ZINB-Based Graph Embedding Autoencoder for Single-Cell RNA-Seq Interpretations

Zhuohan Yu[1]*, Yifu Lu[1]*, Yunhe Wang[2], Fan Tang[1], Ka-Chun Wong[3], Xiangtao Li[1]†

[1] School of Artificial Intelligence, Jilin University, Jilin, China
[2] School of Artificial Intelligence, Hebei University of Technology, Tianjin, China
[3] Department of Computer Science, City University of Hong Kong, Hong Kong SAR
zhuohan20@mails.jlu.edu.cn, luyf20@mails.jlu.edu.cn, wangyh082@hebut.edu.cn,
tangfan@jlu.edu.cn, kc.w@cityu.edu.hk, lixt314@jlu.edu.cn

Abstract

Single-cell RNA sequencing (scRNA-seq) provides high-throughput information about the genome-wide gene expression levels at the single-cell resolution, bringing a precise understanding on the transcriptome of individual cells. Unfortunately, the rapidly growing scRNA-seq data and the prevalence of dropout events pose substantial challenges for cell type annotation. Here, we propose a single-cell model-based deep graph embedding clustering (scTAG) method, which simultaneously learns cell–cell topology representations and identifies cell clusters based on deep graph convolutional network. scTAG integrates the zero-inflated negative binomial (ZINB) model into a topology adaptive graph convolutional autoencoder to learn the low-dimensional latent representation and adopts Kullback–Leibler (KL) divergence for the clustering tasks. By simultaneously optimizing the clustering loss, ZINB loss, and the cell graph reconstruction loss, scTAG jointly optimizes cluster label assignment and feature learning with the topological structures preserved in an end-to-end manner. Extensive experiments on 16 single-cell RNA-seq datasets from diverse yet representative single-cell sequencing platforms demonstrate the superiority of scTAG over various state-of-the-art clustering methods.

Introduction

Single-cell RNA-sequencing (scRNA-seq) techniques enable elucidating the genetic heterogeneity of individual cells, which is essential for characterizing cell types based on the transcriptome (Kolodziejczyk et al. 2015), studying developmental biology (Chowdhury 2021), discovering complex diseases (Costa et al. 2013), and inferring cell trajectories (Tran and Bader 2020). Therefore, accurate identification of cell types has become a key step in single-cell RNA-seq analysis (Macosko et al. 2015). Clustering has been proven to be the most effective method for cell type annotation, as it can identify cell types in an unbiased manner (Kiselev, Andrews, and Hemberg 2019). In early research, traditional clustering methods such as K-means (MacQueen et al. 1967), hierarchical clustering (Johnson 1967) and density-based clustering (Kriegel et al. 2011) have been applied to address clustering tasks. However, clustering analysis of scRNA-seq data remains a statistical and computational challenge, owing to the high heterogeneity of genome coverage and some technical limitations rendering scRNA-seq data very sparse and having a large number of zero elements (Angerer et al. 2017; Grün, Kester, and Van Oudenaarden 2014). Therefore, it is imperative to develop effective computational methods to unleash the full potential of scRNA-seq.

Several clustering methods have been developed to address these limitations. Most studies use sophisticated techniques that involve iterative clustering, for instance, CIDR is a fast PCA-based algorithm for imputation and clustering based on a dissimilarity matrix (Lin, Troup, and Ho 2017). SC3 proposes a consensus-clustering framework for single-cell RNA-seq data, which reduces gene dimensions using PCA and Laplacian transformation (Kiselev et al. 2017). SIMLR uses multi-kernel learning to find a more robust distance metric and to address the high levels of dropout events (Wang et al. 2017). However, these computational methods usually tend to provide suboptimal results on scRNA-seq data because of the extreme sparsity caused by lack of gene expression levels. Moreover, most of them rely on full graph Laplacian matrices, which have high computational and storage costs.

In recent years, deep embedding clustering approaches have successfully developed to model the high-dimensional and sparse scRNA-seq data; such as, scDeepcluster (Tian et al. 2019), scDCC (Tian et al. 2021), scziDesk (Chen et al. 2020), scDHA (Tran et al. 2021), and DCA (Eraslan et al. 2019). They can iteratively refine clusters by learning highly confident assignments using an auxiliary target distribution to achieve better clustering results. However, these deep embedding clustering methods often ignore the structural information propagation and node relationships. Recently, emerging graph neural networks (GNNs) have been demonstrated to naturally capture graph structure information propagated through neighbor information (Zeng et al. 2020). Graph embedding clustering often combines deep autoencoder and graph clustering algorithms, which can learn

*These authors contributed equally.
†Corresponding author.

the latent compact representation to explore both the rich content and structural information (Nie, Zhu, and Li 2017).

Motivated by the above observations, we propose here a single-cell model-based deep graph embedding clustering named scTAG (Fig. 1), which simultaneously learns cell–cell topology representations and identifies cell clusters from an autoencoder (Du et al. 2017). We first utilize the zero-inflated negative binomial model (ZINB) to capture the global probabilistic structure of the data, by learning three characteristic distribution parameters including mean, dispersion and dropout probability. Then, scTAG proposes a ZINB-based graph convolutional autoencoder to preserve the topological structure of the cells in the low-dimensional latent space. After that, Kullback–Leibler (KL) divergence is used to optimize the clustering process. Finally, scTAG can combine three kinds of training loss, including the clustering loss, the ZINB loss and the cell graph reconstruction loss to optimize the cell cluster label assignment and to learn the cell–cell topology representations, generating superior clustering results. The main contributions of our work are summarized below:

- We propose a single-cell model-based deep graph embedding clustering called scTAG, which integrates the zero-inflated negative binomial model into a topology adaptive graph convolutional autoencoder to capture the global probabilistic structure of data.
- scTAG constructs a cell graph and uses a topology adaptive graph convolutional autoencoder to collectively preserve the topological structural information and the cell-to-cell relationships in scRNA-seq data.
- To the best of our knowledge, this is the first article to incorporate ZINB into a graph convolutional autoencoder to model highly-sparse and overdispersed scRNA-seq data.
- We evaluate our model alongside state-of-the-art competitive methods on 16 real scRNA-seq datasets. The results demonstrate that scTAG outperforms all of the baseline methods.

