Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

scAAGA: Single cell data analysis framework using asymmetric autoencoder with gene attention

Rui Meng [a], Shuaidong Yin [a], Jianqiang Sun [b], Huan Hu [c,**], Qi Zhao [a,*]

[a] School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China
[b] School of Information Science and Engineering, Linyi University, Linyi, 276000, China
[c] Institute of Applied Genomics, Fuzhou University, Fuzhou, 350108, China

ARTICLE INFO

Keywords:
scRNA-seq
Deep learning
Gene attention
Data augmentation
COVID-19

ABSTRACT

In recent years, single-cell RNA sequencing (scRNA-seq) has emerged as a powerful technique for investigating cellular heterogeneity and structure. However, analyzing scRNA-seq data remains challenging, especially in the context of COVID-19 research. Single-cell clustering is a key step in analyzing scRNA-seq data, and deep learning methods have shown great potential in this area. In this work, we propose a novel scRNA-seq analysis framework called scAAGA. Specifically, we utilize an asymmetric autoencoder with a gene attention module to learn important gene features adaptively from scRNA-seq data, with the aim of improving the clustering effect. We apply scAAGA to COVID-19 peripheral blood mononuclear cell (PBMC) scRNA-seq data and compare its performance with state-of-the-art methods. Our results consistently demonstrate that scAAGA outperforms existing methods in terms of adjusted rand index (ARI), normalized mutual information (NMI), and adjusted mutual information (AMI) scores, achieving improvements ranging from 2.8% to 27.8% in NMI scores. Additionally, we discuss a data augmentation technology to expand the datasets and improve the accuracy of scAAGA. Overall, scAAGA presents a robust tool for scRNA-seq data analysis, enhancing the accuracy and reliability of clustering results in COVID-19 research.

1. Introduction

The analysis of single-cell RNA sequencing (scRNA-seq) data has gained increasing importance in understanding the heterogeneity and structural information in different cell types [1–3]. In recent years, scRNA-seq technology has made significant progress in the study of COVID-19 [4–7]. It offers researchers a higher resolution and more accurate method to study the impact of the virus on host cells [8–10]. This technology also helps track the initial cellular targets of infection, understand host antiviral response mechanisms [11–13], and identify potential therapeutic targets [14]. Nevertheless, due to the significant impact of the virus on the human body and the diverse cell type heterogeneity, processing and analyzing scRNA-seq data pose challenges. In this context, single-cell clustering plays a pivotal role in analyzing scRNA-seq data [15–17]. Single-cell clustering significantly contributes to single-cell analysis [18]. It can reveal heterogeneity and structural information in single-cell data [19–21], help us discover new or rare cell types [22]. Moreover, it enables researchers to explore the relationships and functions between cells [23–25]. However, traditional clustering methods have limited performance when dealing with high-dimensional data [26]. Deep learning methods demonstrate superior effectiveness in handling such data, given their capacity to handle numerous variables. They automatically learn data feature representations, enabling the discovery of patterns and structures in the data, while also capturing complex relationships at multiple levels. Therefore, it is imperative to develop a data analysis frameworks based on deep learning for more effective scRNA-seq data analysis.

In recent years, many kinds of research such as single-cell multi-omics data analysis [27–29], computational toxicology [30], miRNA-lncRNA interaction [31–33], and circRNA-disease associations prediction [34–36] have been carried out in bioinformatics. These studies have promoted the development of computational methods for single-cell clustering methods to a certain extent. The existing clustering algorithms for single-cell data can be divided into the following four main categories: distance-based methods, graph-based methods, supervised learning-based methods, and deep learning-based methods [37].

* Corresponding author.
** Corresponding author.
E-mail addresses: studentiz@live.com (H. Hu), zhaoqi@lnu.edu.cn (Q. Zhao).

https://doi.org/10.1016/j.compbiomed.2023.107414
Received 2 July 2023; Received in revised form 2 August 2023; Accepted 28 August 2023
Available online 30 August 2023

Distance-based methods in single-cell data clustering, such as SC3 [38], SIMLR [39], PcaReduce [40], rely on measuring the distance or similarity between cells for clustering. Although these methods exhibit fast convergence and improved clustering results, they suffer from a drawback of dependence on randomly selected initial cluster centers, which may hinder convergence to the global optimal solution. Graph-based methods represent scRNA-seq data as a graph structure. Spectral clustering is one of the commonly used methods in graph methods, such as SinNLRR [41], and SCE-NA [42]. These methods offer advantages in terms of speed and scalability and capture the topological structure and modularity features of the data. However, they have limitations in terms of sensitivity, interpretability, and scale selection. Supervised learning-based methods such as SCINA [43], CellAssign [44], scPred [45], guide the clustering process by using known labels or prior knowledge. The aforementioned methods employ various techniques, such as classifiers, regressors, and decision trees, to accomplish prediction and evaluation tasks. Nevertheless, they possess certain limitations in terms of dependence on label quality and challenges in transfer learning. Consequently, deep learning techniques have emerged as a promising resolution to address these drawbacks.

Deep learning-based methods learn the underlying features and representation of the data through neural network models. These methods typically utilize autoencoders, generative adversarial networks, and variational autoencoders. Among the commonly used deep learning methods are scDeepCluster [25], scziDesk [46]. ScDeepCluster is a deep learning algorithm for scRNA-seq cluster analysis. By leveraging the structure of autoencoders and clustering, features can be automatically derived for clustering single-cell RNA sequencing data. The structure of scDeepCluster mainly consists of two parts: an encoder and a clustering component. The encoder maps input data into a low dimensional space through multiple hidden layers, thereby achieving dimensionality reduction and feature extraction of the data. The clustering component divides cells into different clusters by clustering the encoded data. However, scDeepCluster implementation uses an autoencoder structure with identical input and output dimensions, which could limit the encoder's ability to compress and extract features. Therefore, it is necessary to design more efficient and flexible encoders to improve clustering performance. The scziDesk architecture comprises three key components: a feature extractor, a cluster module, and a loss function. The feature extractor is a symmetrical structures, the outputs are three sets of parameters, dropout rate, mean value and dispersion value, in ZINB modeling. The cluster module divides cells into different clusters using a hierarchical clustering algorithm. The loss function assesses the accuracy and stability of clustering results. Although scziDesk leverages a deep neural network model for scRNA-seq cluster analysis, the architecture's breadth and depth should be considered carefully to ensure optimal clustering accuracy and stability.

Based on the situation described above, we propose a more efficient and flexible framework named scAAGA. Our framework effectively extracts gene features from scRNA-seq data by leveraging a gene attention module, thereby enhancing clustering performance. Specifically, we use gene attention module to extract gene features from the expression matrix of each single cell. This process generates a gene scoring matrix, which is multiply with the original expression matrix to obtain a gene attention matrix. Subsequently, we utilize the features of the gene attention matrix to perform single-cell clustering. We extensively evaluate and validate scAAGA on single-cell datasets of COVID-19 PBMC. Additionally, we compare it with other state-of-art methods. The results demonstrate that scAAGA has advantages in identifying subpopulations of immune cells associated with COVID-19. Our study designs a novel approach for analyzing scRNA-seq data, improving single-cell clustering performance and provides useful information for understanding the pathogenesis of COVID-19.

