OXFORD

Briefings in Bioinformatics, 2022, 1–13
https://doi.org/10.1093/bib/bbac018
Problem Solving Protocol

Deep structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural network

Yanglan Gan, Xingyu Huang, Guobing Zou, Shuigeng Zhou and Jihong Guan

Corresponding author. Jihong Guan, Computer Science and Technology, Tongji University 200092, Shanghai, China. Tel: 86-21-65982954; E-mail: jhguan@tongji.edu.cn

Abstract

Single-cell RNA sequencing (scRNA-seq) permits researchers to study the complex mechanisms of cell heterogeneity and diversity. Unsupervised clustering is of central importance for the analysis of the scRNA-seq data, as it can be used to identify putative cell types. However, due to noise impacts, high dimensionality and pervasive dropout events, clustering analysis of scRNA-seq data remains a computational challenge. Here, we propose a new **d**eep **s**tructural **c**lustering method for **sc**RNA-seq data, named scDSC, which integrate the structural information into deep clustering of single cells. The proposed scDSC consists of a Zero-Inflated Negative Binomial (ZINB) model-based autoencoder, a graph neural network (GNN) module and a mutual-supervised module. To learn the data representation from the sparse and zero-inflated scRNA-seq data, we add a ZINB model to the basic autoencoder. The GNN module is introduced to capture the structural information among cells. By joining the ZINB-based autoencoder with the GNN module, the model transfers the data representation learned by autoencoder to the corresponding GNN layer. Furthermore, we adopt a mutual supervised strategy to unify these two different deep neural architectures and to guide the clustering task. Extensive experimental results on six real scRNA-seq datasets demonstrate that scDSC outperforms state-of-the-art methods in terms of clustering accuracy and scalability. Our method scDSC is implemented in Python using the Pytorch machine-learning library, and it is freely available at https://github.com/DHUDBlab/scDSC.

Keywords: deep clustering, scRNA-Seq, ZINB model, autoencoder, graph neural network

Introduction

Single-cell RNA sequencing allows researchers to measure transcriptome-wide gene expression at single-cell resolution and has gradually transformed our understanding of cell biology and human diseases [1]. Despite the unprecedented power of scRNA-seq, processing single-cell data are inherently difficult, especially considering high dimension, technical noises, drop-out events and batch effects in the data [2]. For scRNA-seq data analysis, a critical task is to characterize different cell types in multicellular organisms and their lineage relationships [3]. Knowledge of cell types can reveal cell heterogeneity and diversity across tissues, developmental stages and organisms and provide a deeper understanding of cell and gene function in health and diseases.

As an unsupervised learning method, clustering has been widely used for identifying cell types. Early efforts are mostly based on dimension reduction techniques and traditional clustering methods. The strategy usually first reduce the dimension and then apply the baseline clustering to the low-dimensional data.The widely used

Yanglan Gan is an associate professor in the school of computer science and technology at Donghua University, Shanghai, China. She received the Ph.D. degree in computer science from Tongji University in 2012, China. She has worked as a Visiting Scholar in the Department of Computer Science and Engineering at Washington University in St. Louis from 2009 to 2011, USA. Her research interests include bioinformatics, data mining, and Web services. She has published more than 40 papers on international journals and conferences, including Bioinformatics, IEEE/ACM TCBB, BMC Bioinformatics, BMC Genomics, Knowledge-based Systems, and Soft Computing. She served as a program committee member on BIBM 2021 and GIW 2018. She worked as a reviewer for a variety of international journals and conferences, such as BMC Bioinformatics,IEEE TCBB, Knowledge-based Systems.
Xingyu Huang is currently a master student in the School of Computer Science and technology, Donghua University, China. Before that, she received a Bachelor degree in Computer Science and Technology at Anhui University of Chinese Medicine, 2016. Her research interests include bioinformatics and machine learning.
Guobing Zou is an Associate Professor and Dean of the Department of Computer Science and Technology, Shanghai University, China. He received his PhD in Computer Science from Tongji University, Shanghai, China, 2012. His current research interests focus on data mining, intelligent algorithms and services computing. He has published around 70 papers on premier international journals and conferences,including Information Sciences, Expert Systems with Applications, Knowledge-Based Systems, IEEE Transactions on Services Computing, AAAI, ICWS and ICSOC.
Shuigeng Zhou received the Bachelor's degree from the Huazhong University of Science and Technology, in1988, the Master's degree from the University of Electronic Science and Technology of China, in 1991, and the PhD degree in Computer Science from Fudan University, Shanghai, China, in 2000. He is a full professor in the School of Computer Science, Fudan University. His research interests include data management, data mining, machine learning, and bioinformatics. He has extensively published in domestic and international journals and conferences.
Jihong Guan received the bachelor's degree from Huazhong Normal University in 1991, the master's degree from the Wuhan Technical University of Surveying and Mapping (merged into Wuhan University in 2000) in 1998, and the Ph.D. degree from Wuhan University in 2002. She is currently a Professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. Her research interests include databases, data mining, distributed computing, bioinformatics, and geographic information systems. She has extensively published more than 300 papers in domestic and international Journals (including Nature Communications, IEEE TKDE/ TITS/TSC/TGRS/TCBB, NAR, Bioinformatics) and conferences (including AAAI, ICDE, VLDB, SIGIR, RECOMB, DASFAA).

Received: November 10, 2021. **Revised:** December 27, 2021. **Accepted:** January 13, 2022

dimension reduction methods include principal component analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), diffusion diagram and Uniform Manifold Approximation and Projection (UMAP).It is worth noting that UMAP is a manifold dimension reduction technology based on the theoretical framework of Riemannian geometry and algebraic topology, which is highly competitive in the visualization quality and can retain more global structures [4]. As traditional clustering methods developed for bulk data might not be appropriate for the analysis of scRNA-seq data, researchers turn to devise new methods for clustering single cells. The method SIMLR [5] can effectively perform tasks such as reduced-dimension, clustering and visualization for scRNA-seq data. The essence of SIMLR is to combine multiple kernels to learn the similarity between samples and perform spectral clustering. CIDR [6] is a novel and fast implicit interpolation method, which considers all zeros in gene expression data as dropout candidates to reduce the impact of dropout in scRNA-seq data, and finally cluster single cells using the first few principal coordinates. Recently, with the breakthrough of deep neural networks, several deep clustering methods have emerged, such as deep embedding clustering (DEC) [7], IDEC [8], DCA [9], scDeepCluster [10] and scVAE [11]. Specifically, by utilizing an autoencoder to simultaneously learn feature representation and cluster assignments, the methods DEC and IDEC, respectively, adopting different loss functions as the constraint to improve the resulted cluster cohesion. The method DCA replaces traditional mean square error (MSE) loss function with a zero-inflated negative binomial (ZINB) model-based loss function for better characterizing scRNA-seq data. scDeepCluster also utilizes the ZINB model-based autoencoder to optimize the latent feature learning for the subsequent clustering. The method scVAE adopts a deep variational autoencoder to cluster scRNA-seq data.

Although these deep clustering methods have made great progress [12], they still face two challenges. First, they usually focus on learning the feature representation of the data itself, but ignore the structural relationship among data samples. In reality, such structural information is effective in revealing the latent similarity among cells, and therefore provides a valuable guidance on cell clustering. In previous studies, several methods, including DSC [13], MPSSC [14], conCluster [15] and SDCN [16], exploit the structural information of data for clustering. For example, the method MPSSC enhances sparse structure through L1 penalty to obtain better clustering performance. The method conCluster is based on KNN graph and ensemble clustering. SDCN introduces graph convolutional network to capture the latent relationship among data samples. As these methods mainly rely on Laplacian matrix of the whole graph [17], they are sensitive to the change of similarity graph and the selection of clustering parameters. Meanwhile, the calculation and storage costs of these methods are high. Due to the characteristics of scRNA-seq data, the relationship among cells is far more complex than the neighbor relationship in the geometric space. Even if there is no direct relationship between two cells, and the potential similarity can be revealed through their common neighbors. Therefore, it is necessary to extract the relevant information among cells from a high-order structure. The latest methods including scGNN [18] and scGSLC [19] apply GNN structure to clustering scRNA-seq data and achieve good results in clustering.The method scGNN utilizes GNN and multimodal autoencoder to formulate and aggregate cell–cell relationships.The method scGSLC integrates GNN and a bag of features model, calculates the graph similarity through unsupervised learning and applies it to single-cell clustering problems. These studies indicate that GNN is able to learn the low-dimensional representation of the global topology, exploit neighbor node information and community networks to gradually explore the relationship in the graph, which can been used to obtain structured information and cell segregation.

