Single-cell RNA sequencing data clustering using graph convolutional networks

1st Tianyu Wang
Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
tianyu.wang@uconn.edu

2nd Bingjun Li
Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
bingjun.li@uconn.edu

3rd Sheida Nabavi
Computer Science and Engineering
University of Connecticut
Storrs, CT, USA
sheida.nabavi@uconn.edu

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669529

***Abstract*—Single-cell RNA sequencing (scRNAseq) makes it possible to analyze gene expression profiles at the individual cell scale and to discover intrinsic and extrinsic cellular processes in biological research. Cell clustering is one of the most important steps in analyzing scRNAseq data. With rapid developments of single cell sequencing technologies, scRNAseq data grow in size and heterogeneity. However, traditional clustering methods like Kmeans with or without dimension reduction methods, cannot handle high sparse and massive scRNAseq data. Although some deep learning based methods have been proposed to denoise the data and cluster cells simultaneously, learning informative representations of cells for accurate cell clustering is still a challenging problem to be solved. In this work, we propose a deep learning model that combines a deep graph convolutional network (GCN) and a self-supervised mechanism. The GCN considers not only the gene expressions but also the relationship between cells to represent cells. The self-supervised mechanism is employed to provide the clustering assignments of cells. Moreover, we utilize the negative log-likelihood of the negative binomial (NB) function as loss in the data reconstruction due to the assumption that genes expression values can be represented by the NB model. We compared the performance of our proposed method with those of the existing clustering methods for scRNAseq data and conventional clustering methods. Results show that our method achieves better performance in terms of accuracy, adjusted random index (ARI), and normalized mutual information (NMI).**

***Index Terms*—Single cell RNA sequencing, single cell clustering, deep learning, self-supervised learning, graph convolutional neural network.**

I. Introduction

Single-cell RNA sequencing (scRNAseq) enables researchers to study gene expressions at the cellular level. Cell clustering is an essential step in scRNAseq data analysis. Through the cell clustering analysis, new cell types and cell states can be identified and characterized. K means and hierarchical clustering are popular basic clustering methods that have been used for cell clustering. Dimension reduction methods, like t-Distributed Stochastic Neighbor Embedding (t-SNE) [12] and principal coordinate analysis (PCA), are often used to reduce the dimension of gene expression data before performing Kmeans clustering. Recently, several software tools have been proposed for cell clustering using scRNAseq data and employing Kmeans or hierarchical clustering methods. CIDR [11] has been proposed to address both zero imputation and cell clustering. CIDR first imputes zeros and performs dimension reduction by PCA, then it applies the hierarchical clustering on the gene expression matrix after reducing the data dimension. The heart of this method is designing a new dissimilarity metric based on the imputed gene expression values. Seurat [14] is a package that includes a series of processing steps: data normalization, transformation, decomposition, and Kmeans clustering, to cluster cells using gene expression values. RaceID [5] proposes to use the K-medoids instead of Kmeans, which can improve the clustering performance. SIMLR [20] measures the pairwise cell similarity based on multi-kernels. Then the learned cell-cell similarity is used for visualization, dimension reduction, and Kmeans clustering. SC3 [8] is an ensemble clustering method that computes the consensus matrix using the cluster-based similarity partitioning algorithm (CSPA) [16]. It first reduces the dimension of the gene expression matrix and then computes the pairwise cell distance for clustering. SAFE-clustering [22] is an aggregating method that combines the clustering results from multiple individual methods: CIDR, SC3, Seurat, and t-SNE with Kmeans.

Recently, deep learning based methods have gained interest in the bioinformatics field and several deep learning based methods for cell culstering have been introduced such as DEC, scDeepCluster, scziDesk and DESC. Deep embedded clustering (DEC) [21] proposed to use a self-supervised mechanism to learn feature representations and do clustering analysis simultaneously. It uses a self-supervised mechanism because, unlike the supervised classification problem that the learning models are trained with labeled data, unsupervised clustering models do not require labeled samples. In the self-supervised training, the network is trained iteratively with an auxiliary target distribution (P) that is derived from an estimated distribution (Q). scDeepCluster [17] combines the deep autoencoder method (DCA) [4] and the DEC method for analyzing scRNAseq data. DCA is used to denoise the data, and instead of using mean squared error (MSE) as the loss function, DCA considers using the negative log likelihood

This study was supported by the National Science Foundation (NSF) under grant No. 1942303, PI: Nabavi.

of the zero inflated negative binomial (ZINB) model as the loss. scziDesk [3], similar to scDeepCluster [17], combines the DCA and DEC methods. Compared with scDeepCluster, it proposes to preselect a subset of genes as features to reduce the time consumption and considers the distance between similar cells in the loss function. DESC [10] separates the data denoising and data clustering. It firstly pretrains an autoencoder network with MSE loss to denoise the data, and secondly computes the KL divergence as loss for clustering.

