scGMAI: a Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder

Article *in* Briefings in Bioinformatics · July 2021

DOI: 10.1093/bib/bbaa316

Briefings in Bioinformatics, 00(00), 2020, 1–10

doi: 10.1093/bib/bbaa316
Advance Access Publication Date: 00 Month 0000
Problem Solving Protocol

scGMAI: a Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder

Bin Yu, Chen Chen, Ren Qi, Ruiqing Zheng, Patrick J. Skillman-Lawrence, Xiaolin Wang, Anjun Ma and Haiming Gu

Corresponding author: Bin Yu, College of Mathematics and Physics, Qingdao University of Science and Technology, 99 Songling Rd, Laoshan District, Qingdao 266061, China, and School of Life Sciences, University of Science and Technology of China, 1129 Huizhou Ave, Baohe District, Hefei 230027, Chinaand School of Life Sciences, University of Science and Technology of China, 443 Huangshan Rd, Shushan District, Hefei 230027,China. E-mail: yubin@qust.edu.cn. The telephone number of the corresponding author: +86-532-88958923. E-mail: yubin@qust.edu.cn

Abstract

The rapid development of single-cell RNA sequencing (scRNA-Seq) technology provides strong technical support for accurate and efficient analyzing single-cell gene expression data. However, the analysis of scRNA-Seq is accompanied by many obstacles, including dropout events and the curse of dimensionality. Here, we propose the scGMAI, which is a new single-cell Gaussian mixture clustering method based on autoencoder networks and the fast independent component analysis (FastICA). Specifically, scGMAI utilizes autoencoder networks to reconstruct gene expression values from scRNA-Seq data and FastICA is used to reduce the dimensions of reconstructed data. The integration of these computational techniques in scGMAI leads to outperforming results compared to existing tools, including Seurat, in clustering cells from 17 public scRNA-Seq datasets. In summary, scGMAI is an effective tool for accurately clustering and identifying cell types from scRNA-Seq data and shows the great potential of its applicative power in scRNA-Seq data analysis. The source code is available at https://github.com/QUST-AIBBDRC/scGMAI/.

Key words: scRNA-Seq; autoencoder networks; fast independent component analysis; Gaussian mixture model; cell clustering

Bin Yu is an associate professor at the College of Mathematics and Physics, Qingdao University of Science and Technology, China. His research interests include bioinformatics, artificial intelligence and biomedical image processing.
Chen Chen is a master student at the College of Mathematics and Physics, Qingdao University of Science and Technology, China. His research interests are single-cell RNA-Seq data analysis and modeling.
Ren Qi is a PhD student at the College of Intelligence and Computing, Tianjin University, China. Her research interests are bioinformatics and machine learning.
Ruiqing Zheng is a PhD student at the School of Computer Science and Engineering, Central South University, China. His research interests are bioinformatics and machine learning.
Patrick J. Skillman-Lawrence is a PhD student at the College of Medicine, The Ohio State University, USA. His research interests are bioinformatics, artificial intelligence and biomedical image processing.
Xiaolin Wang is a master student at the College of Mathematics and Physics, Qingdao University of Science and Technology, China. His research interests are single-cell RNA-Seq data analysis and modeling.
Anjun Ma is a PhD student at the Department of Biomedical Informatics, The Ohio State University, USA. His research interests are bioinformatics, computational biology and machine learning.
Haiming Gu is a professor at the College of Mathematics and Physics, Qingdao University of Science and Technology, China. His research interests include artificial intelligence, machine learning and algorithms.

Submitted: 24 July 2020; **Received (in revised form):** 19 October 2020

Introduction

Bulk RNA sequencing is an important technique for transcriptome profiling. However, it only allows for the study of average cell-type-specific gene expression [1]. Next-generation single-cell RNA sequencing (scRNA-Seq) enables researchers to analyze both cellular and transcriptome heterogeneity at the single-cell resolution as it obtains RNA expression profiles from individual cells [2]. The ability to profile individual cells makes scRNA-Seq a powerful tool when researching complex diseases and cancer therapies [3]. The rapid development of scRNA-Seq technology provides strong technical support for accurate and efficient analyzing single-cell gene expression data. Researchers can obtain gene expression information of cells from the scRNA-Seq data and then identify cell types and cell subpopulations, which provides the possibility to study cellular heterogeneity at the single-cell level.

While scRNA-Seq presents many opportunities to advance research, it also generates significant challenges that hinder the downstream analysis of the data it produces [4]. As such, it is necessary to develop a clustering method that minimizes the impact of scRNA-Seq data and accurately distinguishing cell types. scRNA-Seq data are limited both by the myriad of dropout events it possesses, as well as the curse of dimension. The most significant limitation of scRNA-Seq data is a myriad of dropout events, in which a gene's expression is recorded as zero [5]. The sequencing technology cannot detect cell expression or expression levels that are actually low will produce dropout [6]. Researchers have proposed many imputation methods to overcome the negative burden this imposes on scRNA-Seq analysis. SAVER [7], for example, extracts information from genes and cells to produce an accurate gene expression profile. DCA [8] uses a deep counting autoencoder to impute missing gene expression data. Dijk *et al.* [9] created MAGIC to impute the missing values in scRNA-Seq data, reducing the impact of dropout. Both MAGIC and scImpute [10], another imputation tool, estimate gene expression values by projecting the feature's information between cells of the same type. These methods achieve significant results when processing scRNA-Seq data.

Another challenge encountered during scRNA-Seq analysis is the curse of dimensionality [11]. scRNA-Seq data often have high-dimensionality that poses a huge computational challenge for data analysis. Principal component analysis (PCA) [12] and manifold learning t-distributed stochastic neighbor embedding (t-SNE) [13] are the most commonly utilized methods for processing complex, high-dimensional data. Becht *et al.* [14] applied UMAP to quickly analyze scRNA-Seq data with a high level of reproducibility. ZIFA [15] uses a latent variable model derived via factor analysis in conjunction with an additional zero-inflation layer to adjust the model. What is more, ZIFA also automatically resolves dropout events by simulating missing features. Lin *et al.* [16] further improved analysis results with their use of neural networks to reduce the dimensionality of scRNA-Seq data.

