Bioinformatics, 2022, 1–7
https://doi.org/10.1093/bioinformatics/btac099
Advance Access Publication Date: 17 February 2022
Original Paper

Gene expression

scGAC: a graph attentional architecture for clustering single-cell RNA-seq data

Yi Cheng and Xiuli Ma *

Key Laboratory of Machine Perception (MOE), School of Artificial Intelligence, Peking University, Beijing 100871, China

*To whom correspondence should be addressed.

Associate Editor: Anthony Mathelier

Received on September 13, 2021; revised on February 3, 2022; editorial decision on February 7, 2022; accepted on February 15, 2022

Abstract

Motivation: Emerging single-cell RNA sequencing (scRNA-seq) technology empowers biological research at cellular level. One of the most crucial scRNA-seq data analyses is clustering single cells into subpopulations. However, the high variability, high sparsity and high dimensionality of scRNA-seq data pose lots of challenges for clustering analysis. Although many single-cell clustering methods have been recently developed, few of them fully exploit latent relationship among cells, thus leading to suboptimal clustering results.

Results: Here, we propose a novel unsupervised clustering method, scGAC (single-cell Graph Attentional Clustering), for scRNA-seq data. scGAC firstly constructs a cell graph and refines it by network denoising. Then, it learns clustering-friendly representation of cells through a graph attentional autoencoder, which propagates information across cells with different weights and captures latent relationship among cells. Finally, scGAC adopts a self-optimizing method to obtain the cell clusters. Experiments on 16 real scRNA-seq datasets show that scGAC achieves excellent performance and outperforms existing state-of-art single-cell clustering methods.

Availability and implementation: Python implementation of scGAC is available at Github (https://github.com/Joye9285/scGAC) and Figshare (https://figshare.com/articles/software/scGAC/19091348).

Contact: xlma@pku.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Rapid development of single-cell RNA sequencing (scRNA-seq) technology has enabled transcriptional gene expression measurement at cellular level, empowering researchers to reveal the heterogeneity and diversity among cell subpopulations (Guo *et al.*, 2019; Hwang *et al.*, 2018). Unsupervised clustering is one of the most essential analysis methods to identify cell subpopulations in scRNA-seq data. Effective single-cell clustering methods are desperately desired, since many subsequent analyses, such as marker gene identification and differentially expressed gene analysis, highly depend on accurate identification of cell subpopulations.

Despite the improvements in sequencing technologies and the progress of unsupervised clustering methods, clustering single cells still remains highly challenging (Kiselev *et al.*, 2019). First, the gene expression values can be highly variable among cells of the same type (Shalek *et al.*, 2013), or rather smooth in transient cell states, which make the relationship across cells more difficult to capture. Second, due to experimentally technical factors, scRNA-seq data are extremely sparse with a large number of dropouts (Stegle *et al.*, 2015). Another challenge is the high dimensionality of scRNA-seq data. All these challenges make it hard to characterize the feature of cells and the latent relationship among cells, which further hinder the clustering of single cells.

Many single-cell clustering methods have been proposed in recent years. Considering the high dimensionality, a common practice is to learn low-dimensional representation of cells followed by some traditional clustering methods, such as k-means clustering or hierarchical clustering. To learn the low-dimensional representation, some methods construct cell–cell similarity matrix to reduce the dimensionality. For example, CIDR (Lin *et al.*, 2017) first measures dissimilarity between cells based on Euclidean distance between imputed gene expression values, and then performs hierarchical clustering on it. However, Euclidean distance only measures linear similarity among cells, therefore can hardly capture the latent relationship.

To obtain better clustering performance on scRNA-seq data, many methods ensemble different similarity measurements or clustering results. For instance, RCSL (Mei *et al.*, 2021) measures both global and local relationship between cells to construct a similarity matrix, from which a block diagonal matrix is then derived to get the final clustering results. SC3 (Kiselev *et al.*, 2017) performs k-means on multiple similarity matrices, which are constructed based on different similarity measurements and dimensionality reduction methods. Then, it combines multiple results into a consensus matrix and performs hierarchical clustering on it. Another ensemble clustering method, SAME (Huh *et al.*, 2020), integrates results of five clustering methods to get better performance. Although these methods explicitly incorporate cell–cell similarity information into representation of cells, pairwise similarity can only characterize superficial but not latent relationship among cells. Moreover, representation learning and clustering, as two separate parts in most of these methods, cannot benefit from each other, thus leading to suboptimal clustering results.