All the methods above only use the data itself to do the clustering, but seldom consider the underlying relationships in the data when learning the latent representation of cells. Inspired by the work in [2], we propose to use a graph convolutional network (GCN) [7] to consider both the relationship among cells, through graph structure, and genes expression values, through node (cell) attributes, for learning the representation of cells. In our proposed end-to-end self-supervised deep learning network, we combine the GCN and an unsupervised deep clustering method. We first use the K nearest neighbors (KNN) to construct the cell-cell input graph, which can reveal the underlying structure of the cell relationship. Then, we use the GCN and a fully-connected (FC) network module to learn the latent representation of the cells. In the reconstruction part, a hidden layer is used to form the target distribution that is employed for computing the KL divergence, and the negative binomial (NB) distribution is considered to model scRNAseq data. We consider the dispersion and mean parameters of the NB model in the reconstruction part. The code is available at https://github.com/NabaviLab/sigDGCNb.

Our main contributions in this study are: i) employing a parallel GCN and FC network to encode the scRNAseq data, ii) considering both cell similarity and negative binomial distribution to better model the data, and iii) using a novel self-supervised approach to conduct the clustering assignments in an end-to-end trained network.

The rest of the paper is organized as follows: Section II presents the main methods. Section III introduces the real datasets and data preprocessing. In Section IV, results are discussed to show the effectiveness of the proposed method. Section V concludes the paper.

II. Methods

The overall structure of the proposed model is shown in Fig. 1. The proposed clustering model consists of two parallel networks –a GCN and an FC network– and a reconstruction part. Gene expression values and the cell-cell network are the inputs of the GCN; while the gene expression values are the inputs of the FC network. The weighted sum of the features learned by the GCN and those learned by the FC network in each layer is used as the input of the next layer. In the reconstruction part, there are three channels –clustering module, self-supervised module, and negative binomial module– in the output part that are corresponded to different parts of the overall loss function. The output of the clustering module indicates the clustering assignment for each cell.

A. Cell-cell network

We use KNN to construct the cell-cell graph from gene expression data. Given the gene expression matrix $\mathbf{X} \in R^{M \times N}$, where M is number of cells and N is number of selected genes, we first compute the distance between each pair of cells. For each cell, we use KNN to select its top $K(= 5)$ nearest neighbors and construct the connections between them. Thus, we construct the binary cell-cell adjacency matrix $\mathbf{A} \in R^{M \times M}$, where the ones in the matrix represent the connections between pairs of cells. We examined the effect of selecting different K on the performance of the model that is shown in the Results section.

B. Graph convolutional network and fully-connected network modules

Although the GCN takes the structure (cell-cell graph) and node attributes (gene expression values) as input and extracts the community features for each cells with the influence of the relationships between the neighbour cells, it lacks the individual features inner the nodes (cells). Thus, we utilize a parallel encoder network, GCN and FC with the same number of layers L, to learn the individual latent representation for the nodes. The FC layers are defined as:

$$\mathbf{O}^{(l)} = ReLU(\mathbf{O}^{(l-1)} \times \mathbf{W}_{fc}^{(l)} + \mathbf{b}_{fc}^{(l)}), \tag{1}$$

where $\mathbf{O}^{(l-1)}$ is the input of the FC layer l and we denote the gene expression value matrix $\mathbf{X}$ as $\mathbf{O}^{(0)}$. $\mathbf{W}_{fc}^{(l)}$ and $\mathbf{b}_{fc}^{(l)}$ are weight parameters of layer l with $l \in 1, ...L$.

Given graph $G = (V, E)$, where V represents the nodes (cells) and E represents the edges between the nodes, the gene expression values across one cell can be regarded as the node attributes. The adjacency matrix $\mathbf{A} \in R^{M \times M}$ is used to represent the node connections (cell-cell adjacency matrix). The adjacency matrix is further normalized to $\mathbf{L} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}^{1/2}$, where $\mathbf{D} \in R^{M \times M}$ is a diagonal matrix and $\mathbf{I} \in R^{M \times M}$ is the identity matrix. The GCN layer is defined as following:

$$\mathbf{H}^{(l)} = ReLU(\mathbf{L}\mathbf{X}^{(l-1)}\mathbf{W}_{gcn}^{(l)}), \tag{2}$$

where $\mathbf{X}^{(l-1)} \in R^{M \times F^{(l-1)}}$ is the input and $\mathbf{W}_{gcn}^{(l)}$ is the parameter matrix that needs to be updated. For the first GCN layer, $\mathbf{X}^{(\mathbf{0})}$ is $\mathbf{X}$ while for the following GCN layers, $\mathbf{X}^{(l)}$ is the weighted sum of the output of the GCN layer and the FC layer:

$$\mathbf{X}^{(l)} = \alpha\mathbf{H}^{(l)} + (1 - \alpha)\mathbf{O}^{(l)}. \tag{3}$$

Given the gene expression value matrix $\mathbf{X}$, the encoder part (GCN and FC network) outputs the latent representation of cells $\mathbf{X}^{(L)} \in R^{M \times F^{(L)}}$, where $F^{(L)}$ is the dimension of the feature map. We use $\tilde{\mathbf{H}}$ and $\mathbf{H}$ to denote the feature map $\mathbf{X}^{(L)}$ and $\mathbf{H}^{(L)}$ in the following sections.