Both dropout events and the curse of dimensionality directly reduces the accuracy of clustering methods, leading to misidentification of cell types [17]. To improve the accuracy of clustering results, Zheng *et al.* [18] created SinNLRR. This model adds low rank and non-negative structures to similarity matrices via spectral clustering. Wang *et al.* [19] constructed their SIMLR algorithm based on multi-kernel learning. SNN-Cliq [20] used clique detection, graph-based clustering method, to cluster scRNA-Seq data. CellTree [21] used document analysis techniques to generate tree structures that can describe the hierarchical relationship between samples. Ensemble clustering, which uses different scenarios to select the optimal result according to the agreement by the consensus functions [6], has also recently become popular. SC3 [22] is a fairly representative example of how ensemble clustering has been utilized. Another, Seurat [23] integrates scRNA-Seq data with *in situ* RNA patterns to infer cellular localization. This approach has been shown to correctly identify rare cell types.

While there are certainly many methods for analyzing scRNA-Seq data, some of the current clustering methods still have significant limitations and, as a result, cannot accurately cluster cells. These inaccuracies can arise from multiple sources. For example, some imputation methods are not suitable for scRNA-Seq data because they do not get the characteristics of the scRNA-Seq data [24]. Others are unable to overcome the negative impact of dropouts during their analysis of scRNA-Seq data, affecting their eventual results. Additionally, some dimensionality reduction methods failed to account for independent, interconnected features present in the data [25]. Finally, some clustering algorithms simply ignored the distribution of scRNA-Seq data, making these algorithms ill-suited for multimodal scRNA-Seq data. Any and all of these issues are prohibitive to accurately clustering scRNA-Seq data and as such, severely limit the potential of what a researcher can investigate.

Inspired by this, we hoped to further alleviate the impact of dropout events, capture low-dimensional spaces that represent the essential characteristics of scRNA-Seq data and accurately cluster cells and identify cell types. In this paper, we propose scGMAI, a novel Gaussian mixture method based on autoencoder networks and fast independent component analysis (FastICA) for scRNA-Seq data. scGMAI first learned the characteristics of scRNA-Seq data through autoencoder networks. This enabled it to effectively learn features of data and remove redundant information, improving its ability to handle dropout events present in scRNA-Seq data. It then employed FastICA to reduce the dimensionality of reconstruction data. Effective, independent data features were selected to improve the efficiency of scRNA-Seq data analysis. Then, a Gaussian mixture model automatically estimates the number of clusters in scRNA-Seq data. Moreover, scGMAI both revealed the internal structure of the scRNA-Seq data and accurately clustered cells and identified cell types. Finally, we visualized the clustering results with the t-SNE algorithm, which intuitively analyzes the cell distribution characteristics of scRNA-Seq data. scGMAI was tested on 17 publicly available scRNA-Seq datasets and compared with other state-of-the-art methods. Performance evaluation found that scGMAI outperformed each clustering method to which it was compared. This demonstrates that scGMAI is a powerful tool that can be implemented to analyze scRNA-data to accurately cluster cells and identify cell types.

Materials and methods

Overview of scGMAI pipeline

scGMAI is composed of (i) autoencoder networks to reconstruct scRNA-Seq data, (ii) an algorithm, FastICA, to reduce the dimensions of data and (iii) a Gaussian mixture model to cluster cells and identify cell types (Figure 1). This pipeline takes scRNA-Seq data as input. scGMAI enables the clustering of scRNA-Seq data comparable to existing clustering methods owing to its architectures of three valid models.

We first filtered for high abundance genes by removing genes whose expression value was 0 for more than 95% of

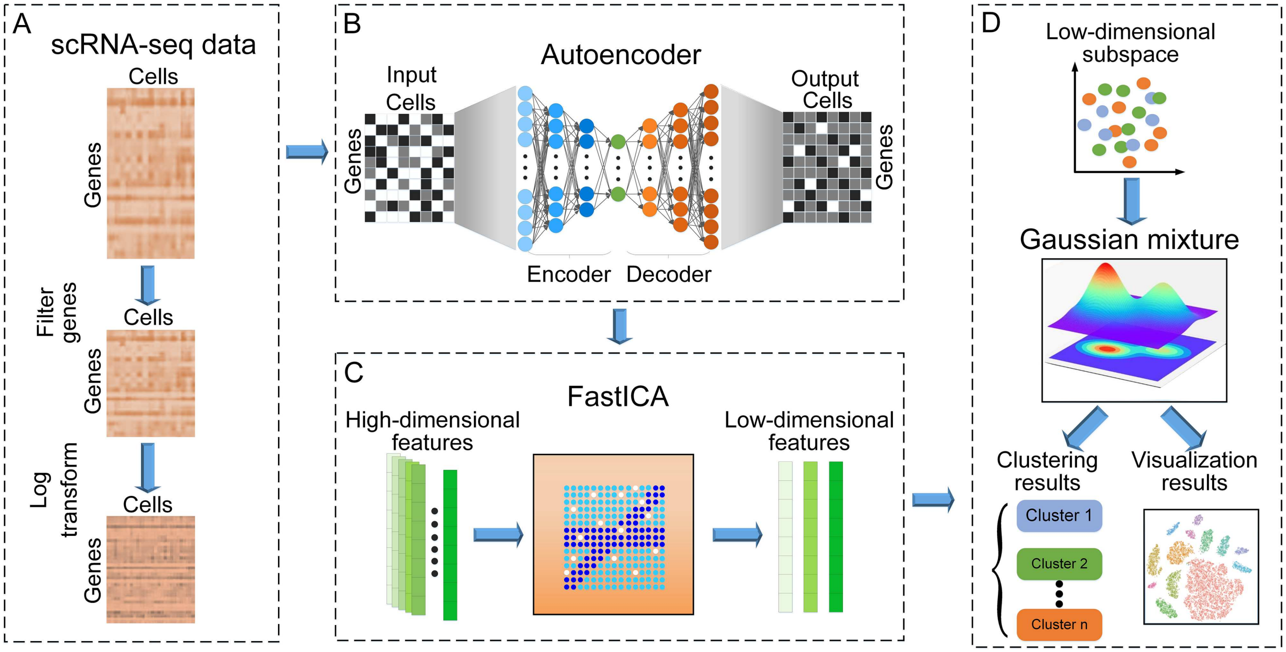

Figure 1. The pipeline of scGMAI. The model is divided into four processes: (**A**) Data preprocessing process. In each dataset, the columns represent cells and the rows represent genes. (**B**) Imputing process. It takes the log-transformation of scRNA-Seq expression matrix as input. The degree of color in the scRNA-Seq matrix represents the gene expression level. The imputation data are obtained after the encoding and decoding process of the autoencoder networks. (**C**) Dimensionality reduction process. The high-dimensional features are entered into FastICA, low-dimensional features are obtained after removing the redundancy and reducing dimension. (**D**) Clustering process. Gaussian mixture model clusters the cells to get the clustering results and identify the cell types. The clustering results are visualized by t-SNE.

cells (Figure 1A). Then, a log-transform was performed on the filtered data. To reduce the number of zero expression values caused by dropout events, we used an autoencoder model to reconstruct the processed data (Figure 1B). The white nodes represent zero expression values caused by dropout events or very small expression values; whereas, the grey and black nodes depict low and high expression values, respectively. The reconstruction data were then entered into FastICA (Figure 1C). A Gaussian mixture model cluster cells using low-dimensional features (Figure 1D). Additional details regarding the datasets, autoencoder networks, the dimensionality reduction algorithm, and the Gaussian mixture model that we employ in the present study can be found in the following subsections.