To simultaneously learn feature representation and perform clustering, the combination of two deep learning-based methods, autoencoder (Hinton and Salakhutdinov, 2006) and Deep Embedded Clustering (DEC) (Xie *et al.*, 2016), has been explored by a few works. As one of the most popular deep frameworks, autoencoder can automatically learn low-dimensional representation of data in an unsupervised manner. scDeepCluster (Tian *et al.*, 2019) uses a zero-inflated negative binomial model-based autoencoder to model the statistical characteristics of scRNA-seq data, and simultaneously performs clustering by DEC. Similarly, DESC (Li *et al.*, 2020) pre-trains a stacked autoencoder; then, trains the encoder and DEC in the meantime. In general, these methods combine representation learning and clustering into a unified framework to improve the clustering performance. However, they only consider the gene expression information during representation learning, without explicitly characterizing the relationship among cells. Therefore, the learned representation of cells is not clustering-friendly enough.

To naturally embed both expression information and relationship information into latent representation, a recently developed method, scGNN (Wang *et al.*, 2021), combines the emerging graph convolutional network (GCN) (Kipf and Welling, 2017) into single-cell clustering. GCN learns representation of nodes in a graph through neighbor information propagation, considering of both node features and graph topology. It has been proved that representation learned by GCN can improve clustering results (Bo *et al.*, 2020). scGNN integrates GCN into its multi-autoencoder framework. It first constructs a cell graph for GCN by a feature autoencoder. Next, it learns representation of cells in the graph through a GCN-based autoencoder and obtains clustering result by performing *k*-means on cell representation. Representations of cells in the same cluster are iteratively adjusted by some cluster autoencoders. However, since the constructed graph may contain some noisy edges connecting cells of different types and the GCN-based autoencoder directly propagates information between neighbors without differentiating them, different cell types can be mixed up, thus misleading the clustering result.

To overcome the above challenges, we propose a novel single-cell clustering method named 'scGAC' (single-cell Graph Attentional Clustering). A cell graph is firstly constructed, with each node representing a cell and each edge connecting cells with weight initialized by Pearson correlation coefficient. To make the cell graph more trustworthy, network denoising is performed on the graph to remove the noisy edges. Based on the denoised cell graph, scGAC learns latent representation of cells through a graph attentional autoencoder to leverage both the gene expression and cell–cell relationship information. The attention mechanism facilitates scGAC to assign different weights to different neighbors when propagating information across the neighborhood. To simultaneously learn representations and optimize clustering, a self-optimizing method is adopted to get clustering results. Experiments on 16 real scRNA-seq datasets show that scGAC has an excellent clustering performance and outperforms state-of-art single-cell clustering methods.

2 Materials and methods

2.1 Architecture of scGAC

scGAC takes the raw-count gene expression matrix $R_{M\times N}$, with M genes and N cells, as its inputs. To facilitate subsequent analyses, $R_{M\times N}$ will be normalized into $X_{m\times n}$ after the removal of low-quality cells and genes. Figure 1 shows the main architecture of scGAC, which contains three major parts:

- Graph Construction: To help information sharing across different cells, an auxiliary cell graph $A_{n\times n}$ is constructed. First, an initial cell–cell similarity matrix $S_{n\times n}$ is constructed by calculating Pearson correlation coefficient based on $X_{m\times n}$. Network enhancement (NE) (Wang *et al.*, 2018) is then applied to remove noisy edges in $S_{n\times n}$. Finally, each cell and its K nearest neighbors in the denoised matrix, $E_{n\times n}$, will be connected by edges to construct $A_{n\times n}$.
- Graph Attentional Autoencoder: The graph attentional autoencoder learns cell embeddings $Z_{d\times n}$ by aggregating the features of neighbors with different weights, where d denotes the bottleneck layer size of the autoencoder. It takes two parts, feature matrix $Y_{D\times n}$ and auxiliary graph $A_{n\times n}$, as its inputs, and outputs a reconstructed matrix $Y'_{D\times n}$. The feature matrix $Y_{D\times n}$ is transformed from $X_{m\times n}$, with D as the feature dimensionality. Mean absolute error (MAE) between $Y_{D\times n}$ and $Y'_{D\times n}$ is calculated to supervise the learning process.
- Self-optimizing clustering: Given the number of clusters, c, a membership matrix $Q_{c\times n}$ is calculated to characterize the probability that a cell belongs to a cluster, update clustering centers and provide a clustering result. Then, an optimized matrix $P_{c\times n}$ is constructed as a target for $Q_{c\times n}$. KL divergence between $Q_{c\times n}$ and $P_{c\times n}$ is calculated to adjust the cell embeddings in return and promote the clustering results.

2.2 Graph construction with network denoising

To share information across similar cells and learn more clustering-friendly cell embeddings, a cell graph is required for the graph attentional autoencoder. Since there is no direct relationship between cells in scRNA-seq data, a trustworthy auxiliary graph should be constructed.

Pearson correlation coefficient is first calculated to measure the similarity between cells and construct the initial similarity matrix $S_{n\times n}$. However, since the scRNA-seq data are noisy and high-dimensional, similarity between cells can be misestimated. Incorrect similarity may seriously mislead the information sharing process and the clustering result.