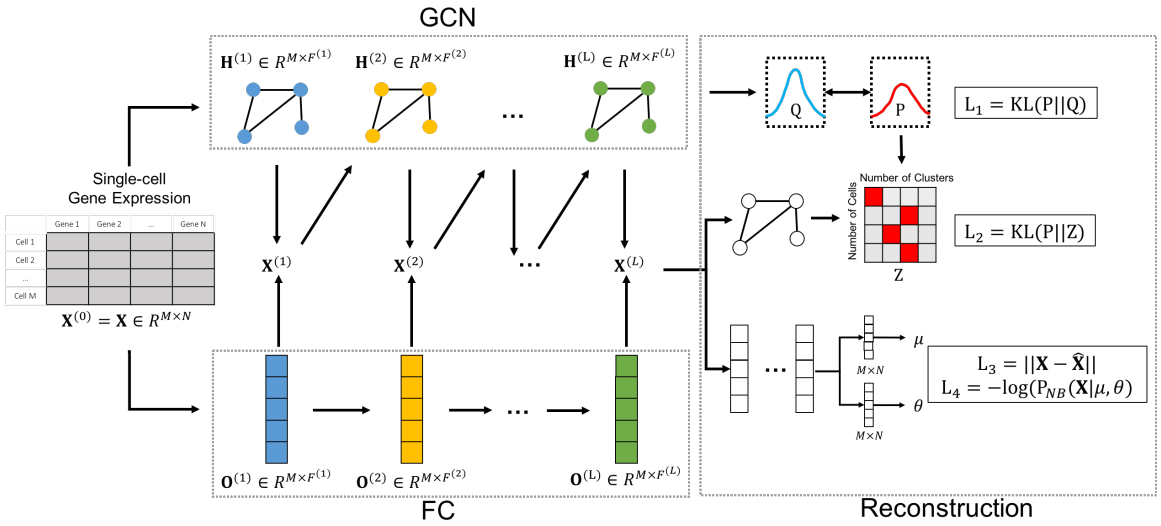

Fig. 1. Structure of the proposed deep learning network (sigDGCNb) for single cell clustering.

C. Self-supervised module

The GCN and FC network modules learn the latent representation of the cells, however, they cannot be directly used for clustering. We employ a self-supervised method to provide the soft clustering labels for the cells. Given the embedded feature map output by the last GCN layer ($\mathbf{H}$), we use the Student's t distribution [12] as a kernel to measure the similarity between the embedded feature of each cell $\mathbf{h}_i$ and each cluster centroid $\mathbf{c}_j$.

$$q_{ij} = \frac{(1 + \|\mathbf{h}_i - \mathbf{c}_j\|^2/t)^{-\frac{t+1}{2}}}{\sum_{j' \neq j}(1 + \|\mathbf{h}_i - \mathbf{c}_{j'}\|^2/t)^{-\frac{t+1}{2}}}, \tag{4}$$

where $\mathbf{h}_i$ is the i-th row in the latent feature map $\mathbf{H}$ and $\mathbf{c}_j$ is the feature of cluster j. For the initialization of $\mathbf{c}_j$, we pretrain an autoencoder network and utilize Kmeans on the latent features. t is the degree of freedom of the Student's t-distribution where the default value is 1. $Q = [q_{ij}]$ is the soft assignment of all the cells, which represents the probability that cell i belongs to cluster j. Next, an auxiliary target distribution (P) is defined as equation (5) to help alternatively optimize the data representation. The components of Q with higher values have more confidence in assigning clustering labels which contribute more in generating the target distribution P. The target distribution P is then used to supervise Q. P and Q are learned alternatively, as a result, P an Q have higher confidence in clustering and the clusters are refined alternatively.

$$p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_{j' \neq j}(q_{ij'}^2/\sum_i q_{ij'})}. \tag{5}$$

The auxiliary target distribution P puts more emphasis on pairwise points with higher similarity, thus it makes cells get closer to the cluster centroids. With the auxiliary target distribution P and the estimated distribution Q, we define the KL-divergence loss between them as:

$$L_1 = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}. \tag{6}$$

By minimizing the KL-divergence loss, the target distribution P and the estimated distribution Q are updated alternatively, which is regarded as a self-supervised mechanism.

D. Clustering module

The parallel network of the GCN and FC network modules learns the latent features of the nodes (cells) in the graph. We implement a GCN layer that is activated by the softmax function to generate the clustering assignment.

$$Z = softmax(\mathbf{L}\tilde{\mathbf{H}}\mathbf{W}), \tag{7}$$

where $Z \in R^{M\times C}$ represents a clustering assignment and C is the number of the clusters. To supervise the updating of distribution Z, we define the loss function as:

$$L_2 = KL(P||Z) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{z_{ij}}. \tag{8}$$