Datasets

We collected the data from 17 publicly available scRNA-Seq datasets containing gene expression values from the cells of various human and mouse tissue. The Ginhoux dataset was obtained from https://www.ncbi.nlm.nih.gov/geo/ under accession code GSE60783. Other datasets were obtained from https://hemberg-lab.github.io/scRNA.seq.datasets/. These datasets contain a number of cells (ranging from 56 to 14 437) and genes (ranging from 11 834 to 55 186). Detailed information regarding these datasets is shown in Supplementary Table S1.

Autoencoder networks

An autoencoder is an unsupervised, deep learning algorithm used to extract low-dimensional features [26]. They have been used to analyze scRNA-Seq data [27] and RNA-protein interactions [28]. Autoencoder networks can directly reduce the dimensionality of high-dimensional data. However, in cases where the original scRNA-Seq data contain a significant amount of redundant information and dropout events, autoencoder networks cannot effectively reduce dimensionality directly. When this happens, the autoencoder can employ a reconstruction process to efficiently reconstruct scRNA-Seq data with a high dropout rate.

Autoencoder networks include an input layer, hidden layers and an output layer [26]. Encoding is the process of mapping raw scRNA-Seq data $x \in R^m$ to an implicit representation $h(x) \in R^n$,

$$h(x) = \sigma_h\left(Wx + b\right) \tag{1}$$

where $W \in R^{n\times m}$ is the encoding weight matrix, $b \in R^n$ is the encoding offset vector and $\sigma_h(x)$ is the activation function.

Decoding is the process of mapping the implicit representation $h(x)$ to the output layer o and reconstructing the implicit representation $h(x) \in R^n$ into reconstruction data $y \in R^m$:

$$y = \sigma_o\left(W'h(x) + b'\right) \tag{2}$$

where W' $\in R^{m\times n}$ is the decoding weight matrix, $b' \in R^m$ is the decoding offset vector, and $\sigma_o(x)$ is the activation function.

The hidden layers are interconnected neural networks with a symmetrical structure. The output functions of each layer of the autoencoder are:

$$\begin{cases} h_1 = \sigma_{h_1}\left(W^1x + b^1\right) \\ h_k = \sigma_{h_k}\left(W^kh_{k-1} + b^k\right), k = 2, 3 \\ o = \sigma_o\left(W^4h_3 + b^4\right) \end{cases} \tag{3}$$

where W^1, W^k and W^4 are corresponding weight matrices, b^1, b^k and b^4 are corresponding offset vectors.

The loss function between the original data $x \in R^m$ and the reconstruction data $y \in R^m$ is:

$$L = \frac{1}{2}\sum \|y - x\|^2 \tag{4}$$

In order to make the reconstruction data closer to the original data, in the encoding and decoding process, autoencoder learns the non-linear feature relationship between cells and genes in scRNA-seq data, and it identifies and processes the outliers in the scRNA-seq data through multiple mapping while ensuring the minimum loss of reconstruction data. This process is achieved by continuously iterating the parameters.

The autoencoder takes the preprocessed expression matrix as an input of the input layer, then it compresses the matrix through three hidden layers with different mapping functions, and finally decompresses data through the output layer. Autoencoder can learn the characteristics of scRNA-Seq data and remove some redundant information through the decoding process. To optimize the performance of autoencoder by iterative parameters, we use three hidden layers and the number of nodes in the three layers are 800, 400 and 800, respectively. We choose the Softplus function for the activation function $\sigma_h(x)$:

$$\sigma_h(x) = \log\left(1 + e^x\right) \tag{5}$$

Compared to other activation functions, Softplus and ReLU more closely mimic the activation model of brain neurons. Additionally, Softplus functions can be regarded as smooth ReLU, which is more suitable for scRNA-Seq data [29]. The autoencoder networks process is shown in Supplementary Table S2.

FastICA

To mitigate the curse of dimensionality, we find it necessary to reduce the dimensions of our reconstruction data. Hyvarinen *et al.* [30] suggest ICA for processing high-dimensional data. ICAclust [31] applies ICA to temporal RNA-Seq analysis and ICA has even been used for magnetic resonance imaging analysis [32]. FastICA [30] separates complex data into independent data to find the largest independent component that contains important features and has certain sparsity. It has advantages over other ICAs due to its fast convergence, parallel computing and minimal storage required. More importantly, FastICA has a better effect on high-dimensional complex and locally independent data. It can reduce the dimensions of scRNA-Seq data that are both independent and related.

It is assumed that the scRNA-Seq reconstruction data x obeys the model $x = As$, where s is the unknown source data with independent components and A is an unknown mixed matrix. ICA is finding the best s and A by x. It first subtracts the mean vector $m = E\left\{x\right\}$ from the data x to center the data. Then, transform x to obtain a new whitening vector $\overline{x}$, and the components of $\overline{x}$ are independent and the covariance matrix is the identity matrix, that is:

$$E\left\{\overline{xx}^{T}\right\} = I \tag{6}$$

The process of whitening is usually achieved by eigenvalue decomposition of the covariance matrix:

$$E\left\{xx^{T}\right\} = EDE^{T} \tag{7}$$

where E is the characteristic matrix of the eigenvector of $E\left\{\overline{xx}^{T}\right\}$. D is a diagonal matrix of eigenvalues: $D = \mathrm{diag}\left(d_1, \cdots, d_n\right)$. So:

$$\overline{x} = ED^{-1/2}E^{T}x \tag{8}$$

$\overline{A}$ is converted into A by the whitening process at the same time, and $\overline{A}$ is an orthogonal matrix:

$$\overline{x} = ED^{-1/2}E^{T}As = \overline{A}s \tag{9}$$

After preprocessing, FastICA is completed by the following steps. First, we initialize vector w ($W = A^{-1}$,w is a row vector of W). Secondly, let $w^{+} = E\left\{xg\left(w^{T}x\right)\right\} - E\left\{g'\left(w^{T}x\right)\right\}w$ (g is a non-linear scalar function). And then, let $w = w^{+}/\left\|w^{+}\right\|$, if it does not converge, continue with step 2. Finally, several FastICA algorithms are used to estimate several independent components containing important information to achieve the purpose of dimensionality reduction.