Second, the grand challenge for deep clustering methods [20] is to accurately simulate on the complex distribution of scRNA-seq data. Considering the sparsity and zero-inflated of scRNA-seq data, it is very important to assume a reasonable distribution of data. Initially, researchers adopt the normal distribution to simulate the distribution of scRNA-seq data. Subsequently, researchers based on the non-negativity of scRNA-seq data, poisson distribution is adopted to describe the distribution of data. However, the sample variance and the mean of Poisson distribution are equal, whereas the real scRNA-seq data exhibits overdispersion pattern. Generally speaking, for scRNA-seq data, the difference between sample variance and sample means value become larger with the average gene expression level increases, which does not conform to the nature of Poisson distribution. On the basis of the Poisson distribution hypothesis, researchers further propose the Gammar distribution as the conjugate prior of *gamma* in the Poisson distribution from a statistical view and prove that the data follows the negative binomial distribution (NB) [21]. Unfortunately, due to the difference among the sequencing technologies, this assumption is not suitable for solving the zero-inflated problem retlated with some kinds of scRNA-seq data. Then, ZINB distribution in the field of is increasingly favored for scRNA-seq data analysis [9, 10, 22]. Specifically, DCA adopts the loss function based on ZINB model rather than the traditional MSE loss function to better simulate the scRNA-seq data distribution [9]. scDeepCluster utilizes the ZINB model to analyze the different expression of microbial sequencing data, and indicates that it can effectively characterize discrete, overdispersion and zero-expansion counting data [10]. However, for scRNA-seq data obtained from different sequencing platforms, researches have disputed the adaptability of the ZINB model. Previous studies have shown that the distribution of UMI count

is not zero expansion [23], and NB distribution is suitable for UMI-based scRNA-seq data [21]. Due to this controversy, it is necessary to explore the characteristics of scRNA-seq data obtained by different sequencing technologies and assume a suitable data distribution for further analysis.

To address these two issues, we propose scDSC, a new deep structural clustering method for scRNA-seq data analysis. scDSC formulates and aggregates cell-cell relationships with graph neural networks (GNN) and learns embedded gene expression patterns using a ZINB model based autoencoder module. Specifically, aiming to characterize the sparsity and drop-out events of scRNA-seq data, we incorporate a ZINB model into the basic autoencoder. Then, a mutual supervision module is used to supervise the end-to-end learning process of structural information and feature representation and jointly optimize the GNNs and the ZINB-based autoencoder module. Further, to expand the applicability in different types of sequencing data, scDSC provides two data distribution model components (ZINB and NB) for users to choose. To benchmark the performance of our approach, we compare the results to a variety of state-of-the-art deep clustering methods. The extensive experimental results on six real scRNA-seq datasets demonstrate that scDSC exhibit obvious advantages over the competing methods. Specifically, scDSC has made significant improvements over the best baseline method.

Materials

The framework of deep structural clustering scDSC

The main architecture of scDSC is proposed to learn effective representations of cells and genes that are useful for performing cell clustering in scRNA-Seq data. As illustrated in Figure 1, it consists of three main computational components, including an autoencoder based on ZINB model for the gene expression embedding, a GNN module for cell graph representation and a mutual supervision module for simultaneous optimization of the autoencoder and the GNN. According to the pipeline, the processed scRNA-seq data are first input into the autoencoder with a ZINB model, embedding the gene expression data into a low dimensional space. Meanwhile, according to the gene expression patterns of cells, we construct a KNN graph and input it into the GNN module to extract the structural relationships among cells. Specifically, by connecting the coding layers of the autoencoder with the GNN layer-by-layer, we integrate the embedded structural information and feature representation to further transmit in the joint framework. Finally, we adopt a multi-module mutual supervision strategy to achieve end-to-end training and simultaneous optimization of the ZINB model-based autoencoder and the GNN. Using the resulted informative and compact representation, cell segregation can be performed with high accuracy and a tractable time complexity.

ZINB model based autoencoder module. To simulate the distribution of scRNA-seq data and learn the effective feature representation of the data, here we adopt an unsupervised autoencoder based on ZINB model. The ZINB distribution is used to model the highly sparse and overdispersed gene expression data.

Assuming that the input scRNA-seq data of the model are $X \in E^{C\times G}$, where each row x_i represents the i-th sample, C is the number of cells, G is the number of genes. The basic autoencoder is composed of two parts: an encoder and a decoder. The encoder maps the input Xinto the encoded representation H, and the decoder maps H into a reconstruction $\bar{X}$ of the input. Supposing that the encoder has L layers, for the l_{th} layer, the data representation learned from this layer is $H_{(l)}$, the weight is $w_{(l)}$ and the bias vector is $b_{(l)}$. The learning process of the l_{th} layer of the autoencoder is defined as:

$$H_{(l)} = \phi(w_{(l)}H_{(l-1)} + b_{(l)}) \tag{1}$$

The encoder stage of the autoencoder maps $X$to the latent representation H as:

$$H = f_{enc}(WX + b) \tag{2}$$

The decoder stage of the autoencoder maps$H$to the reconstruction $\bar{X}$ as:

$$\bar{X} = f'_{dec}(W'H + b') \tag{3}$$

where W and W', respectively, represent the weight matrix of the encoder and the decoder, b and b' are the bias vectors of the encoder and the decoder.

Different from the traditional autoencoder, the autoencoder based on ZINB model connects three independent full connection layers with the last layer of the decoding layer to estimate the three parameters of ZINB [9]: dropout rate π, dispersion degree θ and mean μ. In fact, we need to use the matrix form of these parameters for calculation, and define the matrix forms of π, θ and μ as:

$$Dropout = Sigmoid(w_\pi D) \tag{4}$$

$$Disp = Exp(w_\theta D) \tag{5}$$

$$Mean = diag(S_i) \times exp(w_\mu D) \tag{6}$$

where $D = f_{dec}(f_{enc}(X))$ is the output matrix of the last layer of the decoding layer, and the size factor S_i is the ratio of the total cell count to the median S. The ZINB model simulating the scRNA-seq data distribution is parameterized by the ZINB distribution:

$$NB(X \mid \mu, \theta) = \frac{\Gamma(X+\theta)}{X!\,\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{X} \tag{7}$$