2. Materials and methods

2.1. Datasets

In this work, we utilize single-cell datasets of COVID-19 obtained from Stephenson et al. [47]. These datasets comprise single-cell transcriptome data derived from peripheral blood mononuclear cells (PBMC) collected from individuals with varying degrees of COVID-19 severity, including asymptomatic, mild, moderate, severe, and critical cases. The samples are collected from three renowned centers in the UK: Newcastle, Cambridge, and London. In addition, these datasets also include a cohort of healthy volunteers for comparative analysis. We name all the datasets as COVID-19 PBMC, which comprise a total of 24737 genes analyzed in over 70,000 single cells. The Healthy PBMC dataset comprises 8,004 cells and 16 distinct subtypes. The Asymptomatic PBMC dataset includes 7,011 cells and 17 subtypes. The Mild PBMC dataset consists of 8,639 cells and 17 subtypes. The Moderate PBMC dataset contains 8,530 cells and 17 subtypes. The Severe PBMC dataset encompasses 7,128 cells and 18 subtypes. Finally, the Critical PBMC dataset involves 8,182 cells and 15 subtypes. The summary of COVID-19 PBMC is shown in Table 1.

2.2. Workflow of scAAGA

The workflow of scAAGA is depicted in Fig. 1. Generally, scAAGA is composed of three stages: data preprocessing, dimensionality reduction using gene attention module, and clustering module.

First, we preprocess the data by filtering out low quality cells, normalizing, and selecting highly variable genes resulting in a gene expression matrix after data augmentation. Second, we process the gene expression matrix using the gene attention module to produce a gene scoring matrix. Third, the gene attention matrix is obtained by multiplying the scoring matrix with the original expression matrix. In the end, we employ this attention matrix for dimensionality reduction and subsequent clustering, rather than the original expression matrix. Additional details can be found in the following subsections.

2.3. Data preprocessing

During the data preprocessing step, we obtain the single-cell gene expression matrix from the raw transcriptome sequencing data. Nevertheless, the matrix might encompass substantial noise which could impact the subsequent analyses, primarily due to technological constraints such as variations in library construction, limited sequencing depth, low capture rate, and batch effects. Consequently, it's essential to preprocess the gene expression matrix, eliminate low quality cells and batch effects before moving forward with data analysis [48–50].

The initial step of data preprocessing is quality control (QC), which involves the removal of low quality cells such as dead cells or cells with broken membranes or doublets. In this work, we define X as the count matrix of shape (N_{cells}, N_{genes}) for the scRNA-seq datasets, where N_{cells} is the number of cells and N_{genes} is the number of genes.

We consider cells that express fewer than 200 genes to be of low quality and exclude them from further analysis, because such cells may be damaged, dying, or have poor RNA quality, which can affect the

Table 1
The summary of COVID-19 PBMC datasets.

COVID-19 Datasets	Cells	Genes	Subtypes
Healthy PBMC	8004	24737	16
Asymptomatic PBMC	7011	24737	17
Mild PBMC	8639	24737	17
Moderate PBMC	8530	24737	17
Severe PBMC	7128	24737	18
Critical PBMC	8182	24737	15

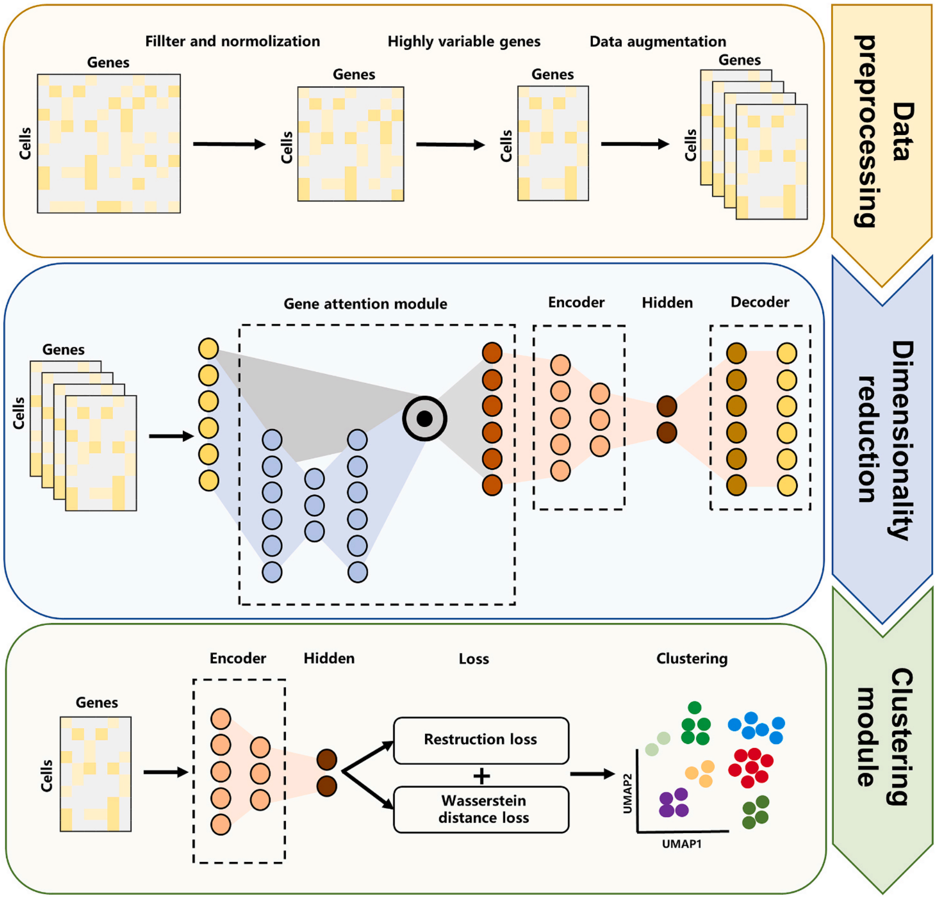

Fig. 1. Overview of the scAAGA. Our framework is composed of three steps: data preprocessing, dimensionality reduction, and clustering module.

accuracy of downstream analyses. Similarly, we filter out genes with expression detected in fewer than 3 cells, as they are considered to be lowly expressed or not expressed in most cells. This is because genes with low expression levels may not be biologically relevant or may be subject to higher levels of noise in the data, which can affect the reliability of downstream analyses. We apply the following formula to filter out these cells and genes:

$$X_{filter} = QC(X_{axis=1} > = 200 \cap X_{axis=0} > = 3) \tag{1}$$

The single cell transcriptome matrix reflects the number of mRNA molecules successfully captured, reverse transcribed and sequenced in each cell. However, due to the inherent variability of the individual operation steps, repeated sequencing of the same cell may yield different count depths. This leads to possible technical bias when comparing gene expression levels between different cells based on raw count data.

To eliminate this bias, we perform data normalization, which can be done by adjusting the count data to obtain comparable relative gene expression abundance between cells. This normalization ensures comparability across cells and genes, allowing for meaningful comparisons. In our study, we employ sequencing depth normalization, specifically using the counts per million (CPM) method to obtain the standardized data X_{norm}. CPM scales the counts in each cell (denoted as i) by a factor (denoted as f_i) such that the total counts in each cell sum to one million, this can be expressed mathematically as follows:

$$f_i = 10^6 / S_i \tag{2}$$

$$X_{norm} = X_{filter} / f \tag{3}$$

Where f is a vector that contains scaling factors f_i for every cell. The dimension of f is equal to N_{cells}. Similarly, the vector S, which represents the total counts of each cell, also has a dimension equal to N_{cells}.