We adjusted the parameters of FastICA algorithm. For the parameter n_components, which means the dimensions of the selected features, was set to be 10, and other parameters are default. The FastICA algorithm process is shown in Supplementary Table S3.

Gaussian mixture clustering

Clustering is an essential step in scRNA-Seq analysis as it divides all samples into different groups according to their attributes. Gaussian mixture model uses a probabilistic model to describe the prototype. Rau *et al.* [33] applied Gaussian mixture model to RNA-Seq co-expression analysis. The posterior probability more accurately simulates the data distribution than the prior probability. The Gaussian mixture model uses the EM algorithm [34] to account for the sensitivity to the initial cluster, noise and outliers, making the clustering result more robust. Therefore, it is feasible to use the Gaussian mixture model to cluster cells from complex scRNA-Seq data.

Denoted $p\left(x|\mu, \Sigma\right)$ as a probability density function:

$$p\left(x|\mu, \Sigma\right) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^{T}\Sigma^{-1}(x-\mu)} \tag{10}$$

assuming the Gaussian mixture distribution of data x:

$$p_M(x) = \sum_{i=1}^{k} \alpha_i \cdot p\left(x|\mu_i, \Sigma_i\right) \tag{11}$$

where k is the number of mixed components, and each component corresponds to a Gaussian distribution. μ_i and Σ_i are both parameters of the Gaussian mixture component, where μ_i is i-th N-dimensional mean vector, Σ_i is the i-th $n \times n$ covariance matrix, α_i is the corresponding ‘mixing coefficient’ of the probability of selecting the i-th mixed component, and $\sum_{i=1}^{k} \alpha_i = 1$.

The steps of Gaussian mixture clustering are as follows: algorithm first initializes the model parameters of the Gaussian mixture distribution, and then iteratively updates the model parameters based on the EM algorithm:

E-step: calculate the posterior probability γ_{ji} of the sample x_j generated by the i-th Gaussian mixture component based on

$$p_M\left(z_j = i|x_j\right) = \frac{P\left(z_j=i\right)\cdot p_M\left(x_j|z_j=i\right)}{p_M\left(x_j\right)} = \frac{\alpha_i\cdot p\left(x_j|\mu_i,\Sigma_i\right)}{\sum\limits_{l=1}^{k}\alpha_l\cdot p\left(x_j|\mu_l,\Sigma_l\right)} \tag{12}$$

M-step: iteratively update model parameters μ_i, Σ_i, α_i based on

$$\mu_i = \frac{\sum\limits_{j=1}^{m}\gamma_{ji}x_j}{\sum\limits_{j=1}^{m}\gamma_{ji}} \tag{13}$$

$$\Sigma_i = \frac{\sum\limits_{j=1}^{m}\gamma_{ji}\left(x_j-\mu_i\right)\left(x_j-\mu_i\right)^T}{\sum\limits_{j=1}^{m}\gamma_{ji}} \tag{14}$$

$$\alpha_i = \frac{1}{m}\sum_{j=1}^{m}\gamma_{ji} \tag{15}$$

Stop the iteration when the conditions (reached the maximum number of iteration rounds) are satisfied. If the conditions are not satisfied, continue to iteratively update the model parameters. Finally, the cluster label λ_j of the sample x_j is determined based on $\lambda_j = \underset{i\in\left\{1,2,\ldots,k\right\}}{\arg\max}\gamma_{ji}$.

To optimize the initialization step of the Gaussian mixture model, we introduce K-means ++ [35] to solve the problem of centroid initialization. K-means++ initializes the centroids to be far away from each other to optimize the initial steps, avoiding trapping in local minima. Firstly, K-means++ selects a center point C_1 randomly from the data X. Then, it selects the new center point C_i according to the probability $P = \frac{D(x)^2}{\sum_{x\in X} D(x)^2}$, where $D(x)$ is the shortest distance from a data point to the selected center point. Lastly, it continues to iterate this step until all the center points are selected. K-means++ can effectively optimize the initialization step.

To automatically estimate the number of clusters in scGMAI, we apply Bayesian information-theoretic criteria (BIC) [36] in the Gaussian mixture model. BIC can effectively select the number of components in scRNA-Seq data. The lower the BIC, the better the clustering performance.

$$\mathrm{BIC} = -2\log(L) + K\log(N) \tag{16}$$

where L is the likelihood value of the BIC model. K is the number of components, and N is the number of samples.

For the Gaussian mixture model, we use the Gaussian mixture function from the scikit-learn mixture module [37]. The parameters are kept at their default values. The number of clusters estimated by scGMAI is shown in Supplementary Table S1. The Gaussian mixture clustering process is shown in Supplementary Table S4.

Performance evaluation

To evaluate the performance of each clustering method, we selected four commonly used clustering evaluation indexes: normalized mutual information (NMI) [38], adjusted rand index (ARI) [39], homogeneity [40] and completeness [40]. The four indexes were calculated for each clustering method based on the prediction labels obtained and the real labels provided by the dataset. A detailed introduction of the four evaluation indexes is provided in Supplementary Si1.

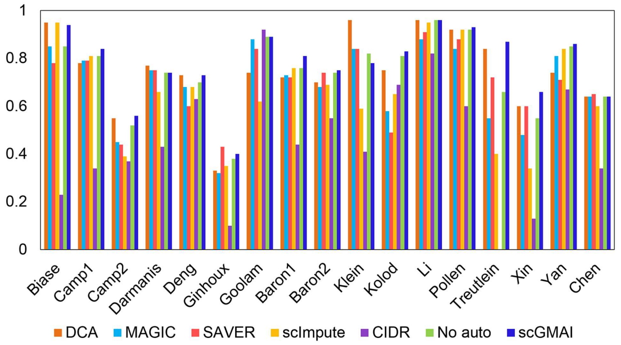

Figure 2. The NMI of the imputation methods on seventeen datasets. Kolod is the abbreviation of Kolodziejski dataset.

Results and discussion

The autoencoder networks in scGMAI can facilitate cell clustering better than other methods

We begin by replacing the autoencoder networks in scGMAI by four other imputation methods, i.e. SAVER [7], MAGIC [9], DCA [8], scImpute [10] and CIDR [41]. We also take NoAuto, which removed scGMAI's autoencoder networks, as one of the comparisons. Seventeen scRNA-Seq data and four indexes (NMI, ARI, Homogeneity and Completeness) are used to evaluate cell clustering performances. The results indicate the autoencoder networks significantly improved the clustering performance in 8 scRNA-Seq datasets. The NMI scores of above seven methods on 17 datasets are shown in Figure 2. The remaining results are available in Supplementary Figure S1.