Related Work

Deep Clustering Methods

In the past years, deep learning methods have been used, advanced to analyze scRNA-seq data, due to their strong learning ability and adaptability. Eraslan *et al.* proposed a deep count autoencoder network named DCA, to denoise the original scRNA-seq data, which takes the count distribution, overdispersion and sparsity of the data into account (Eraslan et al. 2019). Deng *et al.* developed scScope, which introduces a self-correcting layer to perform imputations on zero-values of scRNA-seq data in an iterative way (Deng et al. 2018). Grønbech *et al.* proposed a novel variational auto-encoder-based method named scVAE, which follows the Gaussian mixture distribution and then estimates the loss by sampling from the distribution (Grønbech et al. 2020). Tian *et al.* developed a single-cell model-based deep embedded clustering method (scDCC), which integrates the ZINB model with clustering loss and constraint loss (Tian et al. 2021). Chen *et al.* proposed scziDesk, which combines the deep learning method with a denoising autoencoder to characterize scRNA-seq data and a soft self-training K-means algorithm to cluster the cells under a learned latent space (Chen et al. 2020). However, these deep neural networks hardly preserve the topological structure of scRNA-seq data, thereby ignoring the cell-to-cell relationships in the process of analysis.

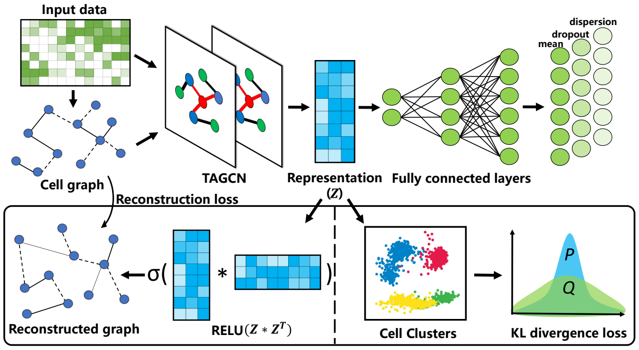

Figure 1: The model architecture of scTAG. scTAG integrates the zero-inflated negative binomial (ZINB) model into a topology adaptive graph convolutional autoencoder to learn the low-dimensional latent representation and adopts Kullback–Leibler (KL) divergence for the clustering tasks.

Deep Graph Clustering Methods

The emergence of the deep graph autoencoder can greatly alleviate scRNA-seq data (Kipf and Welling 2016); compared to other autoencoders, it can learn the underlying low-dimensional representation by providing a global view of the whole graph. Zeng *et al.* proposed a new scRNA-seq data clustering method (GraphSCC), which accounts for structural cell-cell relationships through a graph convolutional network. Then, the representation learned from the network is optimized by a self-supervised module (Zeng et al. 2020). Wang *et al.* introduced scGNN in a hypothesis-free deep learning framework. The framework aggregates the cell–cell relationships using graph neural networks and applies the left-truncated Gaussian mixture model to learn heterogeneous gene expression patterns (Wang et al. 2021). Luo *et al.* proposed a single-cell model based on a graph autoencoder (scGAE), which builds a cell graph and uses the graph autoencoder to preserve the feature and topological structure information of scRNA-seq data (Luo et al. 2021). Rao *et al.* developed an imputation method (GraphSCI) to impute the dropout events in scRNA-seq data by incorporating graph convolutional and autoencoder neural networks (Rao et al. 2021).

Methods

Data Pre-processing

We take the scRNA-seq gene expression matrix X as input, where X_{ij} denotes the expression count of the jth gene ($1 \leq j \leq O$) in the ith cell ($1 \leq i \leq N$). The first step is to filter out genes that are expressed as non-zero in more than 1% of the cells, as well as genes that are not expressed. Considering that the data in the count matrix is discrete and the size factor varies greatly, the normalization is defined as follows:

$$N(X_{ij}) = \ln\left(m(X)\frac{X_{ij}}{\sum_o X_{io}}\right) \tag{1}$$

where $m(X)$ represents the median of the total expression values of the cells. According to Eq. (1), the discrete data is smoothed and is rescaled by natural log transformation. After normalization, we select the first t highly-variable genes based on the ranking of the normalized dispersion values calculated by scanpy package (Wolf, Angerer, and Theis 2018) to identify genes with high-level information.

Cell Graph

In this study, we use the embedding learned from the graph autoencoder to preserve the relationship and neighbor information between the cells. Similar to previous works (Wang et al. 2021), KNN algorithm is employed to construct the cell graph and each node in the graph represents a cell. Indeed, there exits nodes a and b and an edge between a and b; if a is b's neighbor within the k shortest distance, k is set to 15. Euclidean distance is calculated to describe the correlation between the nodes to discover the k shortest distance. After that, the constructed cell graph is an undirected graph and the weight of the edge is uniformly set as 1.

Topology Adaptive Graph Convolutional Autoencoder

To capture the graph structure and node relationships, we developed a variant of the graph convolution autoencoder that uses topology adaptive graph convolutional network (TAGCN) (Du et al. 2017) as the graph encoder. The idea is that TAGCN uses K graph convolution kernels at each layer to extract local features of different sizes, which avoids the drawback of approximate convolution kernels not fully extracting the graph information and thus enhances the learning ability of the model for scRNA-seq data.

The gene expression matrix X and normalized adjacency matrix A are used as inputs. A can be represented as $A = D^{-\frac{1}{2}}(I + \tilde{A})D^{-\frac{1}{2}}$, where $\tilde{A}$ is the adjacency matrix, and $D = diag\{(I + \tilde{A})\mathbf{1}_N\}$ is the degree matrix, where N is the total number of samples and $\mathbf{1}_N$ denotes the N-dimensional vector consisting entirely of one. Considering the l-th hidden layer, it is assumed that each node has C_l features after feature mapping at this time, which means that the input data of the l-th hidden layer is $x_c^{(l)} \in \mathbf{R}^N$, where $c = 1, 2, \ldots, C_l$. The graph convolution process can be defined as follows:

$$y_f^{(l)} = \sum_{c=1}^{C_l} G_{c,f}^{(l)} x_c^{(l)} + b_f 1_N \tag{2}$$

where $y_f^{(l)}$ represents f-th output feature map; b_f is a learnable bias; $G_{c,f}^{(l)}$ represents the polynomial convolution kernel in TAGCN, and its internal architecture uses K graph convolution kernels to extract local features of different sizes, which is defined as:

$$G_{c,f}^{(l)} = \sum_{k=0}^{K} g_{c,f,k}^{(l)} A^k \tag{3}$$

where $g_{c,f,k}^{(l)}$ denotes the polynomial coefficients. Normalized adjacency matrix A is adopted to enable a more stable computation of the entire convolution operation. After each graph convolution operation, a nonlinear operation is applied to the output, as defined below:

$$x_f^{(l+1)} = \sigma(y_f^{(l)}) \tag{4}$$

where $\sigma(\cdot) = \max(0, x)$ denotes a RELU activation function.