After carrying out CPM, we apply a logarithmic conversion from X_{norm} to X_{log} as follow:

$$X_{log} = log(1 + X_{norm}) \tag{4}$$

Subsequently, we perform feature selection to identify genes that exhibit correlation, non-redundancy, and complementarity as features. These genes, known as highly variable genes (HVGs) [51], reflect intercellular variability, indicating greater variability in the data. A common approach for HVGs screening involves grouping genes based on their mean expression and selecting the gene with the highest variance-to-mean ratio within each group as the HVGs. Typically, the number of HVGs selected for downstream analysis ranges between 1000 and 5000, depending on the analysis objectives and dataset characteristics [52].

In this work, we choose 3000 HVGs, which effectively capture the majority of variability. This selection not only ensures the representativeness of our analysis but also reduces unnecessary redundancy in the information. Thus, we consider this to be an appropriate and effective strategy. After selecting HVGs, we obtain the processed data, denoted by X_{pre}, as shown in following formula:

$$X_{pre} = HVGs_{3000}(X_{log}) \tag{5}$$

Data augmentation is a widely used technique in deep learning that aims to expand the training datasets by performing a series of random

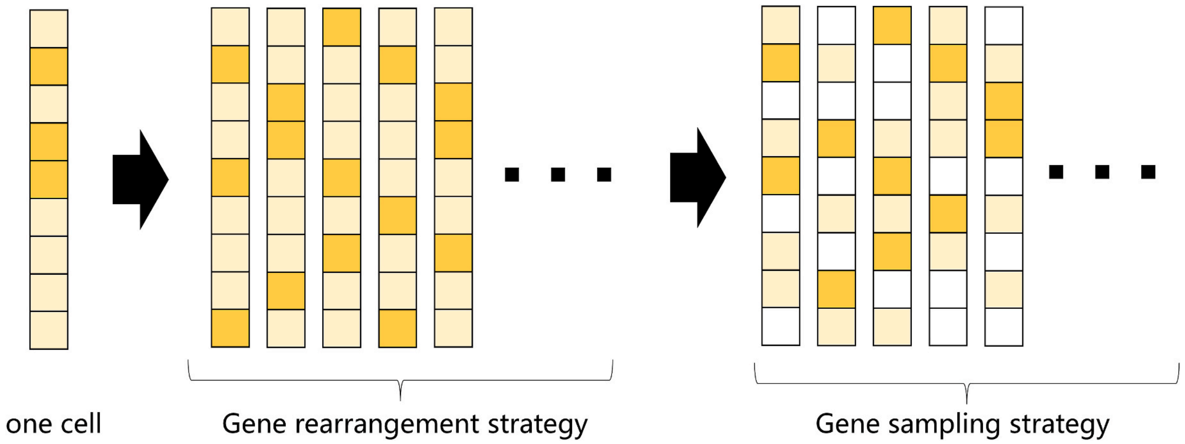

Fig. 2. Data augmentation is divided into two steps. First, we use the gene rearrangement strategy to expand the gene set. Next, we use the gene sampling strategy to randomly select genes. In this figure, each column represents a cell, and each block within a cell corresponds to a gene. We use light yellow blocks to indicate genes with zero expression values and dark yellow blocks to signify genes with non-zero expression data.

transformations on the training data [53–56]. This technique is simple and effective, as it increases the diversity and richness of the training data. Thereby, it can improve the generalization ability and robustness of deep learning models in practical applications.

In this subsection, we adopt data augmentation techniques inspired by image detection approaches to expand COVID-19 PBMC. Specifically, we refer to the treatment of the image datasets in masked autoencoders (MAE), which first divides the image uniformly into non-overlapping blocks and then randomly masks some of the pixels [57]. We can consider MAE as obtaining training samples by randomly sampling the unmasked portion of the image datasets, which using a uniform random sampling strategy without repetition.

We introduce a data augmentation technique for Xpre, as shown in Fig. 2. Our data augmentation process begins by rearranging the genes within each cell while preserving the gene type and the amount of expression (Eq. (6)). This gene rearrangement strategy alters the relative position of genes within a cell, while maintaining the distribution of genes within the cell unchanged.

$$W(i,j) = w(mod(i-1,N)+1, p(j)) \tag{6}$$

Here, *W(i,j)* indicates the element at row *i* and column *j* after rearranging. *Mod(i-1,N)* represents the remainder of *i-1* divided by *N*, which is used to determine the row number after rearranging. *p(j)* is the new ordinal of the column with index *j* after rearranging, ranging from 1 to *N*. Where the value of the expansion factor *N* is determined based on the characteristics of the datasets and the choice of model.

Subsequently, we perform a gene sampling strategy on the rearranged gene set, while maintaining a constant *N*-fold expansion. This strategy involves randomly sampling genes from each cellular gene set to construct a single augmented sample. We define X_{pre} after data augmentation as X_{aug}, as shown below:

$$X_{aug} = rearrange\left(W\left(X_{pr\ e_i}, X_{pr\ e_j}\right)\right) \tag{7}$$

Drawing from statistical principles, the distribution of randomly sampled individuals is commonly employed to represent the distribution of the entire group. By ensuring that the distribution of the samples remains unchanged, we effectively complete the expansion of COVID-19 PBMC. This preservation of the distribution significantly reduces the complexity for subsequent modeling tasks.

2.4. Dimensionality reduction

2.4.1. Autoencoder

Traditional autoencoders are widely used in current single-cell clustering methods due to their unique ability to reconstruct input data. An autoencoder is an unsupervised learning algorithm commonly employed for noise reduction and dimensionality reduction in single-cell transcriptome sequencing data.

An autoencoder must have two parts: an encoder and a decoder. The encoder can learn implicit data features and can decrease the input data's dimensionality to a smaller size than the input data, resulting in a low dimensional representation. The decoder can reconstruct the learned new features into the original input data, and the input and output data of a conventional self-encoder have the same scale. The detailed descriptions of encoder and decoder are as below:

$$z = f_\theta(x) = \sigma(Wx + b) \tag{8}$$

$$\hat{x} = g_{\theta'}(z) = \sigma(W'x + b') \tag{9}$$

In the equations above, x represents the input data, z indicates the potential space representation, $\hat{x}$ corresponds the generated input data, W and W' are the weight matrices of the encoder and decoder, b and b' are the bias vectors. The functions $f_\theta(x)$ and $g_{\theta'}(z)$ represent the encoder and decoder functions, while σ denotes the activation function. The training objective of an autoencoder is to minimize the reconstruction error $L(x, \hat{x})$, which quantifies the difference between the input data and the generated output data:

$$L(x, \hat{x}) = \|x - \hat{x}\|^2 \tag{10}$$

Here, $\|x - \hat{x}\|^2$ denotes the parametric number. By minimizing $L(x, \hat{x})$, the autoencoder can effectively learn useful features of the input data and also generate new data samples.

2.4.2. Asymmetric autoencoder with gene attention module

While traditional autoencoders are commonly used in deep learning clustering models, our research indicates that there is still room for improvement in existing methods for single-cell data clustering. Hence, we explore more sophisticated deep learning framework aiming to improve the accuracy and robustness of single-cell clustering.