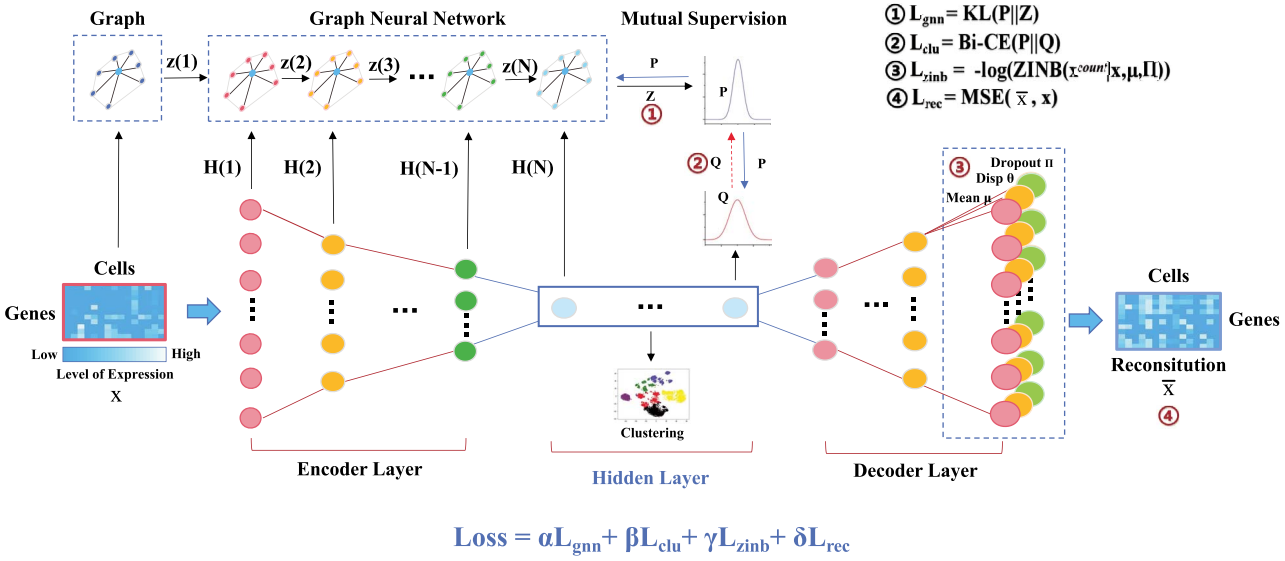

Figure 1. The overview of the proposed method scDSC. In the framework, scDSC is mainly composed of ZINB model-based autoencoder module (*ZAE*), GNN module (*GM*) and multiple mutual supervision module (*MSM*). The coding layers and decoding layers of *ZAE* are fully connected neural network layer, x and $\bar{x}$ are input data and reconstructed data of autoencoder, respectively. We use the encoder hidden layer to learn the embedded feature representation. In the decoder, we use ZINB model to simulate the distribution of scRNA-seq data. ZINB model is composed of three activation layers with different activation functions, namely *Mean*, *Disp* and *Dropout*. At the connection of *GM* and *AEM*, $H_{(l)}$ and $Z_{(l)}$ represents the representation of the l_{th} layer of the coding layer and the representation of the L_{th} layer of the GNN, respectively, and learn more rich information through layer-by-layer integration and transmission. Finally, we use a multiple mutual supervision strategy to conduct the mutual learning of different modules. In *MSM*, we use the target distribution P to calculate the clustering distribution Q and then Q supervises the learning process of P, and P also supervises the learning process of probability distribution Z. In this way, we put multiple networks in one framework to achieve synchronous updating of *AEM* and *GM*.

$$ZINB(X \mid \pi, \mu, \theta) = \pi\delta(X) + (1-\pi)\left[\frac{\Gamma(X+\theta)}{X!\,\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{X}\right] \quad (8)$$

Finally, we define the loss function as the sum of the negative log of ZINB distribution. The training objective of scDSC in this module is to minimize the loss function of autoencoder based on ZINB model:

$$L_{zinb} = \sum -log(ZINB(X \mid \pi, \mu, \theta)) \quad (9)$$

GNN based on K-nearest neighbor graph (KNN). To capture the relationships among cells, we further utilize GNN based on KNN. As in previous studies [19, 24, 25], we construct the KNN graph according to the input gene expression data of cells. In the KNN graph, each node represents a cell, and the edges between nodes represent the relationship between cells. For example, assuming that x_i, x_j and x_k, respectively, represent cell i, cell j and cell k, if x_i and x_j are adjacent, and x_j and x_k are adjacent, it might be inferred that x_i and x_k also have certain similarity, with a potential connecting structure. To better capture the structural information, we adopt and compare four widely used methods of constructing KNN graph, including Cosine Similarity, Pearson correlation coefficient, normalized Cosine Similarity and Heat Kernel [26]:

(1) Cosine Similarity:

$$S_{ij} = \frac{\sum_{i=1,j=1}^{n} x_i \cdot x_j}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{j=1}^{n} x_j^2}} \quad (10)$$

(2) Pearson correlation coefficient:

$$S_{ij} = \frac{\sum_{i=1,j=1}^{n}(x_i - \bar{x})(x_j - \bar{x})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{j=1}^{n}(x_j - \bar{x})^2}} \quad (11)$$

(3) Normalized Cosine Similarity:

$$S_{ij} = \frac{\sum_{i=1,j=1}^{n}\left(\frac{x_i - x_{min}}{x_{max} - x_{min}}\right)\left(\frac{x_j - x_{min}}{x_{max} - x_{min}}\right)}{\sqrt{\sum_{i=1}^{n}\left(\frac{x_i - x_{min}}{x_{max} - x_{min}}\right)^2}\sqrt{\sum_{j=1}^{n}\left(\frac{x_i - x_{min}}{x_{max} - x_{min}}\right)^2}} \quad (12)$$

(4) Heat Kernel:

$$S_{ij} = e^{-\frac{\left\|x_i - x_j\right\|^2}{t}} \quad (13)$$

Based on the constructed KNN, we utilize GNN for information dissemination and integration. The input of the GNN is the KNN graph structure. Subsequently, GNN spreads layer by layer through weight calculation to learn

the representation of the l_{th} layer $Z_{(l)}$ as follows:

$$Z_{(l)} = \phi\left(\tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}Z_{(l-1)}W_{(l-1)}\right) \quad (14)$$

where $\tilde{D}_{ii} = \sum_j (A_{ij} + I_{ij})$, A is the self-cycle of each node, and I is the unit diagonal matrix of adjacent matrix. According to the above equation, $Z_{(l-1)}$ is transmitted through the normalized adjacency matrix $\tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$ to obtain the new representation $Z_{(l)}$. Through this information spread, we obtain the structured information among cells. At the same time, we add the representation learned by the ZINB-based autoencoder to the GNN to obtain richer representation:

$$\tilde{Z}_{(l-1)} = \sigma Z_{(l-1)} + (1-\sigma)H_{(l-1)} \quad (15)$$

As a result, we obtain two different representations of data, including the feature representation of gene expression data and the relationships between cells. To update the next layer representation of GNN, we use the integrated representation $\tilde{Z}_{(l-1)}$ to generate a new representation:

$$Z_{(l)} = \phi\left(\tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}\tilde{Z}_{(l-1)}W_{(l-1)}\right) \quad (16)$$

In the learning process of the GNN, we pay more attention to the representation learned by coding layer and hidden layer. Therefore, the last layer of GNN is designed to connect with the last layer of the hidden layer, and Softmax is used as the activation function:

$$z = Softmax(\tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}\tilde{Z}_{(l-1)}W_{(l-1)}) \quad (17)$$

Finally, we obtain the probability distribution $Z(z_{it} \in Z)$, that is, the probability that cell i belongs to cluster t.

Mutual supervision module. By connecting the coding layer of the autoencoder with the GNN module, we obtain a rich representation of the data through layer-by-layer spread and integration of information. But the resulted information cannot be directly used for clustering. Therefore, we adopt a mutual supervision strategy to train the entire network and perform the clustering task. In the mutual supervision module, it mainly consists of three distributions: target distribution P, clustering distribution Q and probability distribution Z. These three distributions are mutually supervised and unified in the same framework to achieve end-to-end network learning and training.

For cell i and cluster t, we define a soft label: $q_{it} \in Q$, which represents the similarity between the data representation $h(i)$ of the current i_{th} layer coding layer and the clustering center vector $\mu(t)$ initialized by K-means clustering. Q represents the probability set that all cells are assigned to the cluster center. In the selection of probability distribution model, we use Student t distribution to measure the similarity between $h(i)$ and $\mu(t)$:

$$q_{it} = \frac{(1 + \|\mu_t - h_i\|^2/v)^{-\frac{v+1}{2}}}{\sum_{j'}(1 + \|\mu_{t'} - h_i\|^2/v)^{-\frac{v+1}{2}}} \quad (18)$$

where v is the degree of freedom.

