Cell-type composition analysis of scRNA-seq data with deep convolution neural network

He Ma¹², Zhihao Fang³, Zongbin Liu⁴ and Yan Chen⁵*

Abstract

Background: With the rapid development of single-cell RNA sequencing (scRNA-seq), more large-scale single-cell sequencing data has been generated. Due to the continuous increase of single-cell sequencing data, the analysis of cell-type composition from single-cell transcriptomics has also to face huge challenges. Since the emergence of scRNA-seq technology, the size of sequencing datasets has grown more than 1 million times in just over a decade. Meanwhile, as more gene markers are discovered, the data dimension of single-cell sequencing becomes higher. All of these put forward more stringent requirements on data dimensionality reduction and clustering algorithms. Under the constraints of practical factors such as occurrence of noise and dropouts and the limitation of overhead, it is also required an effective and efficient method that can obtain accurate analysis results in a very short time, and has a competitive algorithm stability.

Results: We present scCAE, an effective and efficient method based on convolution autoencoder that can accurately and rapidly analyze cell-type composition from single-cell transcriptomics datasets. Our method achieved the best results in the data sets that simulate the cell differentiation process among existing methods, which achieved the ARI of 69.64% and 68.83% at 10 and 25 clusters tasks. And, in the case of different dropouts, our method also works well. When the sparsity level of data metric is 71%, scCAE can achieved the ARI of 45.29%, which is the highest of the existing methods. In terms of algorithm overhead, our method has also achieved good results by comparing with several existing methods. It takes less time than most methods and takes up much less memory than other algorithms based neural networks.

Conclusions: Our method, scCAE, has more accurate and reasonable results in the analysis of cell-types composition. And, because of the design of imputer, it can deal with a large number of dropouts in the data matrix. Because of the structure of convolution network, scCAE has less time and space overhead than other deep-learning-based methods. Thus, we demonstrate that scCAE is a competitive method for analysis of cell-type composition from scRNA-seq data. We expect that our study can be a stepping stone for further prosperity of single-cell transcriptomics analysis.

Keywords: Convolution autoencoder; cell-type composition; single-cell RNA sequencing; deep-learning-based

1 Background

Single-cell RNA sequencing (scRNA-seq) is a cell transcriptomics research method based on the second generation gene sequencing technology. This technology allows high-throughput expression detection of genes at the single cell level, in-depth analysis of complex cell populations, and characterization of single-cell expression profiles to avoid heterogeneity of individual cell biological information caused by cell homogeneity is covered [1]. The advancement of microfluidic chips makes it possible to isolate a large number of cells.

With the improvement of RNA isolation and amplification methods, it is possible to analyze the transcriptional composition of individual cells using scRNA-seq technology [2–5].

The first scRNA-seq was published in 2009, analyzing only eight cells [6]. Seven years later, 10X Genomics has obtained a data set of more than 1.3 million cells. The appearance of such a large amount of data makes it possible to discover and analyze cell types. Large dataset ensure that the analysis has high accuracy and improves the ability to detect rare cell types. However, it also brings problems such as high dimensions, high dropouts, and excessive computational overhead.
Linear transformations, e.g., principal component analysis (PCA) [7–10], are unable to accurately capture the relationship between cells because of their high dropouts and noise. Non-linear approaches are more flexible as they provide results that are more intuitive and easier to interpret through visual observation. Currently, the most commonly used non-linear dimensionality reduction methods are t-SNE [11] and UMAP [12]. t-SNE calculates the transformation from high-dimensional space (Gaussian distribution) to two-dimensional space (t distribution), and minimizes the difference between all points in these two spatial distributions. However, t-SNE still has problems with long tails, tending to retain local features and slow calculation speed. UMAP uses manifold and projection calculations to achieve the purpose of dimensionality reduction of high-dimensional data.

The methods for dimensionality reduction and clustering of single-cell transcriptomics data mainly include graph-based PCA analysis methods represented by Scampy [10], Seurate [13], PhenoGraph [14], etc.; Data-driven K-means analysis methods represented by SIMLR [15]; Analysis methods based on neural networks represented by SAUCIE [16], DCA [17], AE [18–21], scScope [22], etc.; Other mathematical methods represented by ZINB-WaVE [23], MAGIC [24], etc. In practice, PCA is highly scalable and suitable for many types of data analysis. However, with smaller dataset, the accuracy will decrease significantly. SIMLR can effectively improve the sensitivity of noisy data sets through parallel training based on distance metrics, but because the algorithm can adjust the distance metrics to make cells suitable for clusters, it may artificially improve the calculation results and affect the real effect of the algorithm [25]. ZINB-WaVE provides a more stable and accurate low-dimensional representation of data without the need for preliminary standardization steps. However, this method has a huge flaw that cannot guarantee that the extracted low-dimensional signals are biologically relevant. Analysis methods based on neural networks developed in recent years due to the rise of deep learning, and have achieved good results. For example, scScope uses a recurrent network layer to iteratively interpolate the dropouts of the input scRNA-seq data, which can more efficiently and accurately perform dimensionality reduction and clustering on the data, especially for large dataset.