First, to confirm the utility of reconstructing via autoencoder networks, we compared the version of scGMAI with and without the autoencoder networks. The NMI scores obtained by scGMAI with the autoencoder networks were greater than those without (Figure 2). It follows that the autoencoder networks were able to successfully reconstruct scRNA-Seq data. Additionally, scGMAI achieved the best performance for clustering cells from 11 datasets of any of the other methods assessed (Figure 2). For the remaining 6 scRNA-Seq datasets, scGMAI also achieved significant performance across four measurements over the other six methods. These results highlight the advantage of using scGMAI's autoencoder model versus the four other imputation methods we compared. The complete set of results can be found in Supplementary Table S5.

To further evaluate the performance of the autoencoder networks in scGMAI, we visualized the clustering results by t-SNE (Figure 3 and Supplementary Figure S2-S16). Results were obtained from seven imputation methods on 17 scRNA-Seq datasets. Each point represented a cell, besides, different cells were labeled by different methods and marked by different colors (Figure 3). For other imputation methods, the distances between different clusters were too short, resulting in many overlapping cells. However, scGMAI exhibited moderate distances between clusters, and it can clearly distinguish cell types. More importantly, we found NoAuto model nearly impossible to distinguish each cell type due to poor clustering on many outliers from the Kolodziejski and Klein dataset. However, in

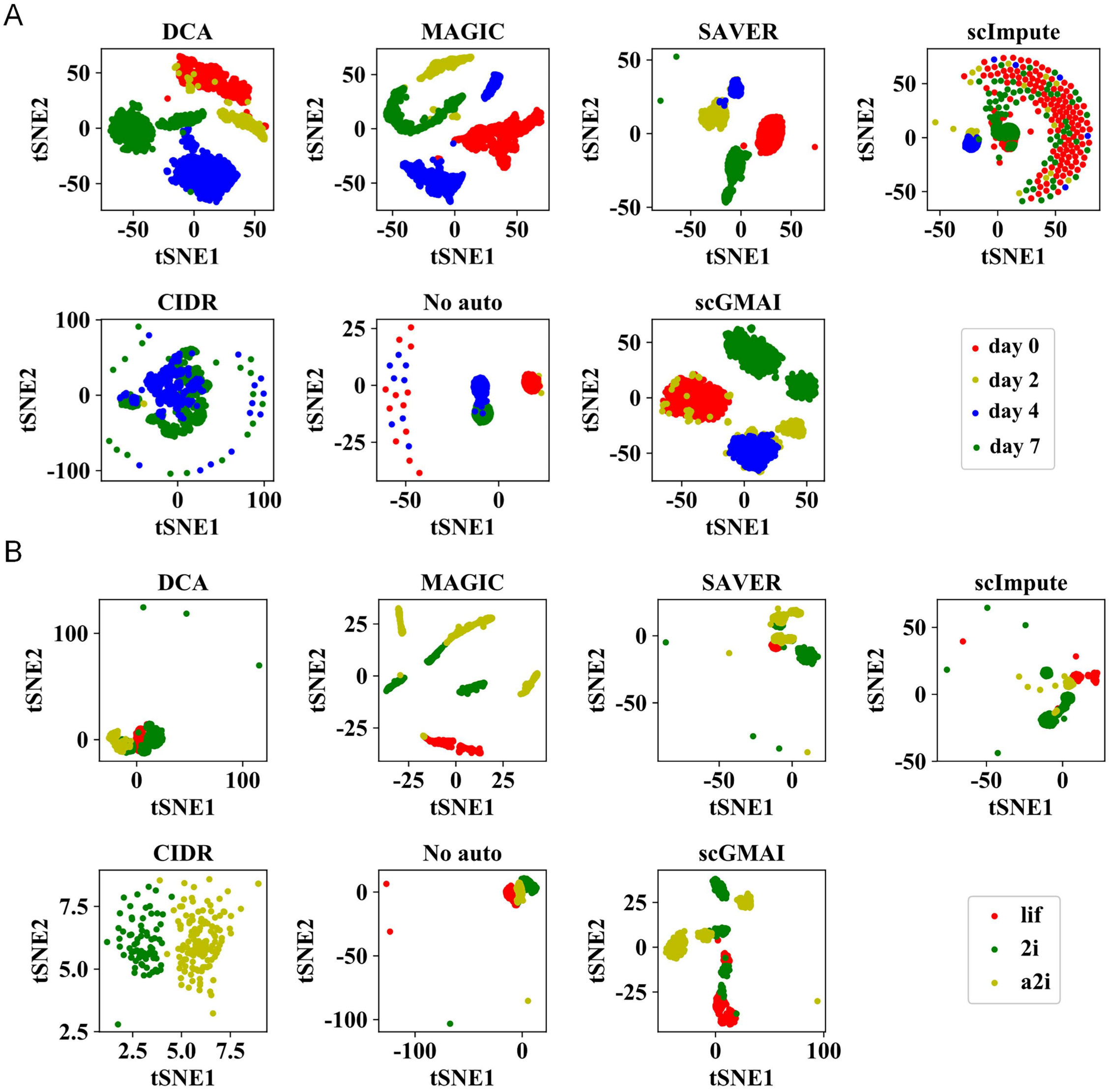

Figure 3. Visualization of imputation methods on two datasets. (**A**): Klein; (**B**): Kolodziejski.

the result of scGMAI, these outliers were well clustered into the corresponding cell types, and the remaining normal points still got better clustering results. In addition, we compared differentially expressed genes (DEGs) logFC scores using the original expression and the reconstructed expression identified in Day 7 cells from the Klein data (Supplementary Figure S17). As shown in Supplementary Figure S17, the DEGs signals are magnified after reconstruction. Although the scRNA-seq data contain a lot of dropout events, scGMAI can still identify cell types more accurately and amplify the DEGs signals than NoAuto. This is because the autoencoder learns the characteristics of scRNA-seq data and reconstructs the data to improve the clustering results of scRNA-seq data. The results showed that scGMAI improved visualization performance when analyzing each of the 17 scRNA-Seq datasets by reducing the noise levels.