Since most of the structure and information of scRNA-seq data X is preserved in the latent embedded representation Z through the TAGCN encoder, the decoder of the graph autoencoder can be defined as the inner product between the latent embedding:

$$Z = f_E(X) \tag{5}$$

$$A_r = \sigma(Z^T Z) \tag{6}$$

where, f_E represents the TAGCN encoder function; A_r is the reconstructed adjacency matrix. Therefore, the reconstruction loss of A and A_r should be minimized in the learning process as below:

$$L_r = ||A - A_r||_2^2 \tag{7}$$

ZINB-based Graph Convolutional Autoencoder

To better capture the structure of single-cell RNA sequencing data by decoding from the latent embedded representation Z, we integrate the zero-inflated negative binomial (ZINB) model into a topology adaptive graph convolutional autoencoder to capture the global probabilistic structure of the data. In the following, we first analyze the reasons for approximating the scRNA-seq data distribution using the zero-inflated negative binomial distribution (ZINB) under the previous studies (Risso et al. 2018; Miao et al. 2018).

Theorem 1 *The data distribution of the scRNA-seq gene expression count matrix can be approximated by zero-inflated negative binomial distribution (ZINB).*

Proof. The data distribution of the single cell RNA sequencing gene expression count matrix generally conforms to three characteristics: 1) discrete; 2) variance greater than the mean; 3) contains many zero values, including non-expressed genes (true zero) or due to technical reasons (dropout zero). In the following, we prove that the ZINB distribution can simulate these three properties. ZINB is defined as follows:

$$f_{\text{ZINB}}(x|\pi, r, p) = \pi I_0(x) + (1 - \pi) f_{\text{NB}}(x|r, p) \tag{8}$$

$$f_{\text{NB}}(x|r, p) = \begin{pmatrix} x + r - 1 \\ x \end{pmatrix} p^r (1 - p)^x \tag{9}$$

where π denotes the proportion of zero values; $I_0(\mathrm{x})$ is an indicator function, which equals 1 when $x = 0$, and 0 otherwise; r and p are the parameters of the negative binomial (NB) distribution, representing the success times and probability, respectively. Since NB distribution belongs to a discrete distribution, ZINB also satisfies the discrete distribution property. When $x = 0$, ZINB can predict the probability of the dropout zero (dropout rate) by π, which is deduced as follows:

$$d = \frac{(1-\pi) f_{\mathrm{NB}}(0)}{\pi + (1-\pi) f_{\mathrm{NB}}(0)} \tag{10}$$

Meanwhile, we can prove that the variance is greater than the mean following the NB distribution. Assuming that the mean is $E(x)$, which is defined as:

$$E(x) = \sum_{x=0}^{\infty} x \begin{pmatrix} x+r-1 \\ x \end{pmatrix} p^r (1-p)^x \tag{11}$$

Let $x' = x - 1$, $r' = r + 1$, then we have:

$$E(x) = \frac{r(1-p)}{p} \sum_{x'=0}^{\infty} f_{\mathrm{NB}}(x'|r', p) \tag{12}$$

Since NB is a discrete distribution, the sum of all probabilities equals 1; that is, $\sum_{x'=0}^{\infty} f_{\mathrm{NB}}(x'|r', p) = 1$. Therefore, $E(x) = \frac{r(1-p)}{p}$. Assuming that the variance is $Var(x)$, which can be defined as:

$$Var(x) = E(x^2) - E(x)^2 = \frac{r(1-p)}{p^2} \tag{13}$$

Then, we can get the relationship between $E(x)$ and $Var(x)$:

$$Var(x) = E(x) + \frac{E(x)^2}{r} \tag{14}$$

Since $r > 0$, $Var(x) > E(x)$.

On this basis, we propose to apply the ZINB distribution model to simulate the data distribution to capture the characters of scRNA-seq data. Then, the ZINB-based Graph Convolutional Autoencoder instead of a regular graph autoencoder, which is trained to attempt to reconstruct its input is defined as follows:

$$\mathrm{NB}(X|\mu, \theta) = \frac{\Gamma(X+\theta)}{X!\Gamma(\theta)} \left(\frac{\theta}{\theta+\mu}\right)^{\theta} \left(\frac{\mu}{\theta+\mu}\right)^{X} \tag{15}$$

$$\mathrm{ZINB}(X|\pi, \mu, \theta) = \pi\delta_0(X) + (1-\pi)\mathrm{NB}(X) \tag{16}$$

where μ and θ represent the mean and dispersion, respectively; π is the weight of the point mass at zero. The proportion $\frac{\theta}{\theta+\mu}$ replaces the probability p in Eq. (9). After that, we append three fully connected layers to estimate the parameters $\{\pi,\mu,\theta\}$ in the latent embedded representation Z as follows:

$$\Pi = \mathrm{sigmoid}(W_{\pi} f_D(Z)) \tag{17}$$

$$M = \exp(W_{\mu} f_D(Z)) \tag{18}$$

$$\Theta = \exp(W_{\theta} f_D(Z)) \tag{19}$$

where f_D is a three-layers fully connected neural network with hidden layers of 128, 256 and 512 nodes; W represents the learned weights of the loss functions; Π, M and Θ are all parameter matrices, representing the dropout probability, mean and dispersion of the network output, respectively. The selection of the activation function depends on the range and definition of the parameters. Dropout probability is between 0 and 1, so the sigmoid function is chosen. In addition, due to the non-negative value of the mean and dispersion, we apply the exponential function. The negative log likelihood of the ZINB distribution can be used as the reconstruction loss function of the original data X, which can be defined as below:

$$L_{\mathrm{ZINB}} = -\log(\mathrm{ZINB}(X|\pi, \mu, \theta)) \tag{20}$$

Self-optimizing Deep Graph Embedded Clustering

Since the deep graph embedded clustering method is unsupervised and not guided by labels, it is unable to get a good optimized feedback during the training process. Therefore, we apply a self-optimizing embedding algorithm that inputs latent embedding into a self-optimizing clustering module. The objective takes the form of Kullback–Leibler (KL) divergence and is formulated as follows:

$$L_c = KL(P||Q) = \sum_i \sum_u p_{iu} \log \frac{p_{iu}}{q_{iu}} \tag{21}$$

where q_{iu} is the soft label of the embedding node z_i. This label measures the similarity between z_i and the cluster central embedding μ_u by a Student's t-distribution, which can be described as follows:

$$q_{iu} = \frac{(1 + ||z_i - \mu_u||^2)^{-1}}{\sum_r (1 + ||z_i - \mu_r||^2)^{-1}} \tag{22}$$

It is worth noting that the initial cluster centers $\{\mu\}$ are generated by spectral clustering through the adjacency matrix after pre-training of the ZINB-based graph convolutional autoencoder. In addition, p_{iu} is the auxiliary target distribution, which puts more emphasis on the similar data points assigned with high confidence on the basis of q_{iu}, as below:

$$p_{iu} = \frac{q_{iu}^2 / \sum_i q_{iu}}{\sum_r (q_{ir}^2 / \sum_i q_{ir})} \tag{23}$$

Since the target distribution P is defined based on Q, the embedding learning of Q is supervised in a self-optimizing way to enable it to be close to the target distribution P.

Joint Embedding and Clustering Optimization

In the whole training process, graph autoencoder embedding and clustering learning are optimized jointly. We minimize the following total objective function:

$$L = \gamma_1 L_r + \gamma_2 L_{\mathrm{ZINB}} + \gamma_3 L_c \tag{24}$$

where L_r and L_{ZINB} are the reconstruction loss and ZINB loss in the graph autoencoder respectively; L_c is the embedding clustering loss; γ_1, γ_2 and γ_3 are weight coefficients assigned to each loss to control the balance of the total objective function. In the experiment, $\{\gamma_1, \gamma_2, \gamma_3\}$ are set to $\{0.3,1.0,1.5\}$. We combine stochastic gradient descent (SGD) and back propagation to jointly optimize graph autoencoder and cluster centers. The gradient of clustering loss L_c with respect to the latent embedding node z_i and cluster center μ_u can be calculated as:

$$\frac{\partial L_c}{\partial z_i} = 2 \sum_u (1 + ||z_i - \mu_u||^2)^{-1} (p_{iu} - q_{iu})(z_i - \mu_u) \tag{25}$$

$$\frac{\partial L_c}{\partial \mu_u} = -2 \sum_i (1 + ||z_i - \mu_u||^2)^{-1} (p_{iu} - q_{iu})(z_i - \mu_u) \tag{26}$$

Given the learning rate is l_r, the cluster center μ_u can be updated as follows:

$$\mu_u = \mu_u - \frac{l_r}{N} \sum_{i=1}^{N} \frac{\partial L_c}{\partial \mu_u} \tag{27}$$

Dataset	Cell	Gene	Class	Platform	Reference
Yan	90	20214	6	Tang	(Yan et al. 2013)
Camp(Brain)	734	18927	6	SMARTer	(Camp et al. 2015)
Camp(Liver)	777	19020	7	SMARTer	(Camp et al. 2017)
QS_Diaphragm	870	23341	5	Smart-seq2	(Schaum et al. 2018)
QS_Limb_Muscle	1090	23341	6	Smart-seq2	(Schaum et al. 2018)
QS_Lung	1676	23341	11	Smart-seq2	(Schaum et al. 2018)
Muraro	2122	19046	9	CEL-seq2	(Muraro et al. 2016)
Adam	3660	23797	8	Drop-seq	(Adam et al. 2017)
Qx_Limb_Muscle	3909	23341	6	10x	(Schaum et al. 2018)
QS_Heart	4365	23341	8	Smart-seq2	(Schaum et al. 2018)
Young	5685	33658	11	10x	(Young et al. 2018)
Plasschaert	6977	28205	8	inDrop	(Plasschaert et al. 2018)
Wang_Lung	9519	14561	2	10x	(Wang et al. 2018)
Qx_Trachea	11269	23341	5	10x	(Schaum et al. 2018)
Tosches_turtle	18664	23500	15	Drop-seq	(Tosches et al. 2018)
Bach	23184	19965	8	10x	(Bach et al. 2017)

Table 1: Summary of the real scRNA-seq datasets

where N denotes the total number of nodes. The polynomial coefficient matrix W_e of the convolution kernel in encoder and the weight matrix W_d in decoder are updated as follows:

$$W_e = W_e - \frac{l_r}{N}\left(\gamma_1 \frac{\partial L_r}{\partial W_e} + \gamma_2 \frac{\partial L_{\mathrm{ZINB}}}{\partial W_e} + \gamma_3 \frac{\partial L_c}{\partial W_e}\right) \tag{28}$$

$$W_d = W_d - \frac{l_r}{N}\left(\gamma_1 \frac{\partial L_r}{\partial W_d} + \gamma_2 \frac{\partial L_{\mathrm{ZINB}}}{\partial W_d}\right) \tag{29}$$

When the maximum number of iterations is reached, the optimization process stops. Then, we can obtain the predictive clustering assignment of each cell through Q after model training.

Experiments

Data Sources

We compared the performance of our model with other baseline methods on 16 real-world scRNA-seq datasets from several representative sequencing platforms. The 16 scRNA-seq datasets used in our experiments are collected from recently published papers about scRNA-seq experiments and the detailed information is described in Table 1. All 16 datasets are from different species, including mouse and human, as well as from different organs, such as brain, lung, and kidney. Specifically, the numbers of cells range from 90 to 23184, and genes range from 14561 to 33658.