As a hot topic in machine learning field in recent years, deep learning is very suitable for life sciences and medical fields. Convolutional neural networks (CNNs) improve the training performance by using spatial relationships to reduce parameters and are the first true multi-layer structure learning algorithms [26]. On this basis, fully convolutional neural networks (FCNs) [27] are proposed, and the further developed U-net is widely used in the field of medical image segmentation [28]. In the field of biological information, deep learning algorithms are also gradually being widely used. Convolutional neural networks can predict molecular properties, thereby further deducing the interaction between drug molecules [29]. Through the use of reinforcement learning and generative adversarial networks, it is possible to find drug structures that meet the conditions according to specified goals, thereby guiding the development of new drugs [30]. In genetic analysis, by converting genetic data into images and using neural network-based image recognition models, the differences of the genome are found [31]. To summarize, the adoption of deep learning methods has brought more possibilities to the analysis of biological information.

Based on the idea of deep learning, we further proposed a convolution-neural-network-based method, named as scCAE. It uses a symmetrical convolutional layer structure at both ends, and an autoencoding structure in the middle. We define this structure as a convolutional autoencoder. In addition, our network also uses a skip layer similar to Unet, and an imputer that handles the dropouts present in the data. In experiments, we found that in the case of scCAE with imputer, compared with the structure without recurrent imputer, the analysis of cell-type composition has been significantly improved, especially in the case of more dropouts. In the end, a standard scCAE method process also needs to consider the quality control, batch correction and louvain clustering. The main contributions of this work are listed as follows.

1 **Increased accuracy and complicated inference.** Accuracy is the primary criterion for evaluating such methods. In terms of the effectiveness of existing algorithms, scCAE has made a significant improvement in the accuracy of cell-type analysis, especially in some more complicated scRNA-seq data, where there are specialized groups of cells differentiation. This is mainly reflected in the higher Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and Silhouette Width (SW) compared to other methods. In addition, our method possesses relatively stable accuracy in experiments and is not easily affected by initial randomness.

2 **Better handling of a large number of dropouts.** Single-cell RNA sequencing usually has more dropouts and noise than groups of cells sequencing. Generally, in cell expression matrices, matrix sparsity greater than 50% is a common phenomenon. The method in this paper effectively eliminates these interferences to the results by using imputer and convolutional layers. Even when
the matrix sparsity exceeds 70%, scCAE still works well.

3 Large scale dataset and less overhead. Faced with the continuous accumulation and development of single-cell RNA sequencing, data sets are becoming larger, and the number and type of cells that need to be analyzed at one time is also increasing. scCAE uses a convolution structure and shares convolution kernels, which greatly reduces the time and space overhead. It supports GPU parallel computing, and takes up less memory space, which means that when handle with big data, our method will be more effective and efficient.

The rest of this paper is organized as follows. Section 2 mainly introduces the specific structure of scCAE and the simulation data to be prepared for subsequent experiments. Section 3 compares scCAE and other methods through experiments on the dataset. The dataset are mainly divided into two types, one is generated by simulation in this paper, and the other is the existing real-world data from Internet and our laboratory. Section 4 concludes the whole paper.

2 Methods
2.1 scCAE model
The main structure of scCAE network adopts a structure similar to U-net as shown in Fig. 1. Under this structure, the network can inherit the characteristics of U-net suitable for large scale data processing, which is reflected in high-dimensional processing capabilities on single-cell RNA sequencing data. This is mainly due to the application of convolutional layer and pooling layer. The convolutional layer helps to identify more local features from the dataset, and the pooling layer helps the network reduce the number of output values by reducing the size of the input.

Due to the particularity of the single-cell sequencing data structure, which is a two-dimensional matrix, the feature information formed by the expression level between different cell individuals is meaningless. Therefore, when data are input, they cannot be input as the form of a two-dimensional matrix. It should be split from the cell barcode to obtain the expression vector of a single cell. In addition, in order to retain as much of the original expression level information as possible, the input of scCAE uses the size of [1, 2048]. Since the input data is a one-dimensional vector, the convolutional layer used by scCAE is a one-dimensional convolution instead of the two-dimensional convolution commonly used in image processing. The pooling layer is also a one-dimensional pooling.

Because the main task of the network is dimensionality reduction and cell-type composition analysis rather than image segmentation, the output of U-net, heatmap structure, will not be used in the middle hidden layer. We added the structure of the neural network autoencoder. This structure can compress the input data with information obtained after convolution. As convolution and pooling layer has effectively extracted the features of the data, the coding layer requires too many neural nodes, which greatly reduces the overhead. For the upsampling part, scCAE does not adopt the U-net’s channel fusion strategy. This is because in the dimensionality reduction task, if use this fusion strategy, the output of the upsampling will rely excessively on the output of the previous convolutional layer rather than the autoencoder. This situation is not conducive to training. For this reason, the upsampling part of scCAE is designed without channel fusion strategy, but the idea of skip layer is retained. We adopt a strategy where the weight sum is 1, and the weighted addition of the two inputs, which not only ensures the efficiency of training, but also makes the network more symmetrical.

Due to the dropouts in the real-world dataset, on this basis, the imputer is added, which has ability to impute the adverse impact of dropouts and noise in sequencing data. In scCAE, the imputer adopts a three-layer neural network structure as 8\times128, 128, 4\times128, which is connected to the output of the convolutional layer and the input of the autoencoder layer by means of a recurrent neural network.

Before the data are input, they also need to be preprocessed such as quality control and elimination of batch effects. After obtaining the dimensionality reduction results through the scCAE network, the clustering method needs to be considered. The output of scCAE is a one-dimensional feature vector, which can be easily integrated with any clustering method. For general-scale data, we use the Louvain method [32] for the final cluster analysis. Louvain Method is superior to other methods in terms of calculation time and has higher accuracy. In the large-scale analysis task, we use the experience of scScope [22] method. Due to the extremely high calculation cost and memory requirements in the graph construction, PhenoGraph [14] can not process millions of cells. By using the combined processing of K-means and PhenoGraph, this problem can be solved well. We first divide the data into several clusters with the idea of divide and conquer, first use k-means for initial clustering, and then use PhenoGraph for analysis.