E. Negative binomial module

We assume gene expression value X_{ij} follows the negative binomial (NB) distribution with mean μ_{ij} and dispersion θ_{ij}:

$$\begin{aligned} P_{NB}(X_{ij}|\mu_{ij},\theta_{ij}) = &\frac{\Gamma(X_{ij}+\theta_{ij})}{\Gamma(X_{ij}+1)\Gamma(\theta_{ij})} \times \\ &(\frac{\theta_{ij}}{\theta_{ij}+\mu_{ij}})^{\theta_{ij}} \times (\frac{\mu_{ij}}{\theta_{ij}+\mu_{ij}})^{X_{ij}}, \end{aligned} \tag{9}$$

where μ_{ij} and θ_{ij} are the parameters to be estimated. We use a FC network as the decoder network to estimate these

parameters. Given $\tilde{\mathbf{H}} = \mathbf{X}^{(L)}$ as the embedded feature map, the decoder network is defined as:

$$\begin{aligned} \mathcal{D} &= ReLU(f(\tilde{\mathbf{H}})), \\ \mu &= \sigma_1(\mathcal{D}\mathbf{W}_\mu), \\ \theta &= \sigma_2(\mathcal{D}\mathbf{W}_\theta), \end{aligned} \tag{10}$$

where σ_1 and σ_2 are activation function. Since the mean μ and dispersion θ are non-negative, we use $ReLU(\cdot)$ for σ_1 and $exp(\cdot)$ for σ_2. We take the negative log-likelihood of the NB distribution with the estimated mean and dispersion parameters, and the MSE of the reconstruction between the input gene expression values and the estimated mean parameters for the loss functions:

$$\begin{aligned} L_3 &= \sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij} - \hat{X}_{ij})^2, \\ L_4 &= -\sum_{i=1}^{M}\sum_{j=1}^{N}\log(P_{NB}(X_{ij}|\mu_{ij}, \theta_{ij})), \end{aligned} \tag{11}$$

where $\hat{X}_{ij}$ is the reconstructed X_{ij}.

F. Overall loss function

We utilize this self-supervised training mechanism to train the proposed end-to-end deep network cell clustering model. The total loss is a combination of the KL-divergence losses, the reconstruction loss, and the loss of the NB model:

$$L = \lambda_1 L_1 + \lambda_2 L_2 + \lambda_3 L_3 + \lambda_4 L_4, \tag{12}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are hyperparameters that controls the relative importance of each part of the total loss. The default value of λ_1 and λ_2 is 0.01, λ_3 is 1, and λ_4 is 0.001.

The proposed model has 4 layers in the GCN and FC network modules ($L = 4$) and the dimension in each layer is 128, 64, 32, 10, respectively. The decoder NB module has 2 layers that have the dimensions of half of the number of genes ($N/2$) and the original input size N. Note that we did not use all the genes as input, instead we chose top $N = 500$ genes with high variance. Similar to the other methods we need to define the number of clusters C first.

III. Datasets

To evaluate the performance of the proposed clustering method, we used five datasets: Usoskin , Baron Human, Baron Mouse, Muraro, and Segerstolpe. First, we preprocessed all the datasets by removing genes that are not expressed across all the cells. Then, we normalized all the datasets by min-max scaling and applied log-transformation. We used the cell labels given by the authors as the ground truth for these datasets.

The Usoskin dataset [18] is available in Gene Expression Omnibus (GEO) dataset with accession number GSE59739. This dataset contains 622 cells which are classified into 4 groups: 139 neurofilament containing (NF), 81 peptidergic nociceptors (PEP), 169 non-peptidergic nociceptors (NP), and 233 tyrosine hydroxylase containing (TH). The filtered dataset has 13776 genes expressed in 622 cells after preprocessing.

The Baron Mouse [1] and the Baron Human [1] datasets are available on GEO with access number GSE84133. They are from the mouse and human pancreas, respectively. The cells are sequenced using the inDrop protocol. The filtered BaronMouse dataset has expression data of 1,886 cells and 14,861 genes from 13 cell populations. In the BaronHuman dataset, there are expression data of 8,569 cells and 17,499 genes from 14 cell populations after filtering.

The Muraro [13] and Segerstolpe [15] datasets are from the human pancreas. The Muraro dataset is available on GEO with access number GSE85241 and sequenced by the CEL-Seq2 protocol. After filtering the dataset, there are expression data of 2,122 cells and 18,915 genes, where the cells are annotated to 9 classes. The Segerstolpe dataset includes expression data of 2,133 cells and 22,757 genes, sequenced by the SMART-Seq2 protocol, from 13 cell populations.

IV. Results

A. Metrics to evaluate the clustering performance

We used three metrics to evaluate the performance of cell clustering: Adjusted Random Index (ARI) [6], Normalized Mutual Information (NMI) [19], and clustering accuracy (ACC). ACC, ARI, and NMI evaluate the consistency between the true clustering labels and the predicted clustering labels of cells. The higher values of ACC, ARI, and NMI (closer to 1) for a clustering method represent that the method has better clustering performance. Assume we have two groups of clusters X and Y of N data points, where $\mathbf{X} = X_1, ..., X_i, ...X_r$ and $\mathbf{Y} = Y_1, ...Y_j, ...Y_s$.

ARI is defined as following:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}/\binom{N}{2}]}, \tag{13}$$

where n_{ij} is the number of common points in both cluster X_i and cluster Y_j, $a_i = \sum_j n_{ij}$, and $b_j = \sum_i n_{ij}$.