Baseline

The performance of scTAG was compared with two base clustering methods including a K-means clustering algorithm and a spectral clustering algorithm, and several state-of-the-art scRNA-seq data clustering methods including four single-cell based deep embedded clustering methods and three single-cell deep graph embedded clustering methods.

- Deep soft K-means clustering for scRNA-seq data (**scziDesk**) (Chen et al. 2020): It incorporates deep learning method with a denoising autoencoder to characterize scRNA-seq data in a latent space.
- Model-based deep embedding method (**scDCC**) (Tian et al. 2021): It is a principle clustering method and applies domain knowledge to the clustering step, which adds prior knowledge to the loss function.
- Deep count autoencoder network (**DCA**) (Eraslan et al. 2019): It takes the sparsity, count distribution, and overdispersion of the original data into account using a negative binomial noise model.
- Deep embedded clustering (**DEC**) (Xie, Girshick, and Farhadi 2016): It applies deep neural networks to simultaneously learn cluster assignments and feature representations in a lower-dimensional feature space.
- Single-cell graph neural network (**scGNN**) (Wang et al. 2021): It gives a framework that aggregates cell–cell relationships using graph neural networks and applies a Gaussian mixture model to learn gene expression patterns.
- Single-cell graph autoencoder (**scGAE**) (Luo et al. 2021): It builds a cell graph and adopts a multitask-oriented graph autoencoder to maintain the structural information of scRNA-seq data.
- GCN-based single-cell clustering (**GraphSCC**) (Zeng et al. 2020): It accounts for the structural relations between cells using a graph convolutional network, and the learned representation is optimized by a dual self-supervised module.

Implementation Details

In the proposed scTAG method, the cell graph was constructed using KNN algorithm with the nearest neighbor parameter k=15 to build the cell graph. In the graph autoencoder, TAG was set as two layers of 128 and 15 nodes, and the nodes of three hidden layers in the fully connected decoder set at 128, 256, and 512. Our algorithm consists of pre-training and formal training, in which pre-training, epochs were set at 1000, while in formal training, epochs were set at 300. Our model was optimized using Adam algorithm with the learning rate 5e-4 in pre-training and 1e-4 in formal training (Kingma and Ba 2015). For baseline methods, the parameters were set the same as in the original papers. We conducted our experiments on a Ubuntu server with NVIDIA GTX 2080Ti GPU with 24 GB memory size. The initial weights and bias used the default settings of Tensorflow.

Clustering Performance

Two widely-used clustering metrics including normalized mutual information (NMI) and adjusted rand index (ARI) were employed to measure the performance of our method and the other nine baseline methods. The higher the value of the metrics, the better clustering performance.

Table 2 summarizes the clustering performance of scTAG and the baseline methods on 16 scRNA-seq datasets. Each clustering method was run ten times to take the average, and the values highlighted represent the best results. Obviously, our method outperforms other the baseline clustering methods for clustering performance. For the 16 scRNA-seq datasets, scTAG achieved the best NMI and ARI on 13 and

	Datasets	Ours	Deep Graph Embedded Methods			Deep Embedded Methods				Base Methods	
		scTAG	scGNN	scGAE	GraphSCC	scziDesk	scDCC	DCA	DEC	K-means	Spectral
NMI	Yan	**0.8118**	0.7599	0.7389	0.6783	0.7656	0.8004	0.8050	0.6750	0.7632	0.7475
	Camp(Brain)	**0.5607**	0.3579	0.5201	0.4347	0.5232	0.4677	0.4609	0.4331	0.4428	0.5540
	Camp(Liver)	0.7767	0.7497	**0.8109**	0.6688	0.7343	0.7488	0.6918	0.6282	0.7878	0.7858
	QS_Diaphragm	**0.9346**	0.7608	0.7351	0.8966	0.9210	0.8223	0.9174	0.8815	0.8846	0.8881
	QS_Limb_Muscle	**0.9616**	0.7726	0.7398	0.7009	0.9468	0.4624	0.7691	0.9257	0.8911	0.9389
	QS_Lung	**0.8038**	0.6642	0.6766	0.6824	0.7543	0.4982	0.6400	0.7285	0.7785	0.7976
	Muraro	**0.8399**	0.6294	0.7619	0.5723	0.7349	0.8347	0.7865	0.7449	0.8194	0.8291
	Adam	**0.8931**	0.2731	0.6784	0.5606	0.8509	0.7494	0.5119	0.7470	0.7201	0.8473
	Qx_Limb_Muscle	**0.9481**	0.7457	0.7569	0.6501	0.9131	0.8774	0.8152	0.7645	0.7715	0.8652
	QS_Heart	**0.8857**	0.6540	0.6039	0.8350	0.8723	0.4242	0.7854	0.8216	0.8299	0.8454
	Young	**0.7968**	0.4145	0.6536	0.6047	0.7394	0.5575	0.5339	0.6420	0.7533	0.7787
	Plasschaert	0.7749	0.5856	0.5563	0.7489	0.7899	0.5786	0.6466	0.6433	**0.8642**	0.5216
	Wang_Lung	0.8210	0.3975	0.3150	0.0270	0.7965	0.0482	**0.8929**	0.8455	0.8917	0.8682
	Qx_Trachea	**0.7966**	0.3587	0.4868	0.7360	0.7341	0.6728	0.5591	0.5658	0.6969	0.7725
	Tosches_turtle	**0.7286**	0.5427	-	0.6937	0.6082	0.7151	0.5947	0.6617	0.7047	0.7127
	Bach	**0.8635**	0.7430	-	0.6950	0.8343	0.8214	0.8372	0.6259	0.7706	0.8342
ARI	Yan	**0.7478**	0.6945	0.6563	0.4599	0.5628	0.7344	0.7189	0.4809	0.6707	0.6502
	Camp(Brain)	**0.4334**	0.3041	0.4152	0.2498	0.4151	0.4263	0.3366	0.2551	0.2763	0.4291
	Camp(Liver)	0.6087	0.5806	**0.6872**	0.4315	0.6005	0.5587	0.5117	0.4215	0.6465	0.6190
	QS_Diaphragm	**0.9628**	0.5646	0.5638	0.9619	0.9517	0.8895	0.9165	0.9372	0.9110	0.9170
	QS_Limb_Muscle	**0.9813**	0.6399	0.5419	0.5303	0.9743	0.3449	0.6567	0.9562	0.8922	0.9615
	QS_Lung	0.6526	0.3631	0.2797	0.5180	0.7401	0.2908	0.4429	0.5793	0.7329	**0.7559**
	Muraro	**0.8878**	0.5080	0.6413	0.3391	0.6784	0.7100	0.8300	0.7245	0.8452	0.8741
	Adam	**0.9108**	0.1608	0.5090	0.2444	0.8680	0.6576	0.3885	0.6903	0.5590	0.8284
	Qx_Limb_Muscle	**0.9581**	0.5899	0.4983	0.4534	0.9441	0.7964	0.7875	0.7346	0.6628	0.7431
	QS_Heart	**0.9371**	0.5222	0.2497	0.8908	0.9324	0.2584	0.7804	0.9025	0.8376	0.8757
	Young	**0.6928**	0.2588	0.5066	0.3795	0.6836	0.3702	0.3716	0.4785	0.6218	0.6543
	Plasschaert	0.8280	0.4272	0.3540	0.7965	0.8634	0.4668	0.4660	0.5978	**0.8999**	0.2916
	Wang_Lung	0.9004	0.1771	0.1035	0.0852	0.8975	0.0351	**0.9501**	0.9237	0.9426	0.9387
	Qx_Trachea	**0.9154**	0.1270	0.1790	0.8725	0.8085	0.4668	0.3145	0.4407	0.7638	0.8667
	Tosches_turtle	**0.6942**	0.5279	-	0.5555	0.6165	0.6150	0.3767	0.4619	0.6439	0.6783
	Bach	**0.9057**	0.6089	-	0.6546	0.8738	0.7549	0.8871	0.5393	0.7391	0.8622