In this subsection, we introduce an enhanced autoencoder architecture consisting of an asymmetric structure and a gene attention module. The proposed framework aims to achieve better performance. The decoder component is designed as a single layer to accurately reconstruct the input data. To improve the learning capacity of the encoder layer and enhance the decoding ability of the decoder layer, we increase the depth of the encoder layer. Additionally, we incorporate a gene attention module within the encoding part of the autoencoder to capture important gene features effectively. These improvements aim to optimize the performance of the autoencoder in capturing and representing gene expression patterns. The implementation process is as follows:

$$X_{input} = Input(X_{aug}) \tag{11}$$

$$X_{score} = Sigmoid(X_{input}) \tag{12}$$

$$X_{atten} = X_{score} {}^{*} X_{input} \tag{13}$$

$$X_{encoder} = Encoder(X_{attention}) \tag{14}$$

Initially, X_{aug} is input into scAAGA and denoted as X_{input} (Eq. (11)). Subsequently, a sigmoid layer of the same dimensions is applied to generate a gene score matrix X_{score} with completely identical dimensions (Eq. (12)). A dot product operation is then performed between X_{score} and X_{aug} to obtain the gene attention matrix $X_{attention}$ (Eq. (13)). $X_{attention}$ is further subjected to subsequent encoding operations (Eq. (14)). Throughout the training process, we incorporate a novel approach, leveraging the Wasserstein distance, to ensure the gene attention module attains a state of relative stability. This approach enables us to monitor, control the convergence of the module, guaranteeing its reliability and consistency during training. In particular, we record the attention matrix at each iteration of the training process and calculate F1-score of each gene. F1-score is computed based on the average gene attention value of the gene in a cell population, as well as the harmonic mean of the difference between this value and 1 minus the variance. The F1-score is computed as follows:

$$\mathrm{F1-score} = \frac{2(average(1 - variance))}{(average + (1 - variance))} \tag{15}$$

A high F1-score indicates both high expression levels and stability of the gene within the cell population. As F1-score values change dynamically during the training process, we use Wasserstein distance to measure the distributional changes in F1-score between adjacent epochs, enabling the identification of the optimal state of scAAGA. Wasserstein distance is a distance measure used to quantify the difference between two probability distributions. It provides a quantification of the discrepancy between the distributions, wherein a smaller distance signifies a greater similarity. The formal definition of Wasserstein distance is as follows:

$$W_{assersstein}(P_{F_t}, P_{F_{t-1}}) = \frac{inf}{\pi \in \Gamma(P_{F_t}, P_{F_{t-1}})} \int_{R \times R} |x - y| d\pi(x, y) \tag{16}$$

Here, P_{F_t} refers to the distribution of F1-score of each gene at epoch t, while $P_{F_{t-1}}$ represents the distribution of F1-score for each gene at the previous epoch. During the training process, the gradual decrease of Wasserstein distance indicates diminishing changes in the gene attention matrix, then reaching to a balanced state. The size of Wasserstein distance is continuously monitored in real-time. This calculated Wasserstein distance is incorporated into the loss function of scAAGA. Here, Wasserstein distance loss function is defined as the following formula:

$$L_1 = W_{asterstein}(P_{F_t}, P_{F_{t-1}}) \tag{17}$$

Next, we obtain low dimensional dense data, denoted as X_{latent}, from the hidden layers of scAAGA, which is used for subsequent clustering analysis:

$$X_{latent} = Linear(X_{encoder}) \tag{18}$$

Subsequently, we reconstruct the data by passing the low dimensional dense data through the decoder of scAAGA. In contrast to traditional autoencoders that typically adopt a symmetric encoding-decoding structure, we design the decoder of the autoencoder as a single layer consisting of a linear neural network, which enhances its powerful reconstruction capability. Using this single layer decoder, we reconstruct X_{latent} as $\widehat{X}$ which shown below:

$$\widehat{X} = Decoder(X_{latent}) \tag{19}$$

In accordance with the design objective of the autoencoder, which aims to minimize the reconstruction error and ensure that $\widehat{X}$ is as close as possible to X_{input}, the reconstruction loss function is formulated as follows:

$$L_2 = \frac{1}{n} \sum_{i=1}^{n} \left(\log(\widehat{X}) - \log(X_{input})\right)^2 \tag{20}$$

The total loss function is a fundamental component of the autoencoder's training process. Our objective is to minimize this loss function, which is comprised of two main components: the reconstruction loss(L_1) and Wasserstein distance loss(L_2). The total loss function L_{total} is:

$$L_{total} = L_1 + L_2 \tag{21}$$

More specifically, L_1 measures the discrepancy between the $\widehat{X}$ and X_{input}. L_2 is a measure of the distance between F1-score distributions of adjacent epochs during the training process of the gene attention module. L_{total} is used to ensure that the gene attention module reaches a relatively stable state and monitor the convergence of scAAGA during training. By minimizing L_{total}, we can train scAAGA to effectively encode the input data, and subsequently use the low dimensional dense data from the hidden layers data to clustering.

2.5. Clustering module

In order to enhance our understanding of the functionality, cell differentiation processes, and cell subtypes associated with COVID-19, we conduct clustering analysis on the single-cell data. The clustering analysis groups co-expressed genes and functionally similar cells together, enabling researchers to uncover valuable information. In this subsection, we utilize Leiden clustering algorithm for this purpose [58].

After passing the original data through scAAGA, we acquire a low dimensional, high-density representation called X_{latent}, which is then fed into the Leiden algorithm for clustering analysis. By using X_{aug} during the training phase, scAAGA is able to learn from a more diverse dataset, capturing complex patterns and relationships within the single-cell gene expression data. On the other hand, during the clustering application, the original gene expression matrix is used to ensure the clustering analysis reflected the inherent characteristics of the datasets without being influenced by the data augmentation techniques.

Leiden algorithm is a widely used and effective clustering algorithm specifically designed for scRNA-seq data analysis. It incorporates a modularity optimization approach, which identifies densely connected communities within the datasets. By maximizing the modularity score, this algorithm assigns cells to distinct clusters based on their gene expression patterns, enabling the identification of biologically relevant cell subtypes. Leiden algorithm is particularly well-suited for the single-cell datasets of COVID-19 PBMC due to its efficiency and scalability. Its ability to handle large-scale and complex subtype analysis makes it an effective tool for exploring the functionality, cell differentiation processes, and cell subtypes related to COVID-19. Moreover, Leiden algorithm excels in capturing the intricate structure of single-cell datasets, even those that are discrete and sparse, providing valuable insights through clustering analysis. In summary, Leiden algorithm is a powerful resource for studying single-cell data of COVID-19, contributing significantly to our understanding of this disease.

3. Results

3.1. Performance evaluation

Comparing the consistency of cell clustering results with published labels is a widely used method for evaluating the performance of clustering methods in scRNA-seq data analysis. In this subsection, three commonly used evaluation metrics are employed to assess clustering quality: adjusted rand index (ARI) [59], normalized mutual information (NMI) [60] and adjusted mutual information (AMI) [60].

(1) ARI can be utilized to measure the similarity between the clustering labels obtained from a clustering algorithm and the reference clustering labels. ARI is an extension of rand index (RI) [61], which is a measure of the agreement between two sets of labels. The formula of RI is as follows:

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} = \frac{\mathrm{TP} + \mathrm{TN}}{C_{\mathrm{n}}^{2}} \tag{22}$$

Here, TP represents the number of sample pairs that are assigned to the same cluster in both the true labels and clustering results. TN represents the number of sample pairs that are assigned to different clusters in both the true labels and clustering results. FP represents the number of sample pairs that are assigned to the same cluster in the clustering results but different clusters in the true labels, and FN represents the number of sample pairs that are assigned to different clusters in the clustering results but the same cluster in the true labels.