NMI is defined as the mutual information (MI) between X and Y normalized by the mean of the entropy of X and Y:

$$\begin{aligned} NMI &= \frac{MI(X,Y)}{\text{mean}(H(X), H(Y))} \\ MI(X,Y) &= \sum_{i=1}^{|X|}\sum_{j=1}^{|Y|}\frac{|X_i \cap Y_j|}{N} log(\frac{N|X_i \cap Y_j|}{|X_i||Y_j|}) \end{aligned} \tag{14}$$

where X_i and Y_j are the number of elements in the i-th and j-th cluster, respectively. The entropy of each cluster is defined as:

$$\begin{aligned} H(X) &= -\sum_{i=1}\frac{|X_i|}{N}\log\frac{|X_i|}{N} \\ H(Y) &= -\sum_{j=1}\frac{|Y_j|}{N}\log\frac{|Y_j|}{N} \end{aligned} \tag{15}$$

To compute the clustering accuracy, we first employed Hungarian [9] method to find the best match that map the

estimated clusters to the true clusters. The clustering accuracy is defined as:

$$ACC = \frac{\sum_{i=1}^{N} 1 \text{ if } u_i == \text{MAP}(v_i)}{N} \quad (16)$$

where u_i is the true clustering label for data point i, v_i is the assigned clustering label for data point i, MAP(v_i) is the mapping from v_i to u_i.

B. Evaluation of the clustering performance

To evaluate the performance of our proposed clustering model, we compared the clustering results of our model with those of six clustering tools developed for scRNAseq data and three classical clustering methods: CIDR [11], SOUP [23], SIMLR [20], RaceID [5], scziDesk [3], DESC [10], t-SNE+Kmeans, PCA+Kmeans, and PCA+hierarchical in terms of ARI, NMI and accuracy. For t-SNE+Kmeans and PCA+Kmeans/hierarchical, we first apply t-SNE [12] or PCA to the datasets to reduce the dimension and then apply the Kmeans or hierarchical methods for clustering.

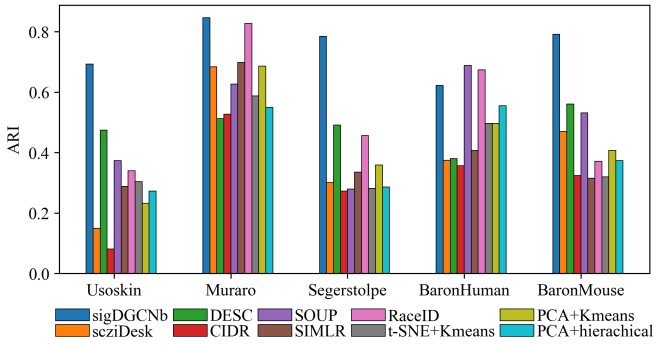

Fig. 2. Bar plots of the ARI values to show the performance of the scRNAseq data clustering tools on five datasets.

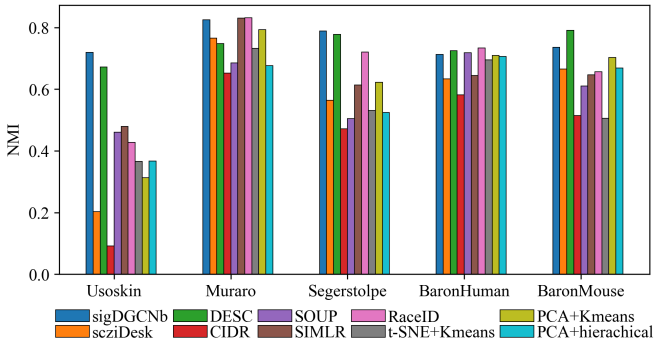

Fig. 3. Bar plots of the NMI values to show the performance of the scRNAseq data clustering tools on five datasets.

Fig. 2 and Fig. 3 show the barplots of ARIs and NMIs of the seven methods using the five datasets. As can be seen in Fig. 2, our proposed method, sigDGCNb, achieves the best ARIs across all the datasets except one. On the Usoskin dataset, sigDGCNb performs the best ARI of 0.693 and improves more than 20% compared with the other methods. All the methods perform relatively well on the Muraro dataset. RaceID has a comparable ARI with that of our method on the Muraro dataset and a better ARI on the Baron Human dataset, however, it does not perform well on the Usoskin, Segerstolpe, and Baron Mouse datasets. The same for SOUP that shows a better ARI on the Baron Human dataset compared to our model, but much worse on the other datasets. Compared with the autoencoder based method scziDesk, our proposed model, sigDGCNb, has a better performance in terms of ARI. It indicates that the integration of the data structure helps improving the learning of latent features. As shown in Fig. 3, sigDGCNb also shows the best performance in terms of NMI on the Usoskin and Segerstolpe datasets and achieves the second best on the Baron Mouse dataset. On the Muraro dataset, compared with the best NMI of 0.831 our method has a comparable performance with the NMI of 0.825.

TABLE I
COMPARISON OF ACCURACY BETWEEN SEVEN METHODS USING FIVE REAL DATASETS

Methods	Usoskin	Muraro	Seger stolpe	Baron Human	Baron Mouse
sigDGCNb	**0.792**	**0.852**	**0.786**	0.686	**0.793**
scziDesk	0.545	0.812	0.478	0.55	0.654
DESC	0.523	0.594	0.566	0.556	0.624
CIDR	0.439	0.666	0.4	0.452	0.419
SOUP	0.373	0.627	0.279	0.688	0.532
SIMLR	0.288	0.699	0.336	0.407	0.315
RaceID	0.513	0.801	0.632	**0.748**	0.571
t-SNE+ Kmeans	0.562	0.721	0.406	0.569	0.550
PCA+ Kmeans	0.547	0.783	0.433	0.543	0.536
PCA+ hierarchical	0.573	0.641	0.426	0.607	0.491