Table 2: Performance of our method and the other baseline methods on 16 scRNA-seq datasets. The bold font indicates the best values among the compared methods.

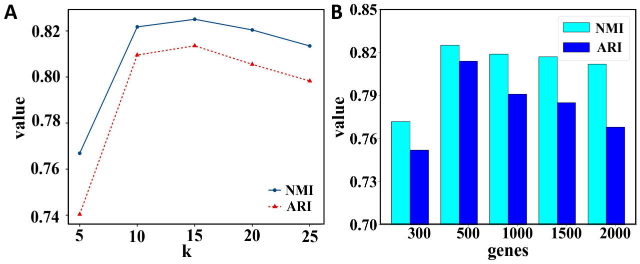

Figure 2: Parameter analysis. (A) Comparison of the average NMI and ARI values with different neighbor parameters, k. (B) Comparison of the average NMI and ARI values with different numbers of genes.

12 of them, respectively, and even in 'QS_Limb_Muscle', the NMI and ARI reached 0.9616 and 0.9813. Meanwhile, we can observe that the general deep graph embedded models have no advantage and the clustering performance is not stable. For example, they performs poorly on 'Wang_Lung'. The main reason is that the information structure preserved by the cell graph alone cannot address the particularities of scRNA-seq data well, and further simulation of the data by ZINB distribution is necessary, which again proves the superiority of scTAG. Furthermore, the clustering performance of deep embedded clustering with the ZINB distribution model including scziDesk and scDCC is better and stable. However, scTAG still has an advantage. This is because the TAGCN could effectively extract the key genes of the scRNA-seq data, so that the latent embedding representation in the model could retain the major information for clustering. For the base methods, we can see that the clustering performance of spectral clustering is generally better than that of K-means, because spectral clustering is based on the cell graph. Although the cell graph as described above does not fit well into the structure of scRNA-seq data, it is still an improvement over K-means clustering. This is also the reason why scTAG adopts spectral clustering to initialize the cluster center when optimizing the clustering loss. In summary, we can conclude that scTAG performs better than the other methods under two clustering evaluation metrics.

Parameter Analysis

Impact of the Neighbor Parameter k: k is the neighbor parameter of the KNN algorithm to construct the cell graph,

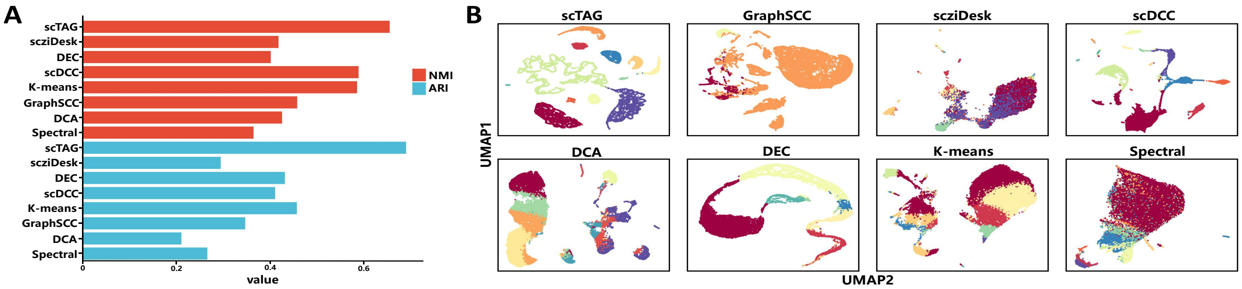

Figure 3: Clustering performance on the ‘Macosko’ dataset. (A) Comparison of NMI and ARI values between scTAG and baseline methods. (B) Comparison of clustering results with 2D visualization by UMAP.

and it determines the number of edges between each node and others. In order to investigate the impact of k, we ran our model with the parameters 5, 10, 15, 25 and 30. Fig. 2(A) shows the NMI and ARI values with different numbers of k. As depicted in Fig. 2(A), we observe that the two metrics first increase rapidly from parameter 5 to 10, and reach the best value at k =15, and then decrease slowly from parameter 15 to 25. Therefore, we set the neighbor parameter k as 15 in our scTAG model.

Different Numbers of Variable Genes Analysis: In single-cell data analysis, highly variable genes can provide more biological information and have more importance in determining the cell type. To explore the impact of the number of selected highly variable genes, we apply scTAG on real datasets with gene numbers from 300 to 2000. Fig. 2(B) shows the bar plot of the average NMI and ARI on the 16 datasets selecting 300, 500, 1000, 1500 and 2000 genes with high variability, respectively. It can be seen that the performance with 500 highly variable genes is the best, while the performance with 300 genes is much worse than the others. Therefore, to save computational resources and reduce running time, we set the number of selected high-variance genes in the model to 500.