During this process, the data are expected to be as close as possible to the clustering center of real data. Therefore, we use soft label frequency F to get higher confidence data representation $p_{it} \in P$.

$$F = \sum_i q_{it} \quad (19)$$

$$p_{it} = \frac{q_{it}^2/F_t}{\sum_{t'} q_{it'}^2/F_{t'}} \quad (20)$$

Finally, we achieve better clustering results by minimizing the binary cross entropy between P and Q:

$$L_{clu} = -p_{it}log(q_{it}) - (1-p_{it})log(1-q_{it}) \quad (21)$$

The formula (17) (18) show that P is calculated by Q, and Q supervises P learning. This mutual supervision strategy can help the autoencoder learn a better representation of data to obtain higher quality clustering. In the layer-by-layer propagation of GNN networks, we get a representation Z that contains both of the data features and the relationship between cells. Similarly, we can use P to supervise Z in the learning process:

$$L_{gnn} = KL(P \parallel Z) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{z_{ij}} \quad (22)$$

Thus, in the training process, distribution Q and distribution Z have a common learning goal, approaching the target distribution P together. They form an interactive relationship that supervises each other to learn and optimize the goal. By constructing a ZINB model simulating scRNA-seq data distribution and a GNN with rich information dissemination, scDSC has four optimization objectives. The loss function of the model is defined as:

$$Loss = \alpha L_{gnn} + \beta L_{clu} + \gamma L_{zinb} + \varepsilon L_{rec} \quad (23)$$

where α, β, γ and ε are the coefficients that controls the relative weights of the four losses.

Datasets

The proposed scDSC method is evaluated on six real scRNA-seq datasets, and each scRNA-seq dataset contains cells whose labels are known as a prior or validated in the previous studies. The detailed characteristics of these datasets are summarized in Table 1. The six real datasets were derived from five different sequencing platforms. The mouse bladder cell dataset consisted of 2746 bladder cells from mouse bladder tissue, profiled by Microwell-seq platform [27]. The worm neuron cell dataset is sequenced by scRNA-seq, including 4186 worm

Table 1. Summary of the real scRNA-seq datasets

Datasets	Sequencing platform	Cells	Genes	Subtypes
Mouse bladder cells	Microwell-seq	2746	20670	16
Worm neuron cells	sci-RNA-seq	4186	13488	10
MTAB	Smart-seq2	1529	4889	5
LPS	Array	306	2047	4
PBMC	10X genomics	4271	16449	8
GSE72056	Smart-seq2	4645	16834	7

neuron cells [28]. MTAB dataset is composed of 1529 human preimplantation embryonic cells collected on days 3, 4, 5, 6 and 7, sequenced by Smart-seq2 [29]. The LPS dataset includes a total of 306 cells collected at 1, 2, 4 and 6 hours [30], profiled by array. The datasets GSE72056 consisted of 4645 melanoma cells isolated from 19 patients, sequenced by Smart-seq2 [31]. PBMC is from the 10X genomics platform, including 4271 peripheral blood mononuclear cells from a healthy donor [32].

As these scRNA-seq datasets obtained from different platforms, we adopt different strategies of data pre-processing according to the characteristics of the datasets. On PBMC, Mouse, and Worm datasets, we pre-process the original scRNA-seq data by the Python package SCANPY (version 1.7.2) [33]. First of all, we filter out the genes with high expression only in a single cell and filter out the genes expressed in less than three cells. Then we normalize the data and transform the data by logTPM. On the GSE72056, LPS, and MTAB datasets, we first perform feature variance filtering and normalization on the data and then perform data imputation by the tool MAGIC [34] according to the data density to obtain higher-quality count data. Finally, we use the pre-processed data as the input of scDSC.

Evaluation metrics

To validate the clustering algorithm effectively, we retrieve those scRNA-seq datasets with cell labels, which are known as a prior or validated in the previous studies. For the input scRNA-seq data $x \in E^{C\times G}$ (C represents the number of cells and G indicates the number of genes), we represent the prior cell labels as a vector $L = [l_1, l_2, l_3 \ldots l_C] \in R^C$. By clustering, we obtain a set of predicted labels $U = [u_1, u_2, u_3 \ldots u_C] \in R^C$. Here, three widely used metrics are utilized to evaluate the clustering performance of different algorithms, including clustering accuracy (ACC), Normalized Mutual Information (NMI) [35] and Adjusted Rand Index (ARI). The larger value means more concordance between the predicted labels and the real labels.

Clustering ACC is commonly used to measure the difference between the real labels and the predicted labels. ACC represents the proportion of the predicted labels obtained by clustering in the real labels of data. ACC is calculated as:

$$ACC = \frac{\sum_{i=1}^{N} \delta(L_i, map(U_i))}{N} \tag{24}$$

where $\delta(x_1, x_2)$ is an indicator function, if $x_1 = x_2$ then $\delta(x_1, x_2) = 1$, otherwise $\delta(x_1, x_2) = 0$. The map function represents the distribution of reconstruction between the predicted and real labels after clustering.

NMI is often used to measure the similarity between the predicted labels and the real labels. NMI is defined as:

$$NMI = \frac{MI(L, U)}{F(H(L), H(U))} \tag{25}$$

where $MI = \sum_{i=1}^{N}\sum_{j=1}^{C} p_{i,j} log\left(\frac{p_{i,j}}{p_i, p_j}\right)$ calculates the mutual information between L and U, $H(L) = -\sum_{i=1}^{N} p_i log(p_i)$ and $H(U) = -\sum_{j=1}^{C} p_j log(p_j)$, respectively, represent the information entropy of label vectors L and U. $F(x_1, x_2)$ can be *max*, *min* or *mean* function; here we choose the *max* function.

The ARI is an improved version of Rand Index (RI) [35], which solves the problem of insufficient punishment for RI. ARI is used to measure the similarity between predicted labels and real labels with a value range of $[-1, 1]$. Through calculation, we assume that the overlap between the two label-groups L and U are summarized in the contingency table R. Each item in Table R represents the number of objects shared between L and U, and ARI can be defined as:

$$ARI = \frac{\sum_{i,j}\binom{n_{i,j}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] /2 - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \tag{26}$$

where (.) denotes the binomial coefficient, n_{ij} denotes the data in the contingency table L, and a_i is the sum of the i line of L and b_j is the sum of the j column of L.

Results

Analysis of different methods for KNN graph construction

We utilize GNN based on KNN to capture the relationship among cells. To better capture the structural information, we utilized four different metrics to calculate the similarity between cells, including the Cosine Similarity, Pearson correlation coefficient, Normalized Cosine Similarity and Heat Kernel [16]. We use error rate to evaluate the quality of the constructed KNN graphs and compare these methods on six real scRNA-seq datasets.

Table 2. Error rate of the constructed KNN graphs based on different methods

Datasets	Cos	Pearson	nCos	Heat Kernel
Mouse bladder cells	0.4632	0.2815	0.3486	0.8817
Worm neuron cells	0.5346	0.1054	0.3365	0.8496
MTAB	0.7947	0.2815	0.8775	0.7481
LPS	0.8143	0.3745	0.8144	0.7392
PBMC	0.8210	0.4103	0.3609	0.8137
GSE72056	0.5343	0.1049	0.4248	0.6134

The lower error rate indicates that the constructed KNN graph structure is closer to the structure of real-data relationship.

As listed in Table 2, when we use Pearson coefficient to calculate the similarity between cells, the constructed KNN has a significantly lower error rate than those of the other three methods on five datasets. Compared with the suboptimal method, the error rate decreases by an average of 26.74%, and the error rate of Pearson coefficient on PBMC dataset is slightly higher than that of ncos method. From the perspective of data processing, the calculation of Pearson correlation coefficient is a centralized process, subtracting the mean value of each component of the vector and calculating the cosine similarity. Therefore, Pearson correlation coefficient is an improvement of ncos in the case of missing dimension value, which can better reflect the correlation of two random variables under the same benchmark. Overall, the experimental result demonstrates that Pearson coefficient achieves the best results in calculating the similarity among cells and constructing KNN graph, where as the error rate of the KNN constructed by heat kernel is much higher. Therefore, we select Pearson coefficient as the similarity metric to construct KNN graphs for the following analysis.