2.2 Training function
The parameters in the scCAE network are mainly obtained through end-to-end training. Due to the variability of parameters, scCAE is insensitive to whether the input data has been normalized and other steps.
Nonetheless, we recommend normalizing the data in practice, because it may help the convergence rate of training. Considering that there are batch effects in general experimental data, scCAE provides a batch correction layer. This is an optional process and is given by

\[ x_c = f_{Batch}(\tilde{x}_c) = lr(\tilde{x}_c - B u_c), \]

where \( f_{Batch} \) is batch correction layer, and \( \tilde{x}_c \in \mathbb{R}^N \) denotes the original input single-cell expression vector. \( lr \) is activation function, leaky Relu. \( B \) is batch correction matrix which needs to be updated by learning and \( u_c \) is binary experimental batch effects indicator vector.

When downsampling through the convolution and pooling layer, the downsampling layer \( f_{Conv} \) is given by

\[ h_{c1} = f_{Conv}(x_c) = P_{Max}[lr(W_{Conv} \ast x_c + b_1)], \]

where \( P_{Max} \) denotes max pooling layer, and \( W_{Conv} \) is learnable convolution kernel. \( b_1 \in \mathbb{R}^M \) is bias unit. After this layer, the single-cell expression profile is compressed from N-dimension into M-dimension, \( M < N \). Note that the convolution and pooling operations here are one-dimensional.

Then, the data will be compressed again with encoder layer \( f_{En} \) by

\[ h_{c2} = f_{En}(h_{c1}) = lr(W_{En}h_{c1} + b_2), \]

where \( W_{En} \in \mathbb{R}^{L \times M} \) is learnable weights and \( b_2 \in \mathbb{R}^L \) is bias, \( L < M \).

The decoder layer \( f_{De} \) decompresses the latent representation \( h_{c2} \) and restores it to the M-dimensional space, which is consistent with the dimension obtained by the downsampling, and is given by

\[ h_{c3} = f_{De}(h_{c2}) = lr(W_{De}h_{c2} + b_3), \]

where \( W_{De} \in \mathbb{R}^{M \times L} \) is learnable weights and \( b_3 \in \mathbb{R}^M \) is bias.

Then, the decoded latent representation will be decompressed again with upsampling layer \( f_{Deconv} \), which is given by

\[ y_c = f_{Deconv}(h_{c1}, h_{c3}) = lr[W_{Deconv}^T(ah_{c1} + bh_{c3}) + b_4], \]

where \( W_{Deconv}^T \) is learnable transposed convolution kernel in deconvolution operation. \( a \) and \( b \) is constant weight, \( 1 = a + b \). And \( b_3 \in \mathbb{R}^N \) is bias.

For Imputer, we propose two ideas, which are two different functions. The first idea is to directly connect Imputer as a neural network. Because convolution can effectively reduce the impact of dropouts and compress the expression vector, Imputer’s main role at this time is to deepen the network and make the network have a stronger inference ability, and is given by

\[ u_c = f_I(y_c) = lr(W_Iy_c + b_I), \]

where \( W_I \in \mathbb{R}^{M \times N} \) is learnable weights and \( b_I \in \mathbb{R}^M \) is bias. With this idea, \( v_c = u_c \) directly recurs to the front network.

The second idea emphasizes that the imputer plays the role of self-correction in it, and adopts a similar step function \( f_{Step} \) for \( u_c \) to obtain \( v_c \). Then, the imputer performs by

\[ v_c(i) = f_{Step}(u_c; h_{c1}) = \begin{cases} 0 & h_{c1}(i) > \text{Threshold} \\ u_c(i) & h_{c1}(i) < \text{Threshold} \end{cases}, \]

where \( v_c \in \mathbb{R}^M \) and \( 0 < i \leq M, \text{Threshold} > 0 \). \( v_c(i), u_c(i) \) and \( h_{c1}(i) \) denote the \( i \)th element in latent representation \( v_c, u_c \) and \( h_{c1} \) respectively.

After obtaining the vector \( v_c \) from imputer, the new latent representation \( h_{c2} \) will be updated as \( h'_{c2} = h_{c1} + v_c \). Then resent \( h'_{c2} \) into encoder, decoder and upsampling layers and learn to get the latent expression vector. Considering the recurrent structure of scCAE, the above function can be updated as

\[ x_c = f_{Batch}(\tilde{x}_c), h_{c1} = f_{Conv}(x_c), h'_{c1} = h_{c1} + v_{c, t-1}, \]

\[ h'_{c2} = f_{En}(h'_{c1}), h'_{c3} = f_{De}(h'_{c2}), y_c = f_{Deconv}(h_{c1}, h'_{c3}), \]

\[ u_{c, t} = f_I(y_c), v_{c, t} = u_{c, t} \text{ or } f_{Step}(u_{c, t}), v_{c, 0} = 0, \]

where \( t = 1 \cdots T \), when we designed scCAE, the value of \( T \) was 2. If \( T = 1 \), there’s no imputer structure in scCAE, we called scCAE-nonImputer.

This paper uses MSEloss (L2 loss) as the loss function, which can be described as

\[ f = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \| y_c^T(i) - x_c(i) \|^2, \]

where \( i = 1 \cdots N \) denotes each gene expression in single-cell profile, and we set \( N = 2048 \). For the choice of optimizer, This paper chose Adam optimizer, and learning rate is 1e-3, betas = (0.5, 0.999).