To evaluate the performance of scGMAI's autoencoder in gene expression, we drew a gene expression heatmap of 17 scRNA-Seq data from the clustering results of scGMAI and NoAuto (Figure 4 and Supplementary Figure S18-S32). By comparing the gene expression heatmaps on Biase dataset, we noticed that the signature genes of Zygote and Blast were identical in the two models. Additionally, the signature genes in two cell types, i.e. 2-cell and 4-cell, of scGMAI model were far different from those of NoAuto (Figure 4B). Differences in signature genes yield inconsistent results of the two models. According to the heatmap, scGMAI performed better than NoAuto (Figure 4A). This is consistent with the fact that autoencoder reduces dropout events and increases expression values of the signature genes by reconstructing a scRNA-Seq expression matrix. Furthermore, the autoencoder makes it better to cluster cells and perform downstream analysis based on signature genes. The gene expression results of other datasets can be found in Supplementary Figure S18-S32.

According to the analysis of the above results, scGMAI's autoencoder networks increase the expression of the genes by reconstructing scRNA-Seq expression matrix. This makes it easier to identify signature genes, improves the accuracy of clustering and enhances the performance of visualizations.

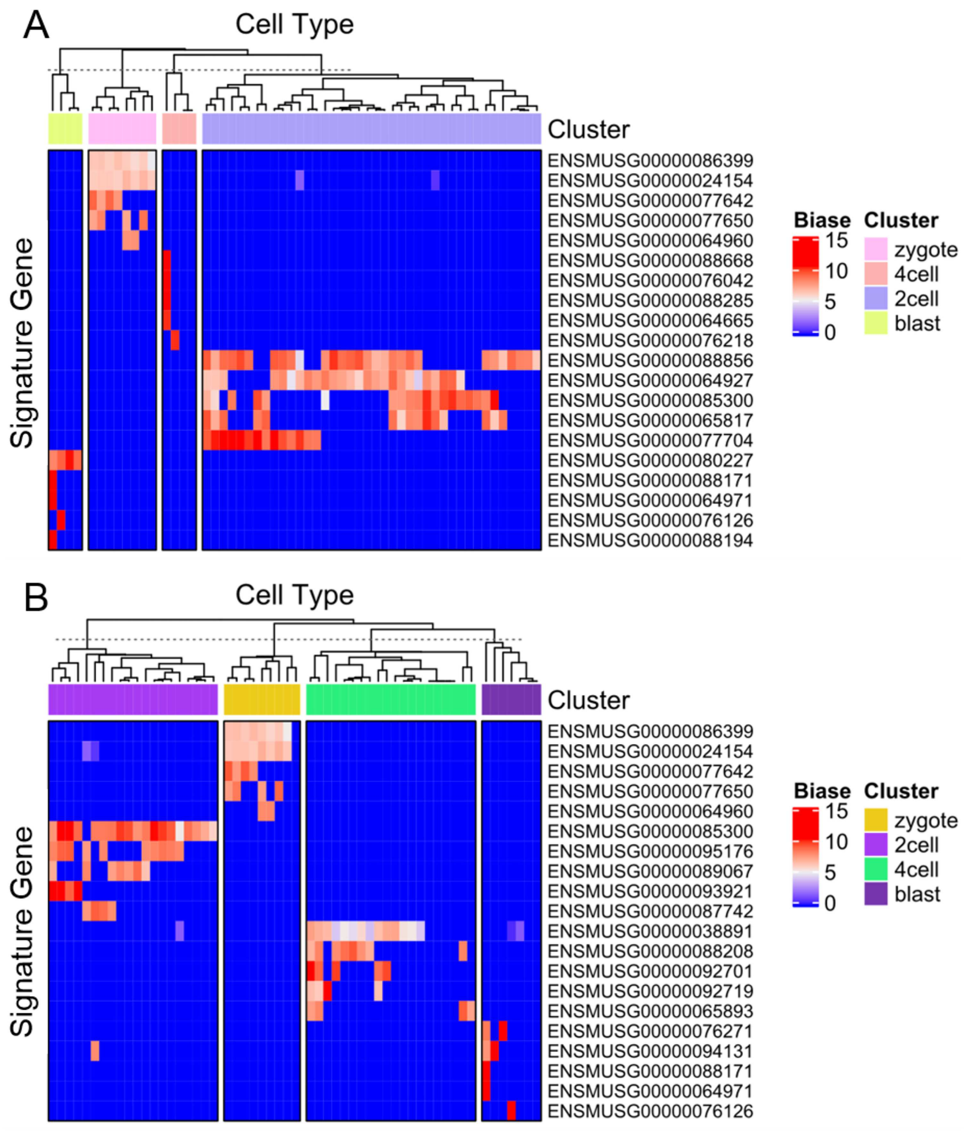

Figure 4. Heatmaps of the two models on Biase dataset: (**A**) scGMAI; (**B**) No auto. Each cluster displayed only TOP 5 signature genes.

FastICA in scGMAI is superior to other dimension-reduction algorithms

To investigate the importance of FastICA in scGMAI, we replaced it with PCA [12], t-SNE [13], UMAP [14], ZIFA [15], NMF [42] and SHARP [43]. Four evaluations (i.e. NMI, ARI, homogeneity and completeness) were used to compare the performance of seven dimensionality reduction algorithms using 17 datasets. We used the FastICA, PCA and t-SNE algorithm from the scikit-learn package; the other four tools were downloaded from reference papers. Additionally, parameters in UMAP and other parameters used in these three tools were default.

The performance of FastICA was significantly better than the other six dimensionality reduction methods across all four of the evaluation metrics (Table 1 and Supplementary Table S6). scGMAI's NMI score using FastICA was greater than all other methods on 14 datasets (Table 1). It also achieved comparable performance to the other models on Deng, Ginhoux and Pollen datasets, the remaining three datasets. This case didn't mean that FastICA's performance is completely superior to other dimensionality reduction methods, but FastICA is more suitable and effective for scGMAI to capture important, independent features from scRNA-Seq than other methods. The main idea of FastICA is to decompose a linear combination of several statistically independent components from a set of mixed data signals, and this combination contains more key information. Compared with PCA and other dimension-reduction methods, FastICA can better describe the random statistical characteristics of variables. Moreover, FastICA also has the advantages of better denoising effect, fast convergence and parallel computing. Therefore, FastICA has a better dimensionality reduction effect for scRNA-Seq data. It can also employ the features it identifies to more accurately cluster cells. In short, the dimensionality reduction performance of FastICA in scGMAI was better than other methods. The complete set of results can be found in Supplementary Table S6.

scGMAI accurately cluster cells from 17 scRNA-Seq datasets

To evaluate scGMAI's clustering performance, the following six benchmark clustering methods were used on each of the 17