Evaluation of balance coefficient σ

In order to integrate the feature representation learned from the autoencoder and the structural information from the GNN, we introduce a coefficient σ to balance the weight of two information distributions (Formula(12)). Aiming to explore the influence of σ value on clustering performance, we apply scDSC to six real scRNA-seq datasets to evaluate its performance. We assign σ with different values ranging from 0 to 1 with step 0.1. Among them, $\sigma = 0$ represents the integrated information of the next layer only contains the feature information $H_{(l)}$ learned by autoencoder, where as $\sigma = 1$ represents that the joint information of the next layer only contains the information $Z_{(l)}$ learned by GNN. Through extensive experiments, we select a better parameter σ to obtain the integrated information beneficial for the clustering performance.

Figure 2 shows the clustering performance (ARI) of scDSC on six real datasets under different σ parameters. We observe that scDSC has poor clustering performance on most datasets at $\sigma = 0$, when σ increase to 0.1, the clustering performance begins to improve significantly,

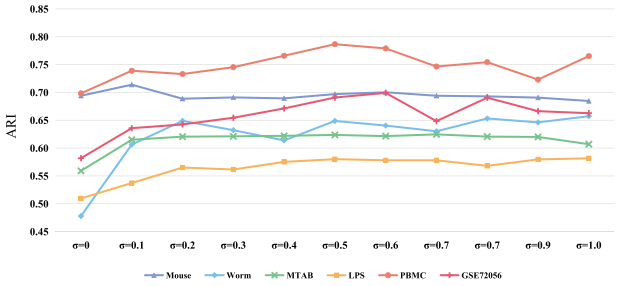

Figure 2. Comparison of clustering performance (ARI) of scDSC on six real datasets under different σ parameters.

indicating that the structural information between cells learned by GNN plays a key role in the whole framework. At the same time, different datasets have different sensitivity to parameter σ. For example, the clustering performance of worm neuron cells dataset fluctuates greatly with the increase of σ, whereas the clustering performance of other datasets gradually tends to be stable after $\sigma \geq 0.2$. The experimental results show that the six datasets have better clustering performance on $\sigma = 0.5$, and the MTAB, LPS, Worm datasets have the best clustering performance on $\sigma = 0.5$, which indicates that GNN and ZINB model-based autoencoder are almost equally important in the proposed deep structural clustering model. When $\sigma = 1$, scDSC also obtains better clustering performance on six datasets. The result indicates that the model can also learn a better feature representation of data based on the mutual supervision of multiple modules, even if the feature representation learned by the autoencoder is not included.

Performance comparison with previous methods

To further evaluate the clustering performance of scDSC, we compare it against eight competing methods. These methods can be categorized into three groups, including traditional clustering methods, the clustering methods based on deep neural network and the clustering methods based on GNN.

- **scDeepCluster** adds a ZINB distribution (ZINB) model simulating the distribution of scRNA-seq data to the denoising autoencoder, learns feature representation and cluster through the explicit modeling of scRNA-seq data.
- **SDCN** [16] is a deep clustering method that uses graph convolutional network to integrate structured

information into deep learning network. It has been applied to clustering images and texts, yet has not been used for the clustering of scRNA-seq data.
- **IDEC** [15] is based on the DEC. It adds the reconstruction loss to retain the partial structure information of the data and to learn better information.
- **SC3** [14] combines multiple clustering solutions through a consistent method to obtain high precision and robust clustering results.
- **scGNN** [19] utilizes GNN and multimodal autoencoder to formulate and aggregate cell–cell relationships. The scGNN graph-embedding contains high-order structural information and preserves topological relationships in cell graphs.
- **CIDR** [6] regards all zeros in gene expression data as dropout candidates to reduce the impact of dropout in scRNA-seq data and finally cluster single cells using the first few principal coordinates.
- **SIMLR** [5] combines multiple cores to learn the similarity between samples and perform spectral clustering, which can perform tasks such as dimensionality reduction, clustering and visualization.
- **DCA** [14] is a deep count autoencoder based on noise reduction. It uses a loss function based on the ZINB model instead of the traditional MSE loss function to better simulate the scRNA-seq data distribution.

In the comparative analysis, we utilize three widely used metrics (ACC, NMI and ARI) to evaluate the clustering performance of each method. As the number of cell clusters k is unknown in real applications, we adopt the Elbow [7, 20, 36] method to determine the optimal value of k. Elbow determines the k value by judging the critical point of distortion degree along the variation of k value, which can be calculated and visualized by using the yellowbrick library of python 3.6 [20].

Figure 3 shows the performance of these clustering methods on the six real scRNA-seq datasets. We observe that scDSC performs better than the other eight competitive methods. Specifically, it is significantly better than the latest scDeepCluster method on Mouse, Worm, MTAB, LPS and GSE72056 datasets and slightly better on PBMC datasets. In particular, on the MTAB dataset, scDSC significantly increases by 7.22% in ACC, 7.05% in NMI and 12.46% in ARI compared with the suboptimal method scGNN. On the Mouse dataset, compared with the suboptimal method scDeepCluster, scDSC significantly increased by 6.12% on ACC, 1.97% on NMI and 9.78% on ARI. On Worm dataset, compared with scDeepCluster, scDSC significantly improves ACC by 7.17%, NMI by 4.93% and ARI by 9.82%.

In order to show the intuitive clustering effect and validate the effectiveness of the model in extracting the low-dimensional representation of high-dimensional data, we use t-SNE to project the features learned from the encoding layer of ZINB model-based autoencoder into the two-dimensional space and visualize the final data embedding. As shown in Figure 4, each point represents a cell, and different colors represent the predicted cell type. In particular, we select two real datasets with more subtypes and higher complexity, including mouse bladder cells (Figure 4**A**) and worm neuron cells (Figure 4**B**). In Figure 4**A**, we observe that scDSC achieves a good separation of mouse dataset with 16 different cell subtypes and the boundary is clear. On the contrary, other methods tend to mix different subtypes of cells. For scDeepCluster, although clear cluster boundaries such as the Dendritic cell_Cd74' and Dendritic cell_Lyz2' cells represented by red and orange points can also be seen, there are still some outliers mixed with Smooth muscle cell' and NK cell' represented by pink and blue points. We have similar observations on the Worm dataset. As shown in Figure 4**B**, on the Worm dataset, compared with scDSC, SDCN, DCA and scDeepCluster can also identify different types of clusters in a sparse manner. However, the identified clusters are dispersed, and the boundaries between clusters are mixed. For example, in SDCN, DCA and scDeepCluster, the flp-1(+) interneurons' and other interneurons' cells represented by black and red are mixed together and are not well distinguished. In SC3 and IDEC, the clustering results are not clear. The touch receptor neurons' cells represent by blue are distributed on the whole figure and are mixed with other types of cells. It is difficult to obtain a clear boundary between clusters. Compared with other clustering methods, the proposed method scDSC identified distinct clusters and clear boundaries between clusters.

In addition, we conducted the clustering visualization analysis on MTAB and LPS datasets with time series information. The MTAB dataset is obtained from 88 human preimplantation embryos, including 1529 cells per embryonic day from E3 to E7. With embryonic stem cells development, the expression pattern of MTAB exhibits heterogeneity. In MTAB time series data, scDSC restores a good data structure and implies a obvious trajectory of cell development from Day 3 to 7 (Figure 4**C**). The LPS datasets consist of scRNA-seq samples of mouse dendritic cells, including 306 cells collected at 1, 2, 4, and 6 hours. Figure 4**D** shows that scDSC well separates four mouse dendritic cells at different time intervals (1, 2, 4 and 6 hours, expressed in different colors), which are well conformed with the cell trajectories. In general, scDSC recover a structure that is not well represented by the raw data, showing a well-aligned trajectory path of cell development.