In terms of clustering accuracy our proposed method also shows a strong performance. The clustering accuracy, described in the previous section (ACCs), of all the clustering methods on the five datasets are shown in Table I. As can be seen, sigDGCNb provides higher accuracy on four datasets and the third best accuracy (comparable with the best and the second best) on the baron Human dataset. On the Usoskin dataset, the ACC of our method is 21.9% more than the second best of the other method. On the Segerstolpe, Baron Mouse, and Muraro datasets, our method achieves the best ACCs, which are 15.4%, 13.9%, and 4% more than the second best ACCs, respectively. Especially compared to the other authoencoder based model, scziDesk, our model has higher clustering accuracy.

Fig. 4 shows the visualization of cells using the six clustering methods on the Usoskin dataset. The dots represent the cells and the colors represent the true labels of the cells in the figure. For our proposed sigDGSNb method, we first applied t-SNE on the learned 10-dimension features of the cells. Then we used the two components of applying t-SNE for the visualization. We did the same process for scziDesk

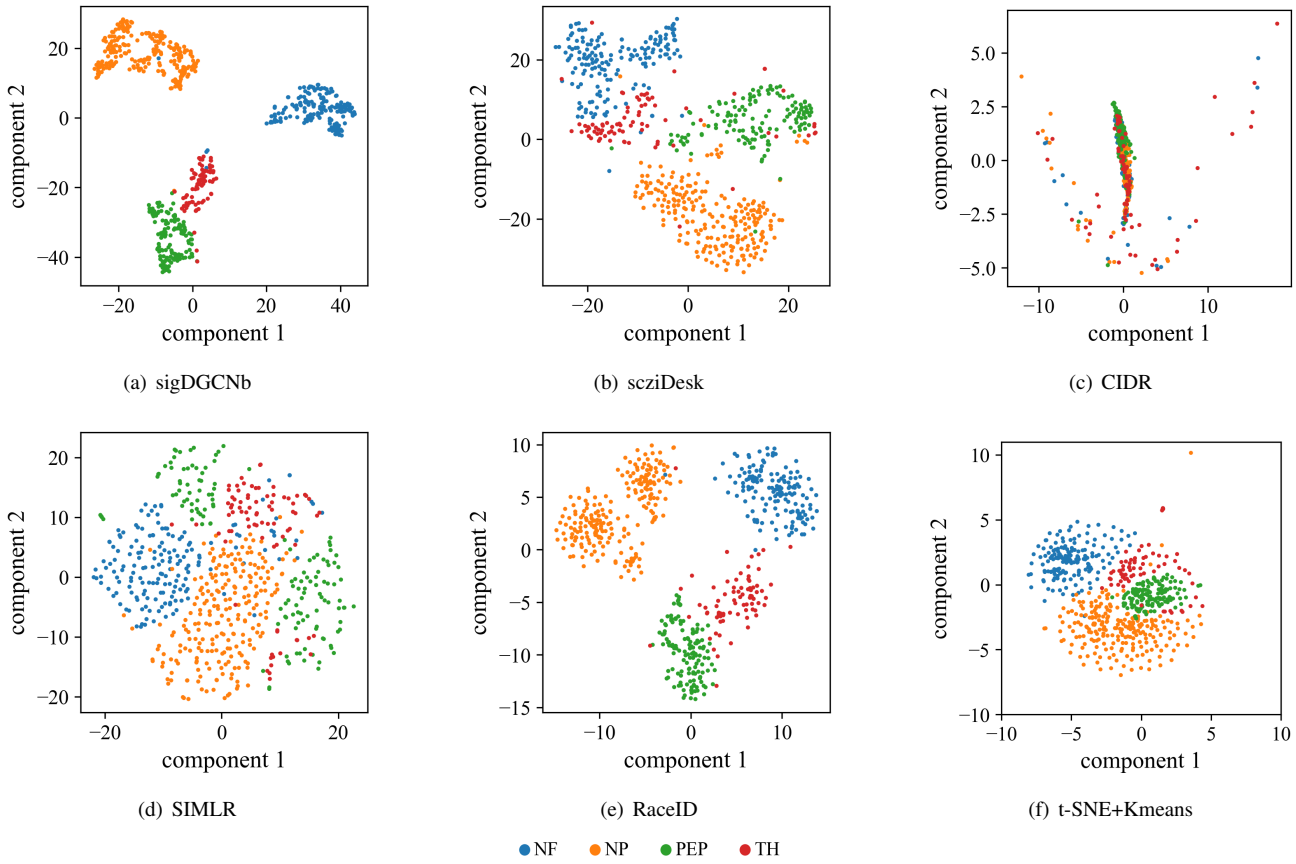

Fig. 4. Visualization of Usoskin dataset using (a) sigDGCNb (b) scziDesk (c) CIDR (d) SIMLR (e) RaceID (f) t-SNE+Kmeans.