Ablation Study

In this experiment, we analyzed the effect of each component of the scTAG method. Specifically, we ablated different components in three cases: 1) No TAG and ZINB-based decoder, with the general GCN and decoder used for replacement. The total loss function consists of reconstruction loss, clustering loss and MSE loss of output and raw data. 2) No TAG convolution operation. 3) No ZINB-based decoder. Table 3 tabulates the average NMI and ARI values on the 16 datasets for the three cases with scTAG. As shown in Table 3, it can be clearly observed that gene screening and extraction of scRNA-seq data via TAG convolution operation improves the clustering performance. Moreover, the ZINB has a significant impact on the final clustering results (ARI), indicating that the simulation of scRNA-seq data through ZINB distribution is necessary. In summary, all components of the scTAG method are reasonable and effective.

Methods	NMI	ARI
Without TAG&ZINB	0.7827	0.7372
Without TAG	0.8013	0.7758
Without ZINB	0.8077	0.7993
scTAG	**0.8252**	**0.8138**

Table 3: Ablation study measured by NMI and ARI values

Scalability of scTAG

To further demonstrate that scTAG can also be utilized to cluster large-scale data, we used a large mouse retina dataset called ‘Macosko’(Macosko et al. 2015), which contains a total of 44808 cells and 23288 genes grouped into 12 cell types. Fig. 3(A) summarizes the clustering peformance of scTAG and the baseline methods. Since scGNN and scGAE clustering methods failed to run on this large-scale dataset, they are not shown in the figure. It can be observed that scTAG maintains good clustering performance even on large-scale data, with the best NMI and ARI values. Furthermore, to illustrate the effectiveness of the latent embedding representation of scTAG and observe more intuitively the clustering effect on large-scale data, we applied UMAP (McInnes, Healy, and Melville 2018) to visualize the final embedding points of scTAG and the baseline methods in two-dimensional space as depicted in Fig. 3(B). It demonstrates that similar cells in the large-scale dataset can be well separated in the latent embedding representation of scTAG, and much better than with the other baseline methods.

Conclusion

In this paper, we propose a single-cell model-based deep graph embedding clustering called scTAG, which combines the ZINB model into a topology adaptive graph convolutional autoencoder for clustering scRNA-seq data. scTAG first extracts the key genes of the scRNA-seq data, preserving the cell–cell topological structure, and then performs graph reconstruction, decoding based on ZINB distribution and finally, self-optimized embedded clustering on the latent representation. Experimental results on 16 real scRNA-seq datasets indicate the superiority of the proposed scTAG method over other state-of-the-art baseline methods. In addition, the robustness and scalability analyse demonstrate that scTAG is effective, robust and scalable.

Acknowledgments

The work described in this paper was substantially supported by the National Natural Science Foundation of China under Grant No. 62076109, and "the Fundamental Research Funds for the Central Universities". The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region [CityU 11200218], one grant from the Health and Medical Research Fund, the Food and Health Bureau, The Government of the Hong Kong Special Administrative Region [07181426], and the funding from the Hong Kong Institute for Data Science (HKIDS) at the City University of Hong Kong. The work described in this paper was partially supported by two grants from the City University of Hong Kong (CityU 11202219, CityU 11203520). This research was substantially sponsored by the research project (Grant No. 32000464) supported by the National Natural Science Foundation of China and was substantially supported by the Shenzhen Research Institute, City University of Hong Kong.

References

Adam, M.; Potter, A. S.; Potter, S. S.; et al. 2017. Psychrophilic proteases dramatically reduce single-cell RNA-seq artifacts: a molecular atlas of kidney development. *Development*, 144(19): 3625–3632.

Angerer, P.; Simon, L.; Tritschler, S.; Wolf, F. A.; Fischer, D.; and Theis, F. J. 2017. Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology*, 4: 85–91.

Bach, K.; Pensa, S.; Grzelak, M.; Hadfield, J.; Adams, D. J.; Marioni, J. C.; and Khaled, W. T. 2017. Differentiation dynamics of mammary epithelial cells revealed by single-cell RNA sequencing. *Nature communications*, 8(1): 1–11.

Camp, J. G.; Badsha, F.; Florio, M.; Kanton, S.; Gerber, T.; Wilsch-Bräuninger, M.; Lewitus, E.; Sykes, A.; Hevers, W.; Lancaster, M.; et al. 2015. Human cerebral organoids recapitulate gene expression programs of fetal neocortex development. *Proceedings of the National Academy of Sciences*, 112(51): 15672–15677.

Camp, J. G.; Sekine, K.; Gerber, T.; Loeffler-Wirth, H.; Binder, H.; Gac, M.; Kanton, S.; Kageyama, J.; Damm, G.; Seehofer, D.; et al. 2017. Multilineage communication regulates human liver bud development from pluripotency. *Nature*, 546(7659): 533–538.

Chen, L.; Wang, W.; Zhai, Y.; and Deng, M. 2020. Deep soft K-means clustering with self-training for single-cell RNA sequence data. *NAR genomics and bioinformatics*, 2(2): lqaa039.

Chowdhury, H. A. 2021. Effective Clustering of scRNA-seq Data to Identify Biomarkers without User Input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15710–15711.

Costa, V.; Aprile, M.; Esposito, R.; and Ciccodicola, A. 2013. RNA-Seq and human complex diseases: recent accomplishments and future perspectives. *European Journal of Human Genetics*, 21(2): 134–142.

Deng, Y.; Bao, F.; Dai, Q.; Wu, L. F.; and Altschuler, S. J. 2018. Massive single-cell RNA-seq analysis and imputation via deep learning. *BioRxiv*, 315556.

Du, J.; Zhang, S.; Wu, G.; Moura, J. M.; and Kar, S. 2017. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*.

Eraslan, G.; Simon, L. M.; Mircea, M.; Mueller, N. S.; and Theis, F. J. 2019. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications*, 10(1): 1–14.

Grønbech, C. H.; Vording, M. F.; Timshel, P. N.; Sønderby, C. K.; Pers, T. H.; and Winther, O. 2020. scVAE: Variational auto-encoders for single-cell gene expression data. *Bioinformatics*, 36(16): 4415–4422.