Ablation study

The proposed scDSC is mainly composed of a ZINB model, an autoencoder and a GNN module. Further, aiming to expand the applicability in different types of sequencing data, scDSC provides two data distribution model components (ZINB and NB) for users to choose. Therefore, to explore the contribution of the ZINB model and the GNN module to the clustering

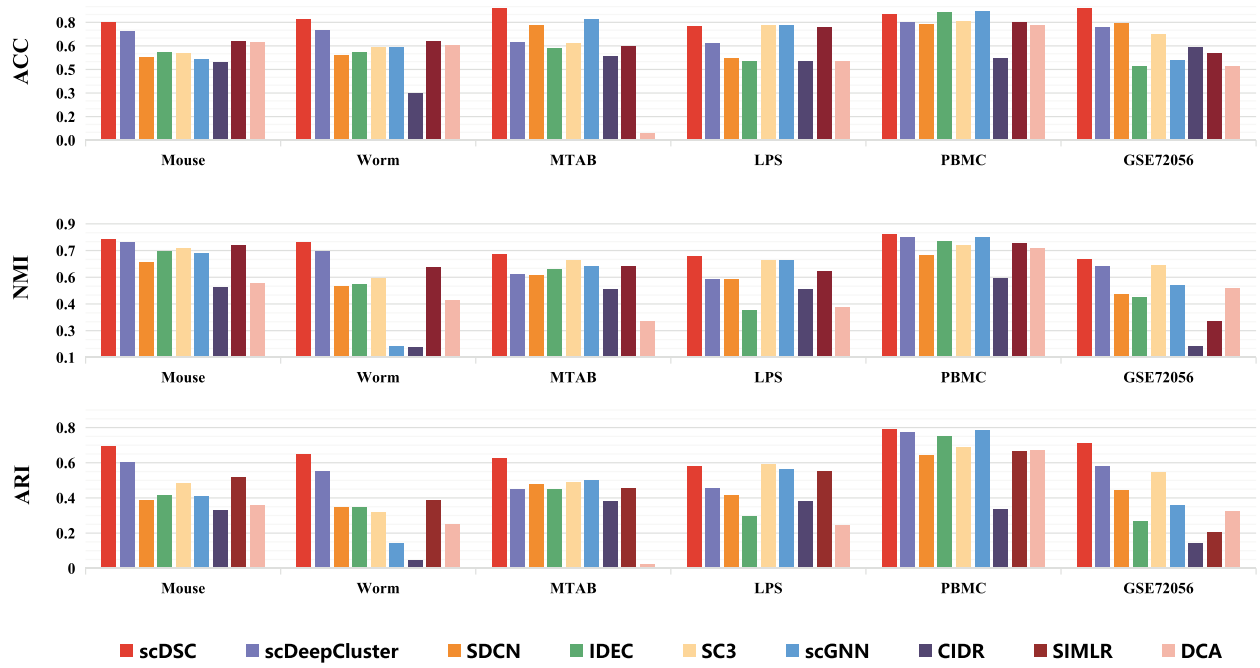

Figure 3. Comparison of clustering performances of scDSC, scDeepCluster, SDCN, IDEC, SC3, scGNN, CIDR, SIMLR and DCA, measured by ACC, NMI and ARI.

performance of scDSC, we set three variants of scDSC: scDSC (NB) (using NB model instead of ZINB to validate the effectiveness of zero inflation), scDSC-G (removing ZINB model to validate the effectiveness of our added ZINB model) and scDSC-Z (removing GNN module to validate the effectiveness of GNN in learning structural information).

As shown in Figure 5, scDSC achieves the best performance compared with its three variants. This result demonstrates that the GNN module and the ZINB model both played an important role in the performance improvement of the whole model. On the PBMC dataset, variant scDSC-Z almost reached the same clustering performance level as scDSC, whereas scDSC-G had poor clustering performance. The reason might be that when constructing KNN graphs, the error rate of the constructed KNN graph is relatively high. This KNN graph further makes the GNN module contain more error information when integrating structural information of cell population with feature representation, resulting in poor clustering performance. Therefore, in order to accurately identify clusters, it is important to obtain a high-quality KNN graph for the GNN module. The experimental results also demonstrate that scDSC (NB) and scDSC achieved similar clustering performance on the scRNA-seq datasets using UMI sequencing, including PBMC, Worm and Mouse. Therefore, Although ZINB has advantages in clustering performance, we recommend using NB model components on Mouse, Worm and other large datasets based on UMI sequencing. Because NB is simpler than ZINB, it can save a certain amount of computing cost and obtain considerable clustering results.

Scalability and efficiency

In order to verify the scalability and efficiency of the proposed model, we test scDSC on simulated datasets with different sizes and analyze the running time. Specifically, we use the Splatter package of the R language to generate seven simulated datasets, respectively, containing 1k, 3k, 5k, 10k, 30k, 50k, 100k cells with 2000 genes.

Here, for obtain a reasonable comparison, the recorded running time includes the time for pre-training and formal training. The results on the seven simulated datasets are shown in Figure 6. From the Figure, we observe that the running time of scDSC does not increase quadratically or exponentially with the number of cells, and its running time increases linearly with the number of cells. In addition, we also compare the accuracy of scDSC on these seven datasets with different sizes. The experimental results demonstrates that scDSC can obtain stable clustering performance on different scale datasets (Supplementary Figure S2).

Conclusion

As the scRNA-seq technology allows researchers to collect large volumes of transcriptomes of individual cells, clustering analysis has been routinely conducted to define cell types from these data. However, existing clustering methods are usually limited to extract the feature representations from gene expression data of individual cells, while ignoring the high-order structural relationships of cell population. In order to overcome this limitation, we proposed a new method, named scDSC, to integrate the structural information of cell