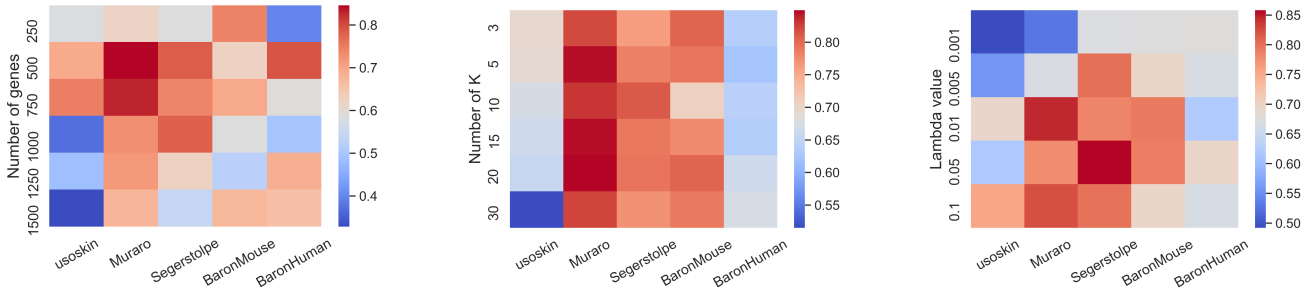

Fig. 5. ARI values of selecting (a) different numbers of genes (N), (b) different number of nearest neighbors (K), (c) different λ_1 and λ_2 values.

method because it is an autoencoder-based method and it outputs the learned features. For the other methods, we used the outputs from their R packages to generate the visualization. We can observe that our method can separate the groups better for the cell clustering compared with the other methods.

C. Evaluation of the sensitivity to hyperparameters

In our method, we selected $N = 500$ genes that have the highest variances across the dataset as input by default. We evaluated the sensitivity of the proposed method to the number of input genes. We selected the top 250, 500, 750, 1000, 1250, 1500 highest variant genes as the input for each dataset. The ARI values are shown as a heatmap in Fig. 5(a), where the red color shows the higher ARI score and the blue color shows the lower ARI score. The model performance for using 500 and 750 high variant genes does not show much difference. Considering the time complexity, we selected the top 500 variable genes as the default input in our method. We need to assign the number of nearest neighbors (K) when constructing the cell-cell network. We varied the parameter K from 3 to 30 to see how this parameter affects the performance of the clustering. It is observed that the ARIs using each dataset do not change notably with changing parameter K as can be seen in Fig. 5(b). Therefore, we can see that selecting a larger

number of nearest neighbors does not add more information and does not have a huge impact on the performance. We also evaluated the sensitivity of choosing the hyperparameter λ_1 and λ_2, mentioned in Section II. We varied λ_1 and λ_2 ($\lambda_1 = \lambda_2$) from 0.001 to 0.1 to examine the clustering performance. From Fig. 5(c) we can see by selecting larger λ_1 and λ_2 the proposed model performs better in terms of ARI than by selecting smaller ones. It indicates that the model benefits from the self-supervised module and the clustering module.

V. Conclusion

In this study, we proposed a novel deep learning method, named sigDGCNb, based on the self-supervised learning mechanism for single-cell clustering. We developed an end-to-end trained model that combines a graph convolutional network and a fully connected networks to consider both the relationship among cells –through a cell-cell network– and gene expressions –through the node attributes in the graph. To learn a powerful representation of cells, we considered the negative binomial distribution to model scRNAseq gene expression values and employed the negative log-likelihood of the NB function as loss in the learning process in addition to the conventional MSE loss. To provide the clustering assignments of cells, we employed the self-supervised mechanism and computed the KL divergence between the target and estimated distributions as the self-supervised loss.

We compared the performance of our proposed method with those of nine other clustering methods (six cell clustering tools for scRNAseq data and three conventional clustering methods) using five real datasets. To evaluate the clustering performance, we used standard metric ARI, NMI, and clustering accuracy on all the datasets. Results show that our method outperforms the other methods. Compared with the other deep learning based method, our method shows better ARI, NMI, and clustering accuracy, which indicates that the model benefits from the integration of the data structure and data attributes.

In conclusion, the proposed deep learning model which integrates data structure and considers data likelihood shows better performance. In future work, we will introduce the attention mechanism to enhance the weights of the data structure.

References

[1] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, D. A. Melton, and I. Yanai, "A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure," *Cell Systems*, vol. 3, no. 4, pp. 346–360.e4, Oct. 2016. [Online]. Available: https://www.cell.com/cell-systems/abstract/S2405-4712(16)30266-6

[2] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural Deep Clustering Network," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1400–1410. [Online]. Available: https://doi.org/10.1145/3366423.3380214

[3] L. Chen, W. Wang, Y. Zhai, and M. Deng, "Deep soft K-means clustering with self-training for single-cell RNA sequence data," *NAR Genomics and Bioinformatics*, vol. 2, no. 2, Jun. 2020. [Online]. Available: https://doi.org/10.1093/nargab/lqaa039

[4] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," *Nature Communications*, vol. 10, no. 1, p. 390, Jan. 2019, bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational models;Machine learning;Statistical methods Subject_term_id: computational-models;machine-learning;statistical-methods. [Online]. Available: https://www.nature.com/articles/s41467-018-07931-2