From the expression of RI, it can be seen that only TP and TN represent correct clustering results, while FP and FN indicate clustering results that do not match the actual labels. The value of RI is fixed between 0 and 1, with a higher RI score indicating better clustering performance. In Eq. (22), TP + TN + FP + FN represents all possible pairings of samples in the datasets, which can be calculated using the combination formula, denoted as C_n^2. Here, C represents a combination, n denotes the number of samples, and C_n^2 symbolizes the total number of all possible pairings of any two samples from n samples.

However, RI has a significant drawback in that it lacks sufficient penalty, which can result in low discriminative power and compromise the evaluation of clustering performance. Therefore, ARI is developed to improve upon the limitations of RI as follows:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathrm{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathrm{E}[\mathrm{RI}]} \tag{23}$$

Here, RI represents the number of correct clustering, is the expected value of RI, and max(RI) is the maximum value of RI, which corresponds to the case where the clustering is completely correct.

ARI ranges from −1 to 1, with a higher score indicating more accurate clustering performance. When ARI is equal to 1, it suggests that the clustering results perfectly match the ground truth or the true clustering assignment. In summary, ARI takes into account the expected value of RI and the maximum possible value of RI, resulting in a more accurate evaluation of clustering performance.

(2) NMI is a metric used in clustering analysis to measure the similarity between two clustering results. It is based on the mutual information between two sets of data, which measures the degree of consistency between the distributions of the two sets. NMI is an important indicator of information that relates to the correlation between two sets of events.

In the context of clustering analysis, NMI is defined as the mutual information between two different clustering results, divided by the entropy of the two clustering results. Specifically, given two different clustering results U and V for a given set of n data points, with clusters C_U and C_V, respectively. NMI between U and V is defined as:

$$\mathrm{NMI} = \frac{\sum_{p=1}^{C_U} \sum_{q=1}^{C_V} \left|U_p \cap V_q\right| \log \frac{U_p \cap V_q}{\left|U_p\right| \times \left|V_q\right|}}{\sqrt{\left(-\sum_{p=1}^{C_U} \left|U_p\right| \log \frac{\left|U_p\right|}{n}\right)\left(-\sum_{q=1}^{C_V} \left|V_q\right| \log \frac{\left|V_q\right|}{n}\right)}} \tag{24}$$

The value range of NMI is between 0 and 1. In this study, NMI is used to compare the clustering labels with the true labels. A higher NMI score indicates a closer match between the clustering labels and true labels, which indicates a better clustering performance.

(3) AMI is a normalized form of mutual information, similar to NMI, but it is adjusted when calculating mutual information to avoid the impact of random events and make the calculation results more accurate. AMI between U and V is defined as:

$$\mathrm{AMI} = \frac{\sum_{p=1}^{C_U} \sum_{q=1}^{C_V} \left|U_p \cap V_q\right| log \frac{U_p \cap V_q}{\left|U_p\right| \times \left|V_q\right|} - E\left(\sum_{p=1}^{C_U} \sum_{q=1}^{C_V} \left|U_p \cap V_q\right| log \frac{U_p \cap V_q}{\left|U_p\right| \times \left|V_q\right|}\right)}{max\left(-\sum_{p=1}^{C_U} \left|U_p\right| log \frac{\left|U_p\right|}{n}, -\sum_{q=1}^{C_V} \left|V_q\right| log \frac{\left|V_q\right|}{n}\right) - E\left(\sum_{p=1}^{C_U} \sum_{q=1}^{C_V} \left|U_p \cap V_q\right| log \frac{U_p \cap V_q}{\left|U_p\right| \times \left|V_q\right|}\right)} \tag{25}$$

Similar to NMI, the value of AMI ranges from 0 to 1, with larger values indicating greater similarity between the clustering results and the real categories.

3.2. Comparison with other methods

To evaluate the performance of scAAGA, we conduct comparative experiments with two state-of-the-art clustering models in the field of bioinformatics, specifically scDeepCluster and scziDesk. In order to enhance the persuasiveness of our experiments, we choose deep learning models that are based on autoencoders. All of the parameters utilized by scDeepCluster and scziDesk are remained at their default value or at the values their authors recommended.

To ensure the authenticity and accuracy of the comparison, we evaluate all three models on COVID-19 PBMC. As illustrated in Table 2, NMI scores reveal that scAAGA outperforms scDeepCluster by a remarkable percentage of 18.5% on the Healthy PBMC datasets and by a notable margin of 22.8% on Asymptomatic PBMC dataset. Moreover, scAAGA achieves a substantial superiority over scziDesk, with a significant difference of 26.9% in terms of NMI on the Healthy PBMC dataset and a relative improvement of 0.3% on the Asymptomatic PBMC dataset.

The trend of scAAGA's superiority over both scDeepCluster and scziDesk is also observed consistently across the remaining datasets. On the Mild PBMC, Moderate PBMC, Severe PBMC, and Critical PBMC datasets, scAAGA achieves higher NMI scores than scDeepCluster, with improvements of 8.7%, 41.9%, 24.9%, and 2.8%, respectively.

Table 2
The NMI of three clustering methods on different COVID-19 PBMC datasets.

Datasets	scAAGA	scDeepCluster	scziDesk
Healthy PBMC	**0.7395**	0.5543	0.4703
Asymptomatic PBMC	**0.6789**	0.4507	0.6415
Mild PBMC	**0.7267**	0.6637	0.5618
Moderate PBMC	**0.7026**	0.4955	0.6912
Severe PBMC	**0.7315**	0.5844	0.4468
Critical PBMC	**0.7289**	0.7107	0.5739

Table 3
The AMI of three clustering methods on different COVID-19 PBMC datasets.

Datasets	scAAGA	scDeepCluster	scziDesk
Healthy PBMC	**0.7189**	0.5289	0.4456
Asymptomatic PBMC	**0.6545**	0.4139	0.6178
Mild PBMC	**0.7023**	0.6589	0.5517
Moderate PBMC	**0.6729**	0.4467	0.6687
Severe PBMC	**0.7029**	0.5681	0.4134
Critical PBMC	**0.6877**	0.6652	0.5563

Table 4
The ARI of three clustering methods on different COVID-19 PBMC datasets.

Datasets	scAAGA	scDeepCluster	scziDesk
Healthy PBMC	**0.6658**	0.3964	0.2803
Asymptomatic PBMC	**0.6126**	0.4877	0.4915
Mild PBMC	**0.6563**	0.4385	0.4264
Moderate PBMC	**0.6354**	0.3851	0.4863
Severe PBMC	**0.5986**	0.4872	0.2964
Critical PBMC	**0.6152**	0.6029	0.3438

Similarly, scAAGA outperforms scziDesk on these datasets, with improvements of 26.9%, 12.1%, 39.3%, and 21.1%, respectively. As shown in Table 3, on the Healthy PBMC dataset, scAAGA achieves an AMI value of 0.7189, which is 19% and 27.3% higher than the values obtained by scDeepCluster and scziDesk, respectively. On the Mild PBMC dataset, scAAGA achieves an AMI value of 0.7023, which is 4.3% and 15.1% higher than the values obtained by scDeepCluster and scziDesk, respectively. From Table 4, we can see that on the Healthy PBMC dataset, scAAGA obtains an ARI score of 0.6658, while scDeepCluster and scziDesk obtain 0.3964 and 0.2803, respectively. On the Mild PBMC dataset, scAAGA obtains an ARI score of 0.6563, while scDeepCluster and scziDesk obtain 0.4385 and 0.4264, respectively. The more detailed results can be found in Tables 3 and 4. These findings suggest that scAAGA, utilizing a gene attention-based asymmetric autoencoder approach, is capable of capturing intricate patterns of gene interactions and feature expression, thereby enhancing clustering accuracy. In contrast, scDeepCluster and scziDesk may not fully exploit the interplay among genes in handling these datasets, resulting in lower clustering performance.