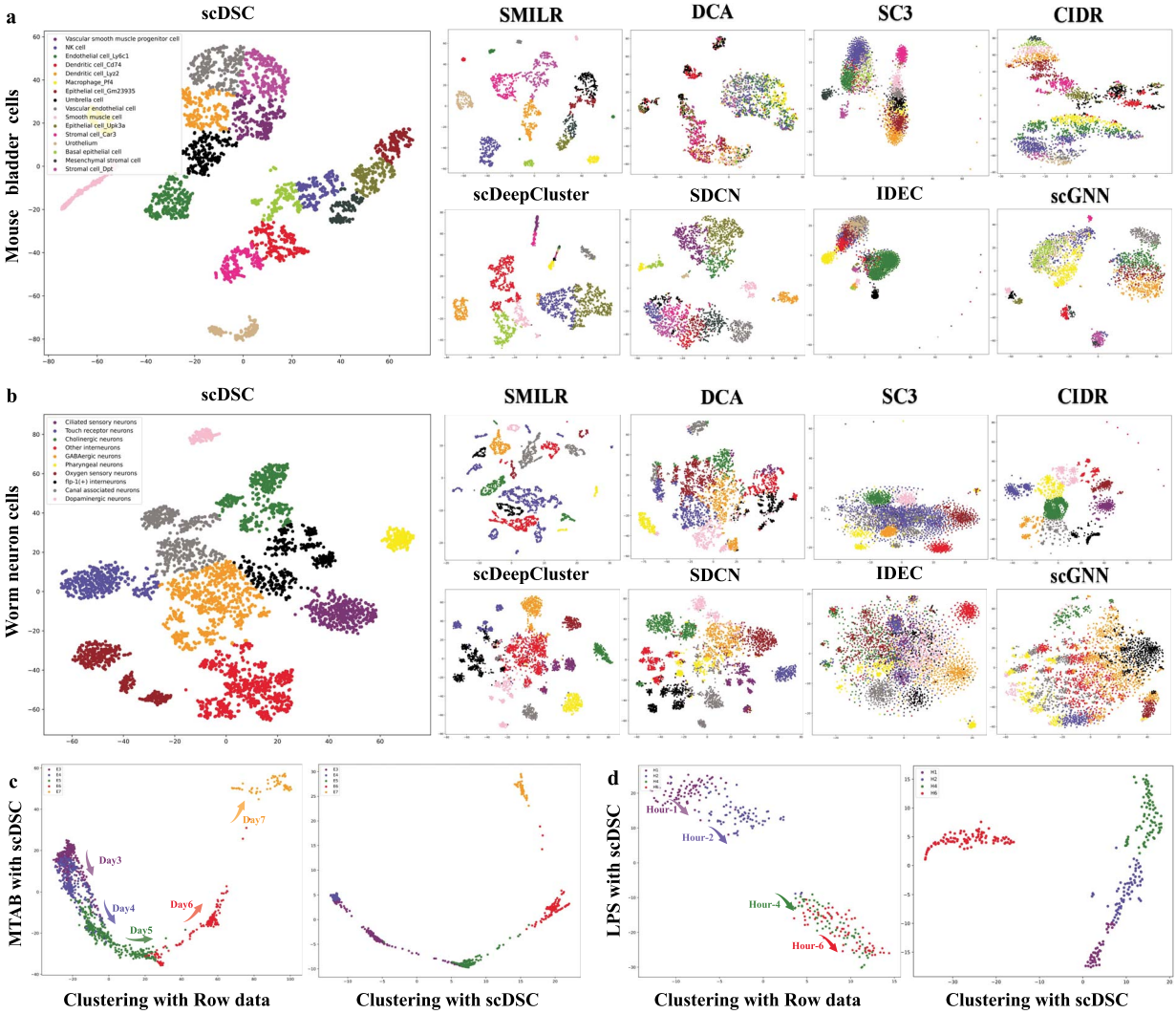

Figure 4. The visualization of the identified clusters of scDSC and eight competitive methods are drawn. These points represent each sample cell, and different colors represent different labels of data. (**A**) The mouse bladder cells. (**B**) The worm neuron cells dataset. (**C**) The MATB dataset. (**D**) The LPS dataset.

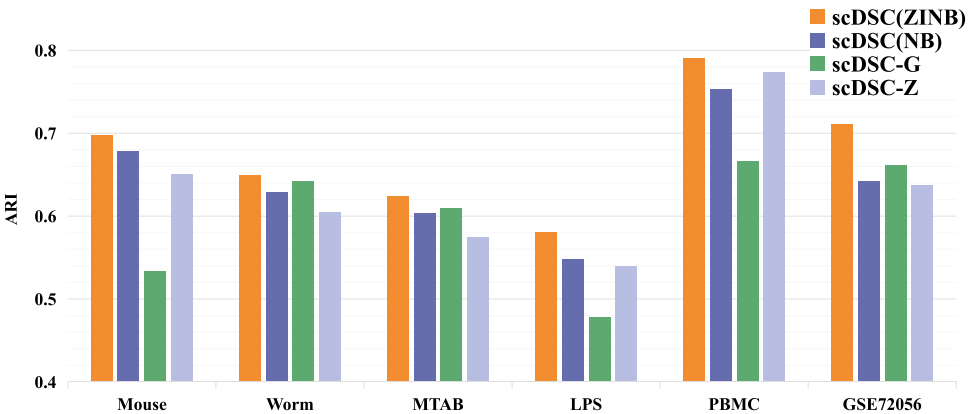

Figure 5. Performance comparison (ARI) of scDSC and its three different variants.

populations into deep clustering of scRNA-seq data. The proposed scDSC consists of three main modules, including a ZINB-based autoencoder, a GNN module and a mutual-supervision module. We integrate the structural information among cells with the learned data representation of autoencoders through GNNs and propagate them layer by layer to obtain rich information in two network structures. Next, the mutual supervision

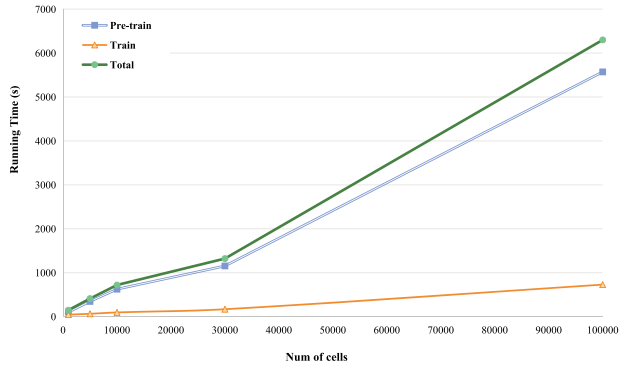

Figure 6. The running time of scDSC on different scale simulation datasets, including pre-training, formal training and total running time.

module is utilized to unify the target distribution, clustering distribution, probability distribution in the same framework, in order to realize end-to-end network learning and training. Moreover, we add the ZINB model to the decoding layer, which was specially designed for scRNA-seq data clustering, and it has proved to better simulate the distribution of scRNA-seq data. To validate the effectiveness of scDSC, we compared it with five state-of-the-art methods on six real scRNA-seq datasets. The extensive experimental results demonstrate that our model scDSC outperforms most of the advanced deep clustering methods on these real scRNA-seq datasets.

Further, although scRNA-seq data can reveal RNA abundance with high throughput, accuracy and sensitivity, it captures only a static snapshot at a point in time. That is to say, it is hard to reveal the kinetics of cell subpopulation along important biological process. On the contrary, RNA velocity describes the rate of gene expression change for an individual gene at a given time point based on the ratio of its spliced and unspliced messenger RNA. In the future, we plan to study subpopulation kinetics and continuous trajectory based on RNA velocity data.

Key Points

- Characterizing different cell types in multicellular organisms is critical to reveal cell heterogeneity and diversity, especially using various clustering methods. A dominant issue in clustering scRNA-seq data is the problems of zero expansion and high dispersion caused by the complex data distribution.
- To tackle this issue, we propose scDSC, a new deep structural clustering method for scRNA-seq data analysis. scDSC introduces ZINB model-based autoencoder and GNN into the deep clustering framework and realize the synchronous learning of feature representation and high-level structural information of the data.
- Specifically, we add a ZINB model to the basic autoencoder, which can effectively learn the data representation from the sparse and zero-inflated scRNA-seq data. The GNN module is introduced to capture the structural information among cells. By joining the ZINB-based autoencoder with the GNN module, the model transfers the data representation learned by autoencoder to the corresponding GNN layer. The multi-module mutual supervision strategy is adopted to realize the simultaneous learning of different types of information and jointly optimize the entire network.
- The experimental results on six real scRNA-seq datasets show that scDSC achieves superior performance compared with state-of-the-art methods.

Data availability

The mouse bladder cells are obtained from the Microwell-seq platform and originated from the Mouse Cell Atlas project (https://figshare.com/s/865e694ad06d5857db4b). We download the count matrix of all 400 000 single cells sorted by tissues and select the cells from the bladder tissue about 2186 cells as experimental data [37]. The worm neuron cells are obtained from the sci-RNA-seq platform [38]. We select the subset of neural cells from the nematode *Caenorhabditis elegans* at the L2 larval stage, and remove the cells with the label Unclassified neurons' in bladder tissue. Finally, we obtain 4186 neural cells as experimental data. MTAB is obtained by Smart-seq2 analysis. After quality control, the author [39] retained 1529 high-quality single-cell transcriptomes from 88 human preimplantation embryos, including 1529 cells per embryonic day (E3–E7). The author use tlr ligands to stimulate mouse primary BMDC and analyzed the gene expression changes on the Affymetrix arrays, using 5 at 9 time points (0.5, 1, 2, 4, 6, 8, 12, 16, 24 hours) A tlr ligand (LPS, pIC, PAM, CpG, GRD) stimulates BMDC. We obtain the LPS dataset consisting of scRNA-seq samples of mouse dendritic cells, which contains 306 cells collected at 1, 2, 4 and 6 hours [30]. 10X PBMC is the 4271 peripheral blood mononuclear cell data of a healthy donor obtained from the 10X scRNA-seq platform [32]. 10X PBMC dataset is downloaded from the website of 10X genomics. The true labels of these datasets are obtained from the corresponding references. GSE72056 is obtained by Smart-seq2 analysis and contains 4645 melanoma tumor cells isolated from 19 patients. Clusters of non-malignant cells are annotated as T cells, B cells, macrophages, endothelial cells, cancer-associated fibroblasts (CAFs) and NK cells on the basis of preferentially or uniquely expressed marker genes [42].