2.3 Data generation

In order to evaluate the scCAE method, the datasets required for the experiments in this paper are three types, one is the simulation datasets generated by the
Splatter [33] package in R language; another is the dataset published by 10x Genomics; and the other is the dataset of circulating tumor cells (CTCs) obtained by scRNA-seq in our laboratory. Splatter is an R package for the simple simulation of single-cell RNA-sequencing data. Since the existing scRNA-seq datasets generally have a large number of cells, and it is impossible to visually observe the cells for labeling, it is difficult to use the existing real datasets for the comprehensive evaluation of our method. Therefore, in this paper, most of the algorithm evaluation datasets need to be generated by splatter and labeled with classification.

As shown in Fig. 2, Fig. 2(a) is an example of simple clustering datasets. The cells are divided into two types, and the boundaries on PC1 and PC2 planes are clear, and the distribution of data points conforms to a uniform normal distribution. Fig. 2(b) is an example of datasets that simulates the differentiation process. Cells are divided into ten categories and the direction of differentiation on the PC1 and PC2 planes is clear, but the boundaries are not clear. Fig. 2(c) is an example of datasets with dropouts for expression. The boundaries between PC1 and PC2 planes are not obvious, and the distribution of various types of data points is uneven and not concentrated.

For simple clustering datasets, the dataset is generally interested in studying mixed cell populations, and wants to understand the existing cell types or the differences between them. we set the parameter nGenes = 2048, method = "groups", the number of types is set to 5/10/20/30/50, the number of cells in each type can be generated randomly.

For cell clustering datasets (complex clustering datasets) that simulate the differentiation process, one type of cell is generally being transformed into another. Splatter approximates this process by simulating a series of steps between two groups and randomly assigning each cell to a step. we set the parameter nGenes = 2048, method = "paths", de.prob = 0.2, the number of types is set to 10/25, the number of cells in each type can be generated randomly.

For datasets with dropouts in cellular RNA expression, generally part of the gene expression in the cell is not detected, which is reflected in the sparseness of the expression matrix. we set the parameters nGenes = 2048, dropout.type = "experiment", dropout.shape = -1, dropout.mindim= 0/1/2/3, the number of types is set to 10, the number of cells in each type can be generated randomly.

In addition, we also used two kinds of existing datasets to evaluate scCAE. One is the dataset from 10x Genomics (http://10xgenomics.com), a dataset containing 33K PBMCs from a Healthy Donor. We use it to evaluate the time cost of our method. And the other one is the scRNA-seq data processed precise isolation of circulating tumor cells (CTCs) from blood samples in our lab, which contains a total of 3243 cells were sequenced with a median of 3226 genes per cell. And there were approximately 17 CTCs in total. By analyzing these existing datasets, we look forward to better assessing the practicality of scCAE.

3 Results
Before analyzing the cell-types composition, it is necessary to further get the appropriate parameters for scCAE and scCAE-nonImputer in the upsampling layer. We adjusted the weights and conducted experiments on the cell clustering datasets (complex clustering datasets) that simulate the differentiation process, as shown in Fig. 3. Under the constraint of $a + b = 1$, weight $a$ was continuously increased from 0.5 to 0.99. A higher value of $a$ means that the input of the up-sampling is more affected by the output of the down-sampling of the convolution network, and the less it is affected by the output of the autoencoder. It can be found through experiments that in the scCAE-nonImputer model, when $a$ is in the range of $[0.5,0.8]$, ARI increases with the increase of $a$. When $a$ is in the range of $[0.8, 0.99]$, ARI drops sharply with the increase of $a$, indicating that the dimensionality reduction capability of the model has been sharply reduced by the parameters at this time, and depends more on the direct connection between down-sampling and up-sampling layer, rather than the autoencoder structure. When the scCAE model was used for the experiments, a double peak appeared, and the local optimum of ARI was obtained when $a = 0.6$ and $a = 0.8$, respectively, and the global optimal ARI was obtained when $a = 0.8$. It can be concluded that when $a = 0.8$ and $b = 0.2$, that is, $X_3(s) = 0.8X_1(s) + 0.2X_2(s)$, scCAE and scCAE-nonImputer performed best. Therefore, we suggest setting the parameters $a = 0.8$ and $b = 0.2$ in the scCAE and scCAE-nonImputer models. Subsequent experiments in this paper will also use these parameters.

After the parameters are determined, the scCAE, scCAE-nonImputer and other existing methods are tested and compared in the simple clustering datasets, thereby obtaining box-plot Figure 4 based on the ARI results.

In order to ensure the reliability of the experimental results, we used several methods to perform cluster analysis on some simulation simple clustering datasets generated when the number of clusters was 5, 10, 20, 30, and 50, respectively. According to Fig. 4, on the simple datasets, when the actual number of
clustering datasets that simulate the different-

effects of the scCAE and scCAE-nonImputer.

simple clustering datasets are still far from

In summary, scCAE has achieved outstanding results in the complex clus-

ting task of simulated single-cell RNA sequencing datasets, which is superior to existing methods.

Single-cell RNA sequencing data usually has dropouts and noise. In order to evaluate the missing value adap-
tation and compensation ability of our method, we simulated a variety of expression matrices by Splatter

However, the simple clustering datasets are still far away from the real datasets, which does not simulate

paper, experiments were conducted on the number of clusters of 10 and 25 respectively, and the results of multiple experiments were recorded for each cluster number. Fig. 5 presented the ARI obtained from each group of experiments.

