective design of single-cell gene expression studies. *Sci Rep.* 2017;7(1):39921.
34. Hou W, Ji Z, Ji H, Hicks SC. A systematic evaluation of single-cell rna-sequencing imputation methods. *Genome Biol.* 2020;21(1):218.
35. Molho D, Ding J, Tang W, Li Z, Wen H, Wang Y, et al. Deep learning in single-cell analysis. *ACM Transactions on Intelligent Systems and Technology.* 2024;15(3):1–62.
36. Abdi H, Williams LJ. Principal component analysis. *Wiley interdisciplinary reviews computational statistics.* 2010;2(4):433–59.
37. Lloyd S. Least squares quantization in pcm. *IEEE Trans Inf Theory.* 1982;28(2):129–37.
38. Bauer DF. Constructing confidence sets using rank statistics. *J Am Stat Assoc.* 1972;67(339):687–90.
39. Smyth G K. Limma: linear models for microarray data. In: *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, pp. 397–420. (2005); Springer, New York.
40. Chu L-F, Leng N, Zhang J, Hou Z, Mamott D, Vereide DT, et al. Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biol.* 2016;17(1):173.
41. Saelens W, Cannoodt R, Todorov H, Saey Y. A comparison of single-cell trajectory inference methods. *Nat Biotechnol.* 2019;37(5):547–54.
42. Trapnell C, Cacchiarelli D, Grimsby J, Pokharel P, Li S, Morse M, et al. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol.* 2014;32(4):381–6.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.