Grün, D.; Kester, L.; and Van Oudenaarden, A. 2014. Validation of noise models for single-cell transcriptomics. *Nature methods*, 11(6): 637–640.

Johnson, S. C. 1967. Hierarchical clustering schemes. *Psychometrika*, 32(3): 241–254.

Kingma, D. P.; and Ba, J. L. 2015. Adam: A method for stochastic gradient descent. In *ICLR: International Conference on Learning Representations*, 1–15.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

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

Kiselev, V. Y.; Kirschner, K.; Schaub, M. T.; Andrews, T.; Yiu, A.; Chandra, T.; Natarajan, K. N.; Reik, W.; Barahona, M.; Green, A. R.; et al. 2017. SC3: consensus clustering of single-cell RNA-seq data. *Nature methods*, 14(5): 483–486.

Kolodziejczyk, A. A.; Kim, J. K.; Svensson, V.; Marioni, J. C.; and Teichmann, S. A. 2015. The technology and biology of single-cell RNA sequencing. *Molecular cell*, 58(4): 610–620.

Kriegel, H.-P.; Kröger, P.; Sander, J.; and Zimek, A. 2011. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3): 231–240.

Lin, P.; Troup, M.; and Ho, J. W. 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome biology*, 18(1): 1–11.

Luo, Z.; Xu, C.; Zhang, Z.; and Jin, W. 2021. scGAE: topology-preserving dimensionality reduction for single-cell RNA-seq data using graph autoencoder. *bioRxiv*.

Macosko, E. Z.; Basu, A.; Satija, R.; Nemesh, J.; Shekhar, K.; Goldman, M.; Tirosh, I.; Bialas, A. R.; Kamitaki, N.; Martersteck, E. M.; et al. 2015. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5): 1202–1214.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Miao; Zhun; Deng; Ke; Wang; Xiaowo; Zhang; and Xuegong. 2018. DEsingle for detecting three types of differential expression in single-cell RNA-seq data. *Bioinformatics*.

Muraro, M. J.; Dharmadhikari, G.; Grün, D.; Groen, N.; Dielen, T.; Jansen, E.; Van Gurp, L.; Engelse, M. A.; Carlotti, F.; De Koning, E. J.; et al. 2016. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4): 385–394.

Nie, F.; Zhu, W.; and Li, X. 2017. Unsupervised large graph embedding. In *Thirty-first AAAI conference on artificial intelligence*.

Plasschaert, L. W.; Žilionis, R.; Choo-Wing, R.; Savova, V.; Knehr, J.; Roma, G.; Klein, A. M.; and Jaffe, A. B. 2018. A single-cell atlas of the airway epithelium reveals the CFTR-rich pulmonary ionocyte. *Nature*, 560(7718): 377–381.

Rao, J.; Zhou, X.; Lu, Y.; Zhao, H.; and Yang, Y. 2021. Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. *Iscience*, 24(5): 102393.

Risso, D.; Perraudeau, F.; Gribkova, S.; Dudoit, S.; and Vert, J. P. 2018. A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications*, 9(1): 284.

Schaum, N.; Karkanias, J.; Neff, N. F.; May, A. P.; Quake, S. R.; Wyss-Coray, T.; Darmanis, S.; Batson, J.; Botvinnik, O.; Chen, M. B.; et al. 2018. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris: The Tabula Muris Consortium. *Nature*, 562(7727): 367.

Tian, T.; Wan, J.; Song, Q.; and Wei, Z. 2019. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4): 191–198.

Tian, T.; Zhang, J.; Lin, X.; Wei, Z.; and Hakonarson, H. 2021. Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. *Nature communications*, 12(1): 1–12.

Tosches, M. A.; Yamawaki, T. M.; Naumann, R. K.; Jacobi, A. A.; Tushev, G.; and Laurent, G. 2018. Evolution of pallium, hippocampus, and cortical cell types revealed by single-cell transcriptomics in reptiles. *Science*, 360(6391): 881–888.

Tran, D.; Nguyen, H.; Tran, B.; La Vecchia, C.; Luu, H. N.; and Nguyen, T. 2021. Fast and precise single-cell data analysis using a hierarchical autoencoder. *Nature communications*, 12(1): 1–10.

Tran, T. N.; and Bader, G. D. 2020. Tempora: cell trajectory inference using time-series single-cell RNA sequencing data. *PLoS computational biology*, 16(9): e1008205.

Wang, B.; Zhu, J.; Pierson, E.; Ramazzotti, D.; and Batzoglou, S. 2017. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature methods*, 14(4): 414–416.

Wang, J.; Ma, A.; Chang, Y.; Gong, J.; Jiang, Y.; Qi, R.; Wang, C.; Fu, H.; Ma, Q.; and Xu, D. 2021. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nature communications*, 12(1): 1–11.

Wang, Y.; Tang, Z.; Huang, H.; Li, J.; Wang, Z.; Yu, Y.; Zhang, C.; Li, J.; Dai, H.; Wang, F.; et al. 2018. Pulmonary alveolar type I cell population consists of two distinct subtypes that differ in cell fate. *Proceedings of the National Academy of Sciences*, 115(10): 2407–2412.

Wolf, F. A.; Angerer, P.; and Theis, F. J. 2018. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1): 1–5.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487. PMLR.

Yan, L.; Yang, M.; Guo, H.; Yang, L.; Wu, J.; Li, R.; Liu, P.; Lian, Y.; Zheng, X.; Yan, J.; et al. 2013. Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nature structural & molecular biology*, 20(9): 1131–1139.

Young, M. D.; Mitchell, T. J.; Braga, F. A. V.; Tran, M. G.; Stewart, B. J.; Ferdinand, J. R.; Collord, G.; Botting, R. A.; Popescu, D.-M.; Loudon, K. W.; et al. 2018. Single-cell transcriptomes from human kidneys reveal the cellular identity of renal tumors. *Science*, 361(6402): 594–599.

Zeng, Y.; Zhou, X.; Rao, J.; Lu, Y.; and Yang, Y. 2020. Accurately clustering single-cell RNA-seq data by capturing structural relations between cells through graph convolutional network. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 519–522. IEEE.