Table 1. The NMI of the dimensionality reduction methods on 17 datasets

Datasets	PCA	t-SNE	UMAP	ZIFA	NMF	SHARP	FastICA
Biase	0.88	0.81	0.76	0.83	0.85	0.89	**0.94**
Camp1	0.73	0.75	0.79	**0.84**	0.37	0.76	**0.84**
Camp2	0.47	0.32	0.31	0.51	0.22	0.38	**0.56**
Darmanis	0.75	0.64	0.62	0.75	0.36	0.62	**0.76**
Deng	0.62	0.67	**0.76**	0.68	0.6	0.66	0.72
Ginhoux	0.41	0.33	0.39	0.4	**0.46**	0.38	0.41
Goolam	**0.89**	0.7	0.72	0.67	0.68	0.74	**0.89**
Baron1	0.76	0.71	0.72	0.48	0.51	**0.78**	**0.78**
Baron2	0.73	0.73	0.73	0.72	0.6	0.72	**0.75**
Klein	0.79	0.76	0.8	0.92	0.52	0.79	**0.95**
Kolodziejski	0.61	0.78	0.61	0.54	0.74	0.74	**0.83**
Li	0.92	0.89	0.88	0.9	0.53	0.91	**0.96**
Pollen	**0.93**	0.84	0.81	0.68	0.72	0.86	0.92
Treutlein	0.78	0.18	0.48	0.73	0.43	0.23	**0.82**
Xin	**0.61**	0.09	0.08	0.6	**0.61**	0.39	**0.61**
Yan	0.86	0.67	0.76	0.85	0.86	0.79	**0.87**
Chen	0.62	0.6	0.61	0.62	0.41	0.58	**0.65**

Note: Bold values mean the best results of all methods.

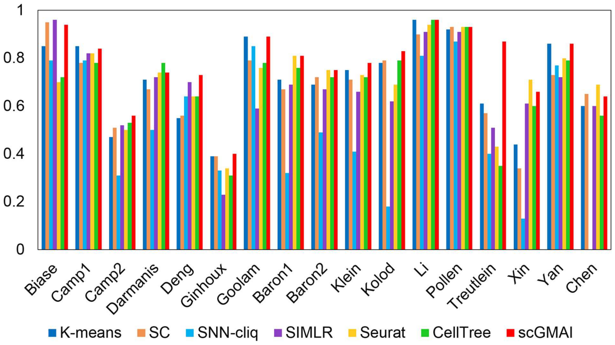

Figure 5. The NMI of the clustering methods on 17 datasets. Kolod is the abbreviation of Kolodziejski dataset. SNN-Cliq does not run on the Chen dataset because of the time complexity.

scRNA-Seq datasets: K-means clustering [44], spectral clustering (SC) [45], SIMLR [19], SNN-Cliq [20], Seurat [23], CellTree [21]. K-means clustering, spectral clustering and the Gaussian mixture model were functions obtained from the scikit-learn package. The n_components parameter in K-means and spectral clustering was set as the number of cell types that were provided by the dataset. All other parameters were kept at their default values. All of the parameters utilized by SIMLR, SNN-Cliq, Seurat and CellTree remained at their default value.

scGMAI performed better than the benchmark methods for all four of the evaluation metrics (Figure 5 and Supplementary Figure S33). The NMI of scGMAI was greater than or equal to the other six clustering methods for 12 of the 17 datasets (Figure 5). Additionally, scGMAI performed better than SNN-Cliq and SIMLR on all but the Biase dataset according to the NMI scores. The results showcase that scGMAI clusters cells with greater accuracy and precision. Moreover, on the homogeneity and completeness evaluations, scGMAI performs well (Supplementary Figure S33). Compared to other statistical-based clustering models, Gaussian mixture clustering uses posterior probability to more accurately simulate data distribution than prior probability. It more adeptly reveals the internal nature and regularity of scRNA-Seq data and cluster cells than previous methodologies. The complete set of results can be found in Supplementary Table S7.

Conclusion

We proposed a new Gaussian mixture clustering method based on autoencoder networks and FastICA, called scGMAI. Many challenges currently exist regarding the analysis of scRNA-Seq data, such as the curse of dimensionality and dropout events. Both of which are addressed by scGMAI. This model uses autoencoder networks to reconstruct data, improving not only the clustering and the visualization but also more accurately identifies the signature genes of cell clusters. FastICA in scGMAI chooses the key independent features to form a low-dimensional space that still represents all of the essential characteristics of the data. FastICA can quickly calculate and process massive data further enhancing the efficiency of scRNA-Seq analysis. Furthermore, the Gaussian mixture model used by scGMAI automatically estimates the number of clusters in scRNA-Seq data and reveals the inherent nature and regularity of scRNA-Seq data.

We compare the performance of scGMAI to other clustering methods using the scRNA-Seq data from 17 public datasets. The results demonstrate the performance of scGMAI is superior to other cell clustering methods that analyze scRNA-Seq data. In short, scGMAI accurately clusters and identifies cell types, augmenting downstream analysis of scRNA-Seq data. Specifically, we improved the accuracy of clustering scRNA-Seq data through the reconstruction of scRNA-Seq data via autoencoder networks. This presents novel methods and perspectives for studying scRNA-Seq data and other areas of biological research, such as gene regulatory networks [46], protein function prediction [47, 48].

Although scGMAI can effectively cluster and identify cell types in scRNA-Seq data, the autoencoder model can be further developed. scGMAI does not seem to have many advantages in running time. In the future, we plan to optimize the iteration process to speed up the running time and continue to improve the scalability of scGMAI.

Key Points

- We propose a new clustering method, scGMAI, to cluster cells from scRNA-Seq data. It accurately clusters and identifies cell types, providing effective help for downstream analysis of scRNA-Seq.
- scGMAI can deal with dropout events by reconstructing data and avoid the curse of dimension by reducing the dimensionality of the scRNA-Seq data.
- Performance evaluation of scGMAI demonstrates its superiority at clustering cells from scRNA-Seq data compared to alternative state-of-the-art methods.

Supplementary data

Supplementary data are available on *Briefings in Bioinformatics*.

Funding

This work was supported by the National Nature Science Foundation of China (61863010), the Key Research and Development Program of Shandong Province of China (2019GGX101001) and the Natural Science Foundation of Shandong Province of China (ZR2018MC007).

Conflict of interest

The authors declare that they have no competing interests.