Table 5
Comparison analysis between scAAGA and its ablation experiments on Healthy PBMC dataset. scAAGA-OA is the version of scAAGA where we remove the gene attention module. scAAGA-S is the version of scAAGA where we utilize a symmetric autoencoder instead of a single-layer decoder structure.

Model	NMI	AMI	ARI
scAAGA	**0.7395**	**0.7189**	**0.6658**
scAAGA-OA	0.6578	0.6411	0.5563
scAAGA-S	0.5423	0.5039	0.3644

To further demonstrate the performance of scAAGA, we compare the 2D visualization of COVID-19 PBMC using UMAP across three methods. In Fig. 3, each data point denotes a cell, and different cells are labeled by different colors. The proximity of data points in the space indicates the similarity between corresponding cells. This figure shows that scAAGA tends to cluster cells that exhibit greater similarity in closer proximity compared to the other methods. Moreover, within each cluster, there are fewer cells from other clusters, which further confirms that our algorithm is more accurate. The 2D visualization using UMAP provides visual evidence supporting the superior performance of scAAGA in accurately capturing the underlying structure and clustering patterns of COVID-19 PBMC.

3.3. Ablation experiments

scAAGA has key components such as gene attention module and single layer decoder. To further validate the effectiveness of the gene attention module in extracting nonlinear features from scRNA-seq data and the reconstruction ability of the single-layer decoder, we conduct ablation studies using 5-fold CV on Healthy PBMC datasets, respectively. These ablation experiments evaluate the impact of individual components on the predictive performance of scAAGA. In these ablation experiments, we remove or change one module at a time while keeping the other parts unchanged. To ensure the accuracy of our results, we perform these experiments on the same data samples.

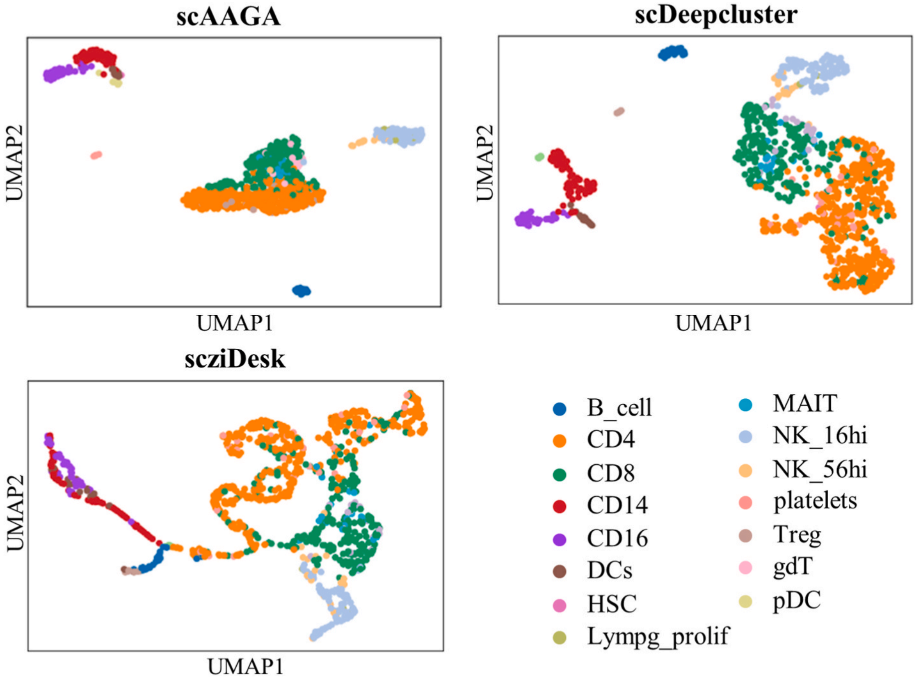

Fig. 3. The 2D visualization of UMAPs for clustering results of scAAGA, scDeepCluster and scziDesk on Healthy PBMC dataset.

Table 6
The evaluation metrics of scAAGA by different data augmentation factor on Healthy PBMC dataset.

Evolution	N = 1	N = 10	N = 20	N = 30	N = 40	N = 50	N = 60	N = 70
NMI	0.7020	**0.7395**	0.7232	0.7310	0.7245	0.7123	0.7069	0.7038
AMI	0.6980	**0.7365**	0.7200	0.7286	0.7213	0.7094	0.7045	0.7016
ARI	0.6034	**0.6658**	0.6407	0.6559	0.6453	0.6016	0.5943	0.5844

- scAAGA-OA: we remove the gene attention module in scAAGA. At this point, Eq. (1) becomes Eq. (24). Following its removal, we no longer utilize the gene attention matrix, instead use the original expression matrix as input for encoding. Additionally, L_2 is not include in the L_{total}.

$$X_{encoder} = Encoder\left(X_{\text{input}}\right) \tag{26}$$

- scAAGA-S: we utilize a symmetric autoencoder instead of using a single-layer decoder structure.

Table 5 shows the cluster evaluation results of the ablation experiment, comparing scAAGA and other two variants on Healthy PBMC dataset. We can observe that scAAGA has better performance than scAAGA-OA and scAAGA-S. scAAGA achieves the highest NMI score of 0.7395, which is 8.2% and 19.7% higher than that of scAAGA-OA and scAAGA-S, respectively. Additionally, scAAGA outperforms the other methods in terms of AMI score, achieving a score of 0.7186, which is 7.7% and 21.4% higher than that of scAAGA-OA and scAAGA-S, respectively. Moreover, scAAGA obtains the best ARI score of 0.6658, which is 16.2% and 35.4% higher than that of scAAGA-OA and scAAGA-S, respectively. These results suggest that the proposed methods in our design can significantly improve the predictive performance of the clustering algorithm.

3.4. Data augmentation factor

We also investigate the impact of different data augmentation factors on the experimental results. Increasing the data augmentation factor can increase the diversity and quantity of training data, thereby improving the generalization ability and performance of scAAGA. However, using an excessively high data augmentation factor may lead to overfitting and compromise the performance of scAAGA. Therefore, selecting an appropriate data augmentation factor requires careful consideration and adjustment based on specific experimental needs and datasets characteristics.

To determine the optimal data augmentation factor, we conduct experiments using Healthy PBMC dataset. The results of the data augmentation factor selection are presented in Table 6. It is evident that the clustering performance is optimal when the dataset is expanded by a factor of 10.