As shown in the figure, when the number of clusters is 10, scCAE performs best, and its optimal ARI can exceed 0.7, and the overall performance is around 0.7. The performance is followed by scScope, whose overall ARI performance is slightly below 0.7. Although the ARI of AE can exceed 0.7, but because the results are too scattered in the figure, it is very unstable. The ARI of scCAE-nonImputer and PCA are generally around 0.65, and the experimental results are very stable. The performance of SAUCIE and DCA is far from impressive. When the number of clusters is 25, which is close to the usual large scRNA-seq datasets, scCAE still performs best. The performance of scCAE-nonImputer is similar to that when the number of clusters is 10. But, the ARI of scScope has dropped significantly, the clustering results are not good and extremely unstable. PCA decreased slightly, and its stability slightly weakened.

The evaluation statistics of ARI, NMI, SW, and DBI (short for Davies Bouldin Index) obtained from the experiment results are summarized in Table 2. When the number of clusters is 10, scCAE’s ARI, NMI, and SW all show that the algorithm has the best effect of analysis. The scScope performed slightly worse than scCAE-nonImputer and PCA is similar to scCAE-nonImputer by the evaluations. When the number of clusters is 25, the dataset are very complex and there are many types of cells, which further tests these methods. At this time, our method, scCAE, still performed the best, whose ARI is 7% points higher than scScope and 6% points higher than PCA. Moreover, the SW index is even better, which reaches 0.2 (0.2665) among these methods. Similarly, scCAE-nonImputer has achieved very good results. In summary, scCAE has achieved outstanding results in the complex clustering task of simulated single-cell RNA sequencing datasets, which is superior to existing methods.

Single-cell RNA sequencing data usually has dropouts and noise. In order to evaluate the missing value adap-
tation and compensation ability of our method, we simulated a variety of expression matrices by Splatter

However, the simple clustering datasets are still far away from the real datasets, which does not simulate

In this

Single-cell RNA sequencing data usually has dropouts and noise. In order to evaluate the missing value adap-
tation and compensation ability of our method, we simulated a variety of expression matrices by Splatter

However, the simple clustering datasets are still far away from the real datasets, which does not simulate

In this
a greater impact. But scCAE and scCAE-nonImputer are hardly affected by this change in sparsity. When the sparsity is 45%, the effect of scCAE-nonImputer is significantly reduced but scCAE performs well. We believe this is due to the lack of imputer for scCAE-nonImputer. When the sparseness of the matrix continues to increase, which means the dropouts become more, it can be inferred that the effect of scCAE still performs best in several methods. From the perspective of ARI, scScope and scCAE are similar, but other methods have been greatly affected. It can be concluded that when there are few missing data in the scRNA-seq data, both scCAE and existing methods can be well analyzed. However, when there are a lot of missing values in the data, the effect of scCAE is far superior to other methods, and slightly better than scScope.

In order to intuitively see the clustering results in the case of high sparsity, this paper visualized the results of some methods when the sparsity is 58%, 72% and the number of clusters is 10, as shown in Fig. 6(b). When the sparsity of the expression matrix is 58%, only the number of clusters inferred by scCAE is 10 types, which is consistent with groundtruth, while PCA, AE, and scScope are 9, 13, and 11 types, respectively. It can be found that the scCAE performance is well, although a small number of cells will be misclassified. When the sparsity of the expression matrix is 72%, the number of clusters inferred by scCAE is 11 that is close to groundtruth. From the perspective of visualization, the classification of groundtruth is similar to vertical layering, and the result of our method is closest to groundtruth. While the clustering results of the AE and scScope algorithms have been unable to see the basic outline of clustering through visualization. From the visualization results, it can be concluded that scCAE has outstanding performance to analyze cell-type composition from single-cell transcriptomics datasets with dropouts.

This paper compared the time of analyzing multiple size single-cell RNA sequencing datasets by using scCAE and some existing algorithms, as shown in Tab. 4. The experiment is performed by extracting 10K, 50K, 100K, 500K, and 1M cells from the 10X Genomics dataset. We run these methods on the server with Intel Xeon E5 CPU and GTX 1080ti GPU. It can be concluded from the table that the running speed of scCAE is still very fast, slightly slower than PCA, faster than scScope and other methods. In addition, we analyzed the memory cost of four neural networks algorithms, scCAE, scCAE, AE, and scScope, as shown in Tab. 4. By inputting the data of [1,2048] to each network, we can obtain the number of parameters used and the size of network. scCAE-nonImputer has the fewest parameters, only 677,824. Since scCAE added Imputer, the parameters reached 2,110,336. Even so, it is still less than half of the number of scScope network parameters, which is 5,353,664. And, in terms of memory size, scCAE takes up very little, about half of scScope. We believe that this increase in speed and decrease in memory is due to the structure of the convolution layer in scCAE.