References

1. Shalek AK, Satija R, Adiconis X, *et al.* Single cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature* 2013;**498**(7453):236–40.
2. Buettner F, Natarajan KN, Casale FP, *et al.* Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nat Biotechnol* 2015;**33**(2):155–60.
3. Van Loo P, Voet T. Single cell analysis of cancer genomes. *Curr Opin Genet Dev* 2014;**24**:82–91.
4. Zheng R, Liang Z, Chen X, *et al.* An adaptive sparse subspace clustering for cell type identification. *Front Genet* 2020;**11**:407.
5. Bacher R, Kendziorski C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biol* 2016;**17**:63.
6. Petegrosso R, Li Z, Kuang R. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. *Brief Bioinform* 2020;**21**(4):1209–23.
7. Huang M, Wang J, Torre E, *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nat Methods* 2018;**15**(7):539–42.
8. Eraslan G, Simon LM, Mircea M, *et al.* Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun* 2019;**10**(1):390.
9. Van Dijk D, Sharma R, Nainys J, *et al.* Recovering gene interactions from single-cell data using data diffusion. *Cell* 2018;**174**(3):716–29.
10. Li WV, Li JJ. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat Commun* 2018;**9**(1):997.
11. Qi R, Ma A, Ma Q, *et al.* Clustering and classification methods for single-cell RNA-sequencing data. *Brief Bioinform* 2020;**21**(4):1196–208.
12. Wold S, Esbensen K, Geladi PJC, *et al.* Principal component analysis. *Chemom Intel Lab Syst* 1987;**2**(1):37–52.
13. Der Maaten LV, Hinton GE. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**:2579–605.
14. Becht E, Mcinnes L, Healy J, *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol* 2019;**37**(1):38–44.
15. Pierson E, Yau C. ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol* 2015;**16**(1):241.
16. Lin C, Jain S, Kim H, *et al.* Using neural networks for reducing the dimensions of single-cell RNA-Seq data. *Nucleic Acids Res* 2017;**45**(17):e156.
17. Sun Z, Wang T, Deng K, *et al.* DIMM-SC: a Dirichlet mixture model for clustering droplet-based single cell transcriptomic data. *Bioinformatics* 2018;**34**(1):139–46.
18. Zheng R, Li M, Liang Z, *et al.* SinNLRR: a robust subspace clustering method for cell type detection by non-negative and low-rank representation. *Bioinformatics* 2019;**35**(19): 3642–50.
19. Wang B, Zhu J, Pierson E, *et al.* Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Methods* 2017;**14**(4):414–6.
20. Xu C, Su Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* 2015;**31**(12):1974–80.
21. Duverle DA, Yotsukura S, Nomura S, *et al.* CellTree: an R/bioconductor package to infer the hierarchical structure of cell populations from single-cell RNA-seq data. *BMC Bioinf* 2016;**17**(1):363.
22. Kiselev VY, Kirschner K, Schaub MT, *et al.* SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**(5):483–6.
23. Satija R, Farrell JA, Gennert D, *et al.* Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 2015;**33**(5):495–502.
24. Chen C, Wu C, Wu L, *et al.* scRMD: imputation for single cell RNA-seq data via robust matrix decomposition. *Bioinformatics* 2020;**36**(10):3156–61.
25. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. *J Mach Learn Res* 2003;**3**(Jan):993–1022.
26. Hinton GE, Salakhutdinov R. Reducing the dimensionality of data with neural networks. *Science* 2006;**313**(5786):504–7.
27. Deng Y, Bao F, Dai Q, *et al.* Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nat Methods* 2019;**16**(4):311–4.
28. Wang C, Liu J, Luo F, *et al.* Pairwise input neural network for target-ligand interaction prediction. *Int Conf Bioinf Biomed* 2014;1:67–70.
29. Dugas C, Bengio Y, Belisle F, *et al.* Incorporating second-order functional knowledge for better option pricing. *Neural Inf Process Syst* 2002;13:472–8.
30. Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. *Neural Netw* 2000;**13**(4–5):411–30.
31. Nascimento M, e Silva FF, Safadi T, *et al.* Independent component analysis (ICA) based-clustering of temporal RNA-seq data. *PLoS One* 2017;**12**(7):e0181195.
32. Calhoun VD, Adali T, Pearlson GD, *et al.* A method for making group inferences from functional MRI data using independent component analysis. *Hum Brain Mapp* 2001;**14**(3): 140–51.

33. Rau A, Maugis-Rabusseau C. Transformation and model choice for RNA-seq co-expression analysis. *Brief Bioinform* 2018;**19**(3):425–36.
34. Do CB, Batzoglou S. What is the expectation maximization algorithm? *Nat Biotechnol* 2008;**26**(8):897–9.
35. Arthur D, Vassilvitskii S. k-Means ++: the advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* 2007;**11**(6): 1027–35.
36. Chen J, Chen Z. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika* 2008;**95**(3):759–71.
37. Pedregosa F, Varoquaux G, Gramfort A, *et al.* Scikit-learn: machine learning in python. *J Mach Learn Res* 2013;**12**(10):2825–30.
38. Strehl A, Ghosh J. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *J Mach Learn Res* 2002;**3**(3):583–617.
39. Hubert L, Arabie P. Comparing partitions. *J Classif* 1985;**2**(1):193–218.
40. Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *J Mach Learn Res* 2010;**11**: 2837–54.
41. Lin P, Troup M, Ho JWCIDR. Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol* 2017;**18**(1):59.
42. Shao C, Höfer T. Robust classification of single-cell transcriptome data by nonnegative matrix factorization. *Bioinformatics* 2017;**33**(2):235–42.
43. Wan S, Kim J, Won KJ. SHARP: hyper-fast and accurate processing of single-cell RNA-seq via ensemble random projection. *Genome Res* 2020;**30**(2):205–13.
44. Hartigan JA, Wong MA. Algorithm AS 136: a k-means clustering algorithm. *J R I State Dent Soc* 1979;**28**(1):100–8.
45. Luxburg U. A tutorial on spectral clustering. *Stat Comput* 2007;**17**(4):395–416.
46. Zheng R, Li M, Chen X, *et al.* Bixgboost: a scalable, flexible boosting based method for reconstructing gene regulatory networks. *Bioinformatics* 2019;**35**(11):1893–900.
47. Yu B, Qiu W, Chen C, *et al.* SubMito-XGBoost: predicting protein submitochondrial localization by fusing multiple feature information and eXtreme gradient boosting. *Bioinformatics* 2020;**36**(4):1074–81.
48. Wang X, Yu B, Ma A, *et al.* Protein-protein interaction sites prediction by ensemble random forests with synthetic minority oversampling technique. *Bioinformatics* 2019;**35**(14):2395–402.