The datasets were derived from the following sources in the public domain: the Mouse bladder cells datasets from https://figshare.com/s/865e694ad06d5857db4b the Worm neuron cells datasets from http://atlas.gs.washington.edu/worm-rna/docs, the MTAB datasets from https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-3929/?query=1529+ the lps datasets from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE17721, the GSE72056 datasets from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi, the PBMC datasets from https://support.10

xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc4k.

Implementation

scDSC is implemented in Python 3 (version 3.6) using PyTorch (version 1.71+cu101). We set the sizes of encoding layers in ZINB model-based autoencoder as (512, 256, 256), the structure of the decoding layer is opposite, and the sizes of embedding layers as (256, 128, 32). We utilize all the data to pre-train the basic autoencoder to obtain a better pre-trained model. Initially we set the learning rate lr = 0.001, epoch = 200 and batchsize = 256, then adopt Adam to adjust the learning rate. During formal training, the size of each layer in the autoencoder module is the same as the pre-trained autoencoder. The initial learning rate lr= 0.001, batchsize = 256, then we used Adam optimizer to train 300 epochs. In hyperparameter search, the GridSearch method is applied to determine and report the best result {0.1, 0.01, 1, 0.1} of the hyperparameter α, β, γ and ε. We run more than 20 times on all datasets and report the average results to ensure data accuracy. All experiments are conducted on NVIDIA Geforce GTX 1080Ti GPU(11G).

Acknowledgements

We would like to thank Ms Huichun Zhu and Mr Cheng Guo for fruitful discussions and Mr Xin Hu for assistance in Python implementation.

Funding

National Natural Science Foundation of China (62172088, 61772128, 61772367); National Key Research and Development Program of China (2016YFC0901704); Shanghai Natural Science Foundation (21ZR1400400, 19ZR1402000).

References

1. 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.
2. Villani AC, Satija R, Reynolds G, *et al.* Single-cell rna-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* 2017;**356**(6335):870.
3. Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell rna-seq data. *Nat Rev Genet* 2019;**20**(5):273–82.
4. Becht E, McInnes L, Healy J, *et al.* Dimensionality reduction for visualizing single-cell data using umap. *Nat Biotechnol* 2019;**37**(1):38–44.
5. Wang B, Ramazzotti D, De Sano L, *et al.* Simlr: A tool for large-scale genomic analyses by multi-kernel learning. *Proteomics* 2018;**18**(2):1700232.
6. Lin P, Troup M, Ho JWK. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biol* 2017;**18**(1):1–11.
7. Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis. In: *International conference on machine learning*. PMLR, 2016, 478–87.
8. Guo X, Gao L, Liu X, *et al.* Improved deep embedded clustering with local structure preservation. In: *Ijcai*, 2017, 1753–9.
9. Eraslan G, Simon LM, Mircea M, *et al.* Single-cell rna-seq denoising using a deep count autoencoder. *Nat Commun* 2019;**10**(1):1–14.
10. Tian T, Wan J, Song Q, *et al.* Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence* 2019;**1**(4):191–8.
11. Grønbech CH, Vording MF, Timshel PN, *et al.* scvae: Variational auto-encoders for single-cell gene expression data. *Bioinformatics* 2020;**36**(16):4415–22.
12. Karim MR, Beyan O, Zappa A, *et al.* Deep learning-based clustering approaches for bioinformatics. *Brief Bioinform* 2021;**22**(1):393–415.
13. Ji P, Zhang T, Li H, *et al.* Deep subspace clustering networks arXiv preprint arXiv:1709.02508. 2017.
14. Park S, Zhao H. Spectral clustering based on learning similarity matrix. *Bioinformatics* 2018;**34**(12):2069–76.
15. Gan Y, Li N, Zou G, *et al.* Identification of cancer subtypes from single-cell rna-seq data using a consensus clustering method. *BMC Med Genomics* 2018;**11**(6):65–72.
16. Bo D, Wang X, Shi C, *et al.* Structural deep clustering network. In: *Proceedings of The Web Conference 2020*, 2020, 1400–10.
17. Tian T, Wan J, Song Q, *et al.* Clustering single-cell rna-seq data with a model-based deep learning approach. *Nature Machine Intelligence* 2019;**1**(4):191–8.
18. Li J, Jiang W, Han H, *et al.* Scgslc: An unsupervised graph similarity learning framework for single-cell rna-seq data clustering. *Comput Biol Chem* 2021;**90**:107415.
19. Wang J, Ma A, Chang Y, *et al.* scgnn is a novel graph neural network framework for single-cell rna-seq analyses. *Nat Commun* 2021;**12**(1):1–11.
20. Kodinariya TM, Makwana PR. Review on determining number of cluster in k-means clustering. *International Journal* 2013;**1**(6):90–5.
21. Huang M, Wang J, Torre E, *et al.* Saver: gene expression recovery for single-cell rna sequencing. *Nat Methods* 2018;**15**(7):539–42.
22. Miao Z, Deng K, Wang X, *et al.* Desingle for detecting three types of differential expression in single-cell rna-seq data. *Bioinformatics* 2018;**34**(18):3223–4.
23. Svensson V. Droplet scrna-seq is not zero-inflated. *Nat Biotechnol* 2020;**38**(2):147–50.
24. Baran Y, Bercovich A, Sebe-Pedros A, *et al.* Metacell: analysis of single-cell rna-seq data using k-nn graph partitions. *Genome Biol* 2019;**20**(1):1–19.
25. Dann E, Henderson NC, Teichmann SA, *et al.* Differential abundance testing on single-cell data using k-nearest neighbor graphs. *Nat Biotechnol* 2021;1–9.
26. Strichartz RS. Eb davies, heat kernels and spectral theory. *Bulletin (New Series) of the American Mathematical Society* 1990;**23**(1):222–7.
27. Han X, Wang R, Zhou Y, *et al.* Mapping the mouse cell atlas by microwell-seq. *Cell* 2018;**172**(5):1091–107.
28. Cao J, Packer JS, Ramani V, *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* 2017;**357**(6352):661–7.
29. Petropoulos S, Edsgärd D, Reinius B, *et al.* Single-cell rna-seq reveals lineage and x chromosome dynamics in human preimplantation embryos. *Cell* 2016;**165**(4):1012–26.
30. Amit I, Garber M, Chevrier N, *et al.* Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science* 2009;**326**(5950):257–63.

31. Tirosh I, Izar B, Prakadan SM, *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science* 2016;**352**(6282):189–96.
32. Zheng GXY, Terry JM, Belgrader P, *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017;**8**(1): 1–12.
33. Alexander Wolf F, Angerer P, Theis FJ. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biol* 2018;**19**(1): 1–5.
34. 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.
35. Rand WM. Objective criteria for the evaluation of clustering methods. *J Am Stat Assoc* 1971;**66**(336):846–50.
36. Wang Y, Shi Z, Guo X, *et al.* Deep embedding for determining the number of clusters. In: *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
37. Han X, Wang R, Zhou Y, *et al.* Mapping the mouse cell atlas by microwell-seq. *Cell* 2018;**172**(5):1091–107.
38. Cao J, Packer JS, Ramani V, *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* 2017;**357**(6352):661–7.
39. Amit I, Garber M, Chevrier N, *et al.* Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science* 2009;**326**(5950):257–63.
40. Tirosh I, Izar B, Prakadan SM, *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science* 2016;**352**(6282):189–96.
41. Petropoulos S, Edsgärd D, Reinius B, *et al.* Single-cell rna-seq reveals lineage and x chromosome dynamics in human preimplantation embryos. *Cell* 2016;**165**(4):1012–26.
42. Zheng GXY, Terry JM, Belgrader P, *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017;**8**(1): 1–12.