[5] D. Grün, M. Muraro, J.-C. Boisset, K. Wiebrands, A. Lyubimova, G. Dharmadhikari, M. van den Born, J. van Es, E. Jansen, H. Clevers, E. de Koning, and A. van Oudenaarden, "De Novo Prediction of Stem Cell Identity using Single-Cell Transcriptome Data," *Cell Stem Cell*, vol. 19, no. 2, pp. 266–277, Aug. 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4985539/

[6] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985. [Online]. Available: https://link.springer.com/article/10.1007/BF01908075

[7] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv:1609.02907 [cs, stat]*, Feb. 2017, arXiv: 1609.02907. [Online]. Available: http://arxiv.org/abs/1609.02907

[8] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, and M. Hemberg, "SC3 - consensus clustering of single-cell RNA-Seq data," *Nature methods*, vol. 14, no. 5, pp. 483–486, May 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5410170/

[9] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.3800020109. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109

[10] X. Li, K. Wang, Y. Lyu, H. Pan, J. Zhang, D. Stambolian, K. Susztak, M. P. Reilly, G. Hu, and M. Li, "Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis," *Nature Communications*, vol. 11, no. 1, p. 2338, May 2020, bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Machine learning;RNA sequencing;Software;Statistical methods Subject_term_id: machine-learning;rna-sequencing;software;statistical-methods. [Online]. Available: https://www.nature.com/articles/s41467-020-15851-3

[11] P. Lin, M. Troup, and J. W. K. Ho, "CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data," *Genome Biology*, vol. 18, p. 59, Mar. 2017. [Online]. Available: https://doi.org/10.1186/s13059-017-1188-0

[12] L. v. d. Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html

[13] M. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. van Gurp, M. Engelse, F. Carlotti, E. de Koning, and A. van Oudenaarden, "A Single-Cell Transcriptome Atlas of the Human Pancreas," *Cell Systems*, vol. 3, no. 4, pp. 385–394.e3, Oct. 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5092539/

[14] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev, "Spatial reconstruction of single-cell gene expression data," *Nature Biotechnology*, vol. 33, no. 5, pp. 495–502, May 2015, bandiera_abtest: a Cg_type: Nature Research Journals Number: 5 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Gastrulation;Machine learning;Statistical methods Subject_term_id: gastrulation;machine-learning;statistical-methods. [Online]. Available: https://www.nature.com/articles/nbt.3192

[15] A. Segerstolpe, A. Palasantza, P. Eliasson, E.-M. Andersson, A.-C. Andréasson, X. Sun, S. Picelli, A. Sabirsh, M. Clausen, M. K. Bjursell, D. Smith, M. Kasper, C. Ammala, and R. Sandberg, "Single-Cell Transcriptome Profiling of Human Pancreatic Islets in Health and Type 2 Diabetes," *Cell Metabolism*, vol. 24, no. 4, pp. 593–607, Oct. 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5069352/

[16] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, no. null, pp. 583–617, Mar. 2003. [Online]. Available: https://doi.org/10.1162/153244303321897735

[17] T. Tian, J. Wan, Q. Song, and Z. Wei, "Clustering single-cell RNA-seq data with a model-based deep learning approach," *Nature Machine Intelligence*, vol. 1, no. 4, pp. 191–198, Apr. 2019, bandiera_abtest: a Cg_type: Nature Research Journals Number: 4 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational models;Machine learning Subject_term_id: computational-models;machine-learning. [Online]. Available: https://www.nature.com/articles/s42256-019-0037-0

[18] D. Usoskin, A. Furlan, S. Islam, H. Abdo, P. Lönnerberg, D. Lou, J. Hjerling-Leffler, J. Haeggström, O. Kharchenko, P. V. Kharchenko, S. Linnarsson, and P. Ernfors, "Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing," *Nature Neuroscience*, vol. 18, no. 1, pp. 145–153, Jan. 2015. [Online]. Available: https://www.nature.com/articles/nn.3881

[19] N. X. Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance," *Journal of Machine Learning Research*, vol. 11, no. 95, pp. 2837–2854, 2010. [Online]. Available: http://jmlr.org/papers/v11/vinh10a.html

[20] B. Wang, J. Zhu, E. Pierson, D. Ramazzotti, and S. Batzoglou, "Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning," *Nature Methods*, vol. 14, no. 4, pp. 414–416, Apr. 2017, bandiera_abtest: a Cg_type: Nature Research Journals Number: 4 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Gene expression;Genome informatics;Machine learning;Statistical methods Subject_term_id: gene-expression;genome-informatics;machine-learning;statistical-methods. [Online]. Available: https://www.nature.com/articles/nmeth.4207

[21] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," *arXiv:1511.06335 [cs]*, May 2016, arXiv: 1511.06335. [Online]. Available: http://arxiv.org/abs/1511.06335

[22] Y. Yang, R. Huh, H. W. Culpepper, Y. Lin, M. I. Love, and Y. Li, "SAFE-clustering: Single-cell Aggregated (from Ensemble) clustering for single-cell RNA-seq data," *Bioinformatics*, vol. 35, no. 8, pp. 1269–1277, Apr. 2019. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty793

[23] L. Zhu, J. Lei, L. Klei, B. Devlin, and K. Roeder, "Semisoft clustering of single-cell data," *Proceedings of the National Academy of Sciences*, vol. 116, no. 2, pp. 466–471, Jan. 2019.