4. Discussion and conclusion

ScRNA-seq has emerged as a pivotal tool for understanding cell heterogeneity and structure. In the field of COVID-19 research, scRNA-seq technology plays a crucial role in studying the single-cell-level structure and function of life. However, preprocessing and analysis of scRNA-seq data remain challenging due to the virus's varying effects on different cell types and the resulting heterogeneity. To address these challenges, single-cell clustering methods have become a pivotal step in analyzing scRNA-seq data. In this work, we propose scAAGA, a single cell data analysis framework based on an asymmetric autoencoder with a gene attention module. We apply scAAGA to scRNA-seq data analysis, specifically focusing on COVID-19 single-cell gene expression data. The results demonstrate that scAAGA consistently outperforms scDeepCluster and scziDesk in terms of ARI, NMI, and AMI scores. The improvements range from 2.8% to 27.8% in NMI scores, indicating the superior clustering performance of scAAGA. The 2D visualization using UMAP further supports the superior performance of scAAGA. Ablation experiments are conducted to validate the effectiveness of scAAGA's gene attention module and single-layer decoder. The results show that removing the gene attention module or using a symmetric autoencoder (scAAGA-OA and scAAGA-S) results in lower clustering performance compared to scAAGA. Furthermore, the impact of different data augmentation factors on the clustering performance is investigated. The results indicate that expanding the datasets by a factor of 10 yielded optimal clustering performance for scAAGA.

The good performance of scAAGA can be attributed to several key factors. First, we propose a data augmentation method based on MAE. This method increases the diversity and richness of the training data, improving the generalization ability and robustness of the deep learning models. Furthermore, we introduce a gene attention module within the encoding part of scAAGA, which enables our model to capture important gene interactions and extract nonlinear features from scRNA-seq data. Notably, we also incorporate a novel approach, leveraging Wasserstein distance, to ensure the gene attention module attains a state of relative stability. At last, the utilization of a single-layer decoder further enhances the reconstruction capability of the autoencoder. Overall, these key factors show advantages in identifying subpopulations of immune cells associated with COVID-19, providing an approach for the analysis of scRNA-seq data and useful information for understanding the pathogenesis of COVID-19.

However, scAAGA still has a few limitations and potential challenges. First, this framework may be sensitive to variations in experimental protocols, cell types, and disease conditions, which could impact its performance in different scenarios. Second, the training process can be time-consuming, particularly when dealing with larger datasets. This limitation may restrict the accessibility and scalability of scAAGA for researchers with limited computational resources. Third, it is crucial to continue benchmarking scAAGA against other state-of-the-art clustering algorithms as they emerge. The field of scRNA-seq data analysis is rapidly evolving, and new methods may provide novel insights or outperform existing approaches. Regular benchmarking will ensure that scAAGA remains competitive and validates its continued use. Future work could involve further optimization of scAAGA, exploration of additional evaluation metrics, and application of the proposed method to other scRNA-seq datasets beyond COVID-19. The continued development of advanced clustering methods will contribute to the field of bioinformatics and provide valuable insights into complex biological systems. In addition, the gene scoring matrix and gene attention matrix are potentially useful in identifying genes with high scores, which can help gain insights into the significance and functions of these cell populations. Our future research endeavors will focus on exploring the effects of such genes on the cell populations.

Funding

This study was supported by National Natural Science Foundation of China (Grant No. 11805091), Foundation of Education Department of Liaoning Province (Grant No. LJKZ0280), Natural Science Foundation of Liaoning Province (Grant No. 2023-MS-288).

Data availability

The codes and datasets are available online at https://github.com/zhaoqi106/scAAGA.

Declaration of competing interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

List of abbreviations

scRNA-seq	single-cell RNA sequencing
PBMC	peripheral blood mononuclear cells
QC	quality control
CPM	counts per million
HVGs	highly variable genes
NMI	normalized mutual information
AMI	adjusted mutual information
ARI	adjusted rand index
UMAP	uniform manifold approximation and projection
5-fold CV	5-fold cross-validation
MAE	masked autoencoders

References

[1] B. Hwang, J.H. Lee, D. Bang, Single-cell RNA sequencing technologies and bioinformatics pipelines, Exp. Mol. Med. 50 (2018) 1–14.

[2] H. Zhang, M. Lu, G. Lin, et al., SoCube: an innovative end-to-end doublet detection algorithm for analyzing scRNA-seq data, Briefings Bioinf. 24 (2023) bbad104.

[3] M.D. Luecken, F.J. Theis, Current best practices in single-cell RNA-seq analysis: a tutorial, Mol. Syst. Biol. 15 (2019), e8746.

[4] F. Tang, C. Barbacioru, Y. Wang, et al., mRNA-Seq whole-transcriptome analysis of a single cell, Nat. Methods 6 (2009) 377–382.

[5] S. Zhang, K. Amahong, C. Zhang, et al., RNA-RNA interactions between SARS-CoV-2 and host benefit viral development and evolution during COVID-19 infection, Briefings Bioinf. 23 (2021) bbab397.

[6] X. Li, P. Zhang, Z. Yin, et al., Caspase-1 and gasdermin D afford the optimal targets with distinct switching strategies in NLRP1b inflammasome-induced cell death, Research 2022 (2022), 9838341.

[7] E. Papalexi, R.J.N.R.I. Satija, Single-cell RNA sequencing to explore immune cell heterogeneity, Nat. Rev. Immunol. 18 (2018) 35–45.

[8] X. Ren, W. Wen, X. Fan, et al., COVID-19 immune features revealed by a large-scale single-cell transcriptome atlas, Cell 184 (2021) 1895–1913.

[9] S. Zhang, K. Amahong, X. Sun, et al., The miRNA: a small but powerful RNA for COVID-19, Briefings Bioinf. 22 (2021) 1137–1149.

[10] B.J. Meckiff, C. Ramirez-Suastegui, V. Fajardo, et al., Imbalance of regulatory and cytotoxic SARS-CoV-2-reactive CD4(+) T cells in COVID-19, Cell 183 (2020) 1340–1353.

[11] M. Liao, Y. Liu, J. Yuan, et al., Single-cell landscape of bronchoalveolar immune cells in patients with COVID-19, Nat. Med. 26 (2020) 842–844.

[12] S. Lin, Y. Wang, L. Zhang, et al., MDF-SA-DDI: predicting drug–drug interaction events based on multi-source drug fusion, multi-source feature fusion and transformer self-attention mechanism, Briefings Bioinf. 23 (2021) bbab421.

[13] J. Guo, X. Wei, Q. Li, et al., Single-cell RNA analysis on ACE2 expression provides insights into SARS-CoV-2 potential entry into the bloodstream and heart injury, J. Cell. Physiol. 235 (2020) 9884–9894.

[14] A. Lopez-Cortes, P. Guevara-Ramirez, N.C. Kyriakidis, et al., In silico analyses of immune system protein interactome network, single-cell RNA sequencing of human tissues, and artificial neural networks reveal potential therapeutic targets for drug repurposing against COVID-19, Front. Pharmacol. 12 (2021), 598925.

[15] L. Yu, Y. Cao, J.Y.H. Yang, et al., Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data, Genome Biol. 23 (2022) 49.

[16] Q. Yang, B. Li, P. Wang, et al., LargeMetabo: an out-of-the-box tool for processing and analyzing large-scale metabolomic data, Briefings Bioinf. 23 (2022) bbac455.

[17] Q. Ding, W. Yang, M. Luo, et al., CBLRR: a cauchy-based bounded constraint low-rank representation method to cluster single-cell RNA-seq data, Briefings Bioinf. 23 (2022), bbac300.