To verify the effect of scCAE method on the accuracy of clustering in actual datasets, we applied scCAE to analyze cell-type composition from single-cell transcriptomics datasets with CTCs. This dataset was collected from our laboratory. We have developed a microfluidic device with an innovative hydrodynamic structure to separate circulating tumor cells (CTCs) from cancer patient blood samples. The device demonstrated excellent cell enrichment performance, and the isolated cell solution contained high purity of CTCs and some white blood cells, while maintaining very high viability (>98%), highly beneficial for single-cell sequencing. The isolated cell solution was split into two tubes, one used for immunostaining, and another in which single-cell RNA sequencing was conducted. The immunostaining results confirmed that 17 CTCs existed in approximately 3200 cells. (Currently, the method and dataset are being submitted). Even so, the content of CTCs is still not easily analyzed in cell samples. We have adopted many other solutions, which hardly analyze it well. We hope that scCAE can distinguish cell-type of CTCs well, and maintain the integrity and accuracy of CTCs while maintaining the integrity and rationality of other cell populations. The visualization results are shown in the Fig. 7(a), which proves that our method accurately identifies CTCs (Class 12 in red box). In Class 12, the expression of KRT8, KRT18, PPBP and other gene markers in the cells were significantly up-regulated, as shown in the Fig. 7(b). Epithelial markers such as Keratin 8 (KRT8) and Keratin 18 (KRT18) are commonly used markers to identify cancer cells. Taking into account the number of cells, we firmly believe that such cells are CTCs. In addition, according to the Fig. 7(b), scCAE reasonably distinguishes other cell types. The good performance of scCAE on the CTCs dataset once again proved that the method has excellent adaptability, and is particularly suitable for discovering a small number of cell sets or subsets based on different expressions between cells.

4 Conclusion

This paper proposes an effective and efficient method based on convolution autoencoder, which aims to accurately and rapidly analyze cell-type composition from single-cell transcriptomics. Through its performance on single-cell RNA sequencing datasets and comparison with other similar methods, it is found that the
scCAE method is superior to the existing methods in terms of clustering results, dropouts compensation, clustering speed, and computational overhead. This article innovatively combines the convolutional neural network and the autoencoder to obtain the scCAE network that has achieved good results in the field of biological information analysis. We believe this structure will also achieve excellent results in other similar fields in the future. It can also be concluded from the experiments in this paper that in the case of too large number of clusters and too many dropouts, the performance of scCAE still have a certain room for improvement, and we will continue to improve it and make it adaptable in various biological applications.

Declarations

Abbreviations
ARI: Adjusted rand index; DBI: Davies-Bouldin index; NMI: Normalized mutual information; RNA-seq: RNA sequencing; scRNA-seq: Single cell RNA sequencing; SW: Silhouette Width.

Ethics approval and consent to participate
Blood samples from NSCLC patients were collected and processed in accordance with an IRB (Institutional Review Board) approved protocol (SIAT IRB-190315-H0338) in Shenzhen Institutes of Advanced Technology. All patients provided informed consent to participate in this study.

Consent for publication
Not applicable.

Availability of data and materials
The single cell RNAseq data and the code to generate it are publicly available in Github (https://github.com/SIAT-Fang/scCAE_data).

Funding
This work is supported by the National Natural Science Foundation of China (61875221, 61702087), the Guangdong Science and Technology Research Program (2019A050503008), the Fundamental Research Program of Shenzhen (JCYJ20170413153034718), the Fundamental Research Funds for the Central Universities (N172008008) and the Youth Innovation Promotion Association CAS (2016321).

Competing interests
The authors declare that they have no competing interests.

Author’s contributions
HM and YC conceived, deduced the algorithm. ZF performed the data analysis. HM and ZF wrote the manuscript. All authors read and approved this manuscript.

Acknowledgements
Not applicable.

Author details
1 College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China. 2 Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, Shenyang, China. 3 College of Engineering, Peking University, Beijing, China. 4 Shenzhen Ziggaz Biotechnology Ltd, 518055, Shenzhen, China. 5 Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 518055, Shenzhen, China.