[18] M.R. Karim, O. Beyan, A. Zappa, et al., Deep learning-based clustering approaches for bioinformatics, Briefings Bioinf. 22 (2021) 393–415.

[19] D. Usoskin, A. Furlan, S. Islam, et al., Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing, Nat. Neurosci. 18 (2015) 145–153.

[20] S. Zhang, X. Sun, M. Mou, et al., REGLIV: molecular regulation data of diverse living systems facilitating current multiomics research, Comput. Biol. Med. 148 (2022), 105825.

[21] M. Crow, A. Paul, S. Ballouz, et al., Characterizing the replicability of cell types defined by single cell RNA-sequencing data using MetaNeighbor, Nat. Commun. 9 (2018) 884.

[22] D. Grün, A. Lyubimova, L. Kester, et al., Single-cell messenger RNA sequencing reveals rare intestinal cell types, Nature 525 (2015) 251–255.

[23] J.A. Farrell, Y. Wang, S.J. Riesenfeld, et al., Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis, Science 360 (2018) eaar3131.

[24] J. Fu, Q. Yang, Y. Luo, et al., Label-free proteome quantification and evaluation, Briefings Bioinf. 24 (2023) bbac477.

[25] G. Eraslan, L.M. Simon, M. Mircea, et al., Single-cell RNA-seq denoising using a deep count autoencoder, Nat. Commun. 10 (2019) 390.

[26] H. Hu, Z. Li, X. Li, et al., ScCAEs: deep clustering of single-cell RNA-seq via convolutional autoencoder embedding and soft K-means, Briefings Bioinf. 23 (2022) bbab321.

[27] H. Hu, Z. Feng, H. Lin, et al., Gene function and cell surface protein association analysis based on single-cell multiomics data, Comput. Biol. Med. 157 (2023), 106733.

[28] Y. Chu, Y. Zhang, Q. Wang, et al., A transformer-based model to predict peptide–HLA class I binding and optimize mutated peptides for vaccine design, Nat. Mach. Intell. 4 (2022) 300–311.

[29] H. Hu, Z. Feng, H. Lin, et al., Modeling and analyzing single-cell multimodal data with deep parametric inference, Briefings Bioinf. 24 (2023), bbad005.

[30] T. Wang, J. Sun, Q.J.C.i.B. Zhao, et al., Investigating cardiotoxicity related with hERG channel blockers using molecular fingerprints and graph attention mechanism, Comput. Biol. Med. 153 (2023), 106464.

[31] H. Liu, G. Ren, H. Chen, et al., Predicting lncRNA–miRNA interactions based on logistic matrix factorization with neighborhood regularized, Knowl. Base Syst. 191 (2020), 105261.

[32] L. Zhang, P. Yang, H. Feng, et al., Using network distance analysis to predict lncRNA–miRNA interactions, Interdiscip Sci 13 (2021) 535–545.

[33] W. Wang, L. Zhang, J. Sun, et al., Predicting the potential human lncrna–mirna interactions based on graph convolution network with conditional random field, Briefings Bioinf. 23 (2022) bbac463.

[34] C.-C. Wang, C.-D. Han, Q. Zhao, et al., Circular RNAs and complex diseases: from experimental results to computational models, Briefings Bioinf. 22 (2021) bbab286.

[35] Q. Zhao, Y. Yang, G. Ren, et al., Integrating bipartite network projection and KATZ measure to identify novel CircRNA-disease associations, IEEE Trans. NanoBioscience 18 (2019) 578–584.

[36] E. Ge, Y. Yang, M. Gang, et al., Predicting human disease-associated circRNAs based on locality-constrained linear coding, Genomics 112 (2020) 1335–1342.

[37] R. Li, J. Guan, S. Zhou, Single-cell RNA-seq data clustering: a survey with performance comparison study, J. Bioinf. Comput. Biol. 18 (2020), 2040005.

[38] V.Y. Kiselev, K. Kirschner, M.T. Schaub, et al., SC3: consensus clustering of single-cell RNA-seq data, Nat. Methods 14 (2017) 483–486.

[39] B. Wang, J. Zhu, E. Pierson, et al., Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning, Nat. Methods 14 (2017) 414–416.

[40] J. Žurauskienė, C.J.B.b. Yau, pcaReduce: hierarchical clustering of single cell transcriptional profiles, BMC Bioinf. 17 (2016) 1–11.

[41] S. Islam, U. Kjällquist, A. Moliner, et al., Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq, Genome Res. 21 (2011) 1160–1167.

[42] E.Z. Macosko, A. Basu, R. Satija, et al., Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets, Cell 161 (2015) 1202–1214.

[43] Z. Zhang, D. Luo, X. Zhong, et al., SCINA: a semi-supervised subtyping algorithm of single cells and bulk samples, Genes 10 (2019) 531.

[44] A.W. Zhang, C. O'Flanagan, E.A. Chavez, et al., Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling, Nat. Methods 16 (2019) 1007–1015.

[45] J. Alquicira-Hernandez, A. Sathe, H.P. Ji, et al., scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data, Genome Biol. 20 (2019) 1–17.

[46] L. Chen, W. Wang, Y. Zhai, et al., Deep soft K-means clustering with self-training for single-cell RNA sequence data, NAR Genom Bioinform 2 (2020) lqaa039.

[47] E. Stephenson, G. Reynolds, R.A. Botting, et al., Single-cell multi-omics analysis of the immune response in COVID-19, Nat. Med. 27 (2021) 904–916.

[48] P. Melsted, A.S. Booeshaghi, L. Liu, et al., Modular, efficient and constant-memory single-cell RNA-seq preprocessing, Nat. Biotechnol. 39 (2021) 813–818.

[49] Y. Chu, A.C. Kaushik, X. Wang, et al., DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features, Briefings Bioinf. 22 (2019) 451–462.

[50] S. Baek, I. Lee, Single-cell ATAC sequencing analysis: from data preprocessing to hypothesis generation, Comput. Struct. Biotechnol. J. 18 (2020) 1429–1439.

[51] S.H. Yip, P.C. Sham, J.J.B.i.b. Wang, Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data, Briefings Bioinf. 20 (2019) 1583–1589.

[52] A. Butler, P. Hoffman, P. Smibert, et al., Integrating single-cell transcriptomic data across different conditions, technologies, and species, Nat. Biotechnol. 36 (2018) 411–420.

[53] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (2019) 60.

[54] L. Taylor, G. Nitschke, Improving Deep Learning with Generic Data Augmentation, Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1542–1547.

[55] F. Xu, D. Miao, W. Li, et al., Specificity and competition of mRNAs dominate droplet pattern in protein phase separation, Phys. Rev. Res. 5 (2023), 023159.

[56] A. Mikołajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in: 2018 International Interdisciplinary PhD Workshop, IIPhDW), 2018, pp. 117–122.

[57] K. He, X. Chen, S. Xie, et al., Masked autoencoders are scalable vision learners, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022) 16000–16009.

[58] V.A. Traag, L. Waltman, N.J.J.S.r. Van Eck, From Louvain to Leiden: guaranteeing well-connected communities, Sci. Rep. 9 (2019), 5233.

[59] D. Steinley, Properties of the hubert-arable adjusted rand index, Psychol. Methods 9 (2004) 386–396.

[60] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.

[61] V. Robert, Y. Vasseur, V.J.J.o.C. Brault, Comparing high-dimensional partitions with the co-clustering adjusted rand index, J. Classif. 38 (2021) 158–186.