References
1. Chu, Y., Corey, D.R.: RNA sequencing: platform selection, experimental design, and data interpretation. Nucleic acid therapeutics 22(4), 271–274 (2012)
2. Cusanovich, D.A., Redding, J.P., Garfield, D.A., Daza, R.M., Aghamirzaie, D., Marco-Ferreeres, R., Pliner, H.A., Christiansen, L., Qiu, X., Steemers, F.J., et al.: The cis-regulatory dynamics of embryonic development at single-cell resolution. Nature 555(7697), 538–542 (2018)
3. Davie, K., Janssens, J., Koldere, D., De Waegeneer, M., Pech, U., Kreft, L., Albar, S., Makhzami, S., Christaens, V., Gonzalez-Blas, C.B., et al.: A single-cell transcriptome atlas of the aging drosophila brain. Cell 174(4), 982–998 (2018)
4. Han, X., Wang, R., Zhou, Y., Fei, L., Sun, H., Lai, S., Saadatpour, A., Zhou, Z., Chen, H., Ye, F., et al.: Mapping the mouse cell atlas by microwell-sequ. Cell 172(5), 1091–1107 (2018)
5. Zengel, A., Hochgener, H., L¨ onneberg, P., Johnson, A., Memic, F., Van Der Zwan, J., H¨ aring, M., Braun, E., Borm, L.E., La Manno, G., et al.: Molecular architecture of the mouse nervous system. Cell 174(4), 999–1014 (2018)
6. Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B.B., Siddiqui, A., et al.: mmna-seq whole-transcriptome analysis of a single cell. Nature methods 6(5), 377 (2009)
7. Ji, Z., Ji, H.: Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. Nucleic acids research 44(13), 117–117 (2016)
8. Kiselev, V.Y., Kirschner, K., Schaub, M.T., Andrews, T., Yiu, A., Chandna, T., Natarajan, K.K., Reik, W., Barabona, M., Green, A.R., et al.: Sc3: consensus clustering of single-cell rna-seq data. Nature methods 14(5), 483–486 (2017)
9. Tasic, B., Menon, V., Nguyen, T.N., Kim, T.K., Jarosky, T., Yao, Z., Levi, B., Gray, L.T., Sorenson, S.A., Dolbeare, T., et al.: Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. Nature neuroscience 19(2), 335 (2016)
10. Wolf, F.A., Angerer, P., Theis, F.J.: Scany: large-scale single-cell gene expression data analysis. Genome biology 19(1), 15 (2018)
11. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research 9(Nov), 2579–2605 (2008)
12. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
13. Marcosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A.R., Kamitaki, N., Martersteck, E.M., et al.: Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 161(5), 1202–1214 (2015)
14. Levine, J.H., Simonds, E.F., Bendall, S.C., Davis, K.L., El-ad, D.A., Tadmor, M.D., Litvin, O., Fienberg, H.G., Jager, A., Zunder, E.R., et al.: Data-driven phenotypic dissection of amn reveals progenitor-like cells that correlate with prognosis. Cell 162(1), 184–197 (2015)
15. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D., Batzoglou, S.: Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. Nature methods 14(4), 414 (2017)
16. Amadio, M., Van Dijk, D., Slima, K., Chen, W.S., Mohsen, H., Moon, K.R., Campbell, A., Zhao, Y., Wang, X., Venkatasswamy, M., et al.: Exploring single-cell data with deep multitasking neural networks. Nature methods, 1–7 (2019)
17. Eraslan, G., Simon, L.M., Mircea, M., Mueller, N.S., Theis, F.J.: Single-cell rna-seq denoising using a deep count autoencoder. Nature communications 10(1), 1–14 (2019)
18. Teixeira, V., Camacho, R., Ferreira, P.G.: Learning influential genes on cancer gene expression data with stacked denoising autoencoders. In: 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1201–1205 (2017). IEEE
19. Tan, J., Hammond, J.H., Hogan, D.A., Greene, C.S.: Adage-based integration of publicly available pseudomonas aeruginosa gene expression data with denoising autoencoders illuminates microbe-host interactions. MSystems 1(1) (2016)
20. Beaulieu-Jones, B.K., Moore, J.H.: Missing data imputation in the electronic health record using deeply learned autoencoders. In: Pacific Symposium on Biocomputing 2017, pp. 207–218 (2017). World Scientific
21. Xie, R., Wen, J., Quitadamo, A., Cheng, J., Shi, X.: A deep auto-encoder model for gene expression prediction. BMC genomics 18(9), 845 (2017)
22. Deng, Y., Bao, F., Dai, Q., Wu, L.F., Altschuler, S.J.: Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. Nature methods 16(4), 311–314 (2019)

23. Risso, D., Perraudeau, F., Gribkova, S., Dudoiit, S., Vert, J.-P.: A general and flexible method for signal extraction from single-cell rna-seq data. Nature communications 9(1), 1–17 (2018)

24. Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A.J., Burdziak, C., Moon, K.R., Chaffer, C.L., Pattabiraman, D., et al.: Recovering gene interactions from single-cell data using data diffusion. Cell 174(3), 716–729 (2018)

25. Kiselev, V.Y., Andrews, T.S., Hemberg, M.: Challenges in unsupervised clustering of single-cell rna-seq data. Nature Reviews Genetics 20(5), 273–282 (2019)

26. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)

27. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)

28. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241 (2015). Springer

29. Duvenaud, D.K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P.: Convolutional networks on graphs for learning molecular fingerprints. In: Advances in Neural Information Processing Systems, pp. 2224–2232 (2015)

30. Guimaraes, G.L., Sanchez-Lengeling, B., Outeiral, C., Farias, P.L.C., Aspuru-Guzik, A.: Objective-reinforced generative adversarial networks (organ) for sequence generation models. arXiv preprint arXiv:1705.10843 (2017)

31. Poplin, R., Chang, P.-C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P.T., et al.: A universal snp and small-indel variant caller using deep neural networks. Nature biotechnology 36(10), 983–987 (2018)

32. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008(10), 10008 (2008)

33. Zappia, L., Phipson, B., Oshlack, A.: Splatter: simulation of single-cell rna sequencing data. Genome biology 18(1), 174 (2017)
Figures

**Figure 1** Detailed structure of scCAE model. Schematic overview.

**Figure 2** Datasets generated by Splatter. (a) Simple clustering datasets were generated, method of Splatter was set as “groups”. (b) Complex clustering datasets were generated, method of Splatter was set as “paths”. (c) Datasets with dropouts were generated, the type of dropout was set as “experiment”.

**Figure 3** Parameter selection for scCAE and scCAE-nonImputer. Under the constraint of $a + b = 1$, find the appropriate parameters $a$ and $b$ on the cell clustering datasets (complex clustering datasets).
Figure 4 Comparison of simple datasets experiments (results below 90% are not shown in the figure). Compare different methods on the simple datasets, the number of clusters was 5, 10, 20, 30, and 50, respectively.

Figure 5 Comparison of complex datasets experiments. Compare different methods on the datasets with dropoutss, the number of clusters was 10 and 25, respectively.
Figure 6 Analysis results of datasets with different sparsity. (a) Compare different methods on the simple datasets, the sparsity of input matrix was 23%, 34%, 45%, 58%, 72%, respectively. (b) Visualization results of some methods, as the sparsity of input matrix was 58%, 72%.
Figure 7 Analysis results of dataset with CTCs. (a) Visualization analysis results of dataset with CTCs, as the red box shows, there are more than a dozen CTCs. (b) Violin plot shows expression distribution of marker genes for clusters.
Table 1 Table of results of simple clustering datasets.

| Clusters | Method          | ARI  | NMI  | SW   | DBI   |
|----------|-----------------|------|------|------|-------|
| 5        | scCAE           | 0.9976 | 0.9945 | 0.9956 | 6.4008 |
|          | scCAE-nonImputer | 0.9951 | 0.9687 | 0.9712 | 6.8198 |
|          | AE              | 0.9871 | 0.9765 | 0.9761 | 6.3953 |
|          | PCA             | 0.9987 | 0.9997 | 0.9979 | 6.4037 |
|          | scScope         | 0.9932 | 0.9866 | 0.9885 | 6.4276 |
| 10       | scCAE           | 0.9931 | 0.9874 | 0.9747 | 6.3905 |
|          | scCAE-nonImputer | 0.9932 | 0.9878 | 0.9738 | 6.3753 |
|          | AE              | 0.9802 | 0.9845 | 0.9698 | 6.3417 |
|          | PCA             | 0.9972 | 0.9960 | 0.9864 | 6.3905 |
|          | scScope         | 0.9931 | 0.9885 | 0.9726 | 6.4016 |
| 20       | scCAE           | 0.9903 | 0.9865 | 0.9366 | 5.9700 |
|          | scCAE-nonImputer | 0.9829 | 0.9323 | 0.6945 | 6.7836 |
|          | AE              | 0.9824 | 0.9803 | 0.9047 | 6.1111 |
|          | DCA             | 0.9939 | 0.9819 | 0.8900 | 6.9888 |
|          | PCA             | 0.9949 | 0.9973 | 0.9611 | 6.0857 |
|          | scScope         | 0.9867 | 0.9847 | 0.9242 | 6.0415 |
| 30       | scCAE           | 0.9888 | 0.9867 | 0.8822 | 5.7940 |
|          | scCAE-nonImputer | 0.9653 | 0.8983 | 0.4506 | 7.3511 |
|          | AE              | 0.9818 | 0.9832 | 0.8668 | 5.8714 |
|          | DCA             | 0.9268 | 0.9504 | 0.6706 | 1.0423 |
|          | PCA             | 0.9947 | 0.9981 | 0.9751 | 5.7991 |
|          | scScope         | 0.9864 | 0.9863 | 0.9072 | 5.7043 |
| 50       | scCAE           | 0.9632 | 0.9607 | 0.6059 | 6.5855 |
|          | scCAE-nonImputer | -     | -     | -     | -     |
|          | AE              | 0.9630 | 0.9769 | 0.6650 | 5.8220 |
|          | PCA             | 0.9769 | 0.9907 | 0.9144 | 6.5992 |
|          | scScope         | 0.9783 | 0.9838 | 0.7686 | 5.8045 |

Table 2 Table of results of complex clustering datasets.

| Clusters | Method          | ARI  | NMI  | SW   | DBI   |
|----------|-----------------|------|------|------|-------|
| 10       | scCAE           | 0.6883 | 0.8525 | 0.2665 | 4.8739 |
|          | scCAE-nonImputer | 0.6533 | 0.8367 | 0.0821 | 4.8293 |
|          | AE              | 0.5860 | 0.8128 | 0.0498 | 4.5712 |
|          | DCA             | 0.5611 | 0.8029 | 0.0479 | 4.9296 |
|          | PCA             | 0.6271 | 0.8335 | 0.1982 | 4.8254 |
|          | scScope         | 0.6130 | 0.8212 | 0.0401 | 4.5668 |
|          | SAUCIE          | 0.5507 | 0.7785 | -0.0594 | 5.1974 |
| 25       | scCAE           | 0.6964 | 0.8082 | 0.4789 | 5.1798 |
|          | scCAE-nonImputer | 0.6559 | 0.7950 | 0.4156 | 5.3215 |
|          | AE              | 0.6687 | 0.7956 | 0.3491 | 4.9497 |
|          | DCA             | 0.5757 | 0.7476 | 0.2311 | 1.1427 |
|          | PCA             | 0.6505 | 0.7982 | 0.3835 | 5.2827 |
|          | scScope         | 0.6387 | 0.8051 | 0.4147 | 5.0494 |
|          | SAUCIE          | 0.6028 | 0.7518 | 0.3040 | 4.9899 |

Table 3 Time analysis

|         | sample size | 10K | 50K | 100K | 500K | 1M   |
|---------|-------------|-----|-----|------|------|------|
| scCAE   | 0.1227      | 0.1420 | 0.1523 | 0.3577 | 0.5602 |
| AE      | 0.1273      | 0.1503 | 0.1747 | 0.4057 | 0.7443 |
| PCA     | 0.1011      | 0.1045 | 0.1136 | 0.1364 | 0.1409 |
| scScope | 0.1386      | 0.2102 | 0.2545 | 0.5136 | 0.9375 |
| DCA     | 0.1432      | 0.4693 | 0.7227 | 2.9023 | 4.9705 |
| scVI    | 0.2136      | 1.0364 | 1.2807 | 4.8080 | 8.2080 |

Table 4 Memory analysis

|                         | scCAE-nonImputer | scCAE | scScope | AE   |
|-------------------------|------------------|-------|---------|------|
| Total params            | 677,824          | 2,110,336 | 5,353,664 | 3,218,816 |
| Input size (MB)         | 0.01             | 0.01  | 0.01    | 0.01 |
| Params size (MB)        | 2.59             | 8.05  | 0.4057  | 0.1364 |
| Estimated total size (MB)| 4.38            | 10.12 | 20.69   | 20.69 |

Page 14 of 14