Here, we adopt NE to denoise $S_{n\times n}$. Given a weighted graph, NE can increase the eigengap while preserving the eigenvectors of its adjacency matrix, thus leading to better detection of clusters (Wang *et al.*, 2018). As a result, weights of edges connecting dissimilar cells will be weakened.

Specifically, we perform NE to $S_{n\times n}$ to obtain a re-weighting similarity matrix $\hat{E}_{n\times n}$, and calculate the denoised similarity matrix $E_{n\times n}$ as

$$E_{ij} = \begin{cases} S_{ij} & \text{if } \hat{E}_{ij} \geq t \\ 0 & \text{otherwise} \end{cases}, \qquad (1)$$

where t is a pre-defined threshold. If $\hat{E}_{ij}$ is too small (smaller than t), S_{ij} will be treated as noise and E_{ij} will be set to zero.

For each cell, we select the K most similar cells as its neighbors based on $E_{n\times n}$. Each pair of neighbors will be connected by an edge in $A_{n\times n}$.

2.3 Graph attentional autoencoder

Inspired by graph attention networks (GAT) (Veličković *et al.*, 2018), we designed a graph attentional autoencoder, which composes of an encoder containing two stacked graph attentional layers and a structurally symmetric decoder, to embed topological information into the latent representation of cells.

Given a cell graph, the graph attentional layer learns a cell's feature by aggregating features of its neighbors with different weights. Since the weights are automatically assigned according to the features of a cell and its neighbors, latent relationship can be naturally captured. Therefore, the learned feature can be more clustering-friendly.

To be specific, the function of the graph attentional layer can be formulated as:

$$\mathbf{h}'_i = \sigma\left(\sum_{j\in N_i} \alpha_{ij}\mathbf{W}\mathbf{h}_j\right), \qquad (2)$$

where $\mathbf{h}'_i$ is the new feature of cell i, N_i is the set of neighbors of cell i, $\mathbf{h}_j$ is the input feature of cell j, $\mathbf{W}$ is a learnable transformation

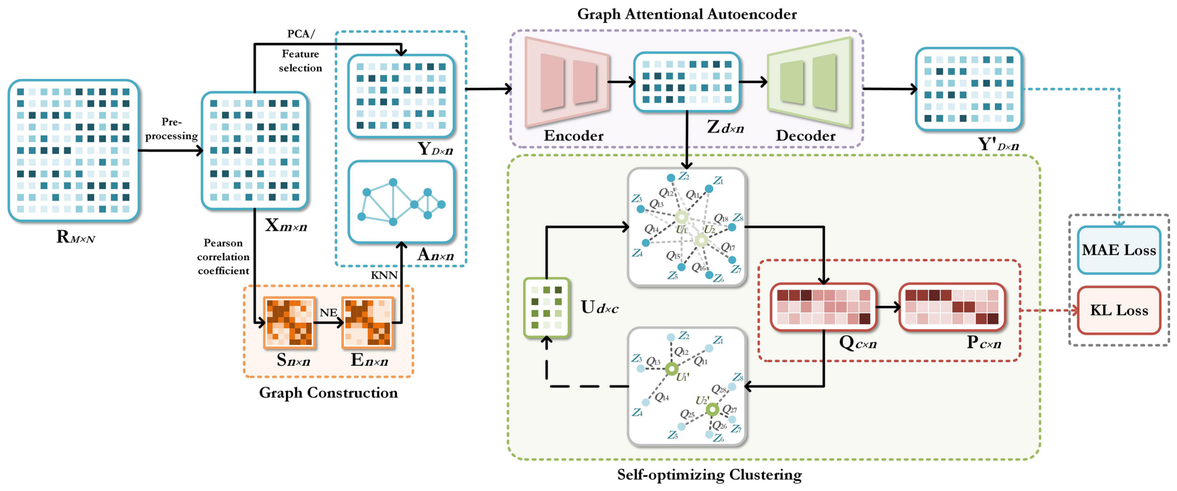

Fig. 1. Architecture of scGAC. First, given the expression matrix $R_{M\times N}$, pre-processing is performed to obtain a normalized matrix $X_{m\times n}$. Second, scGAC measures an initial cell–cell similarity matrix $S_{n\times n}$ based on Pearson correlation coefficient and denoises it by NE. A cell graph $A_{n\times n}$ is then constructed by selecting K nearest neighbors for each cell. Besides, $X_{m\times n}$ is reduced to $Y_{D\times n}$ to accelerate the learning process. Third, a graph attentional autoencoder learns embedding of cells, considering of both gene expression feature and cell–cell relationship feature. It takes both $Y_{D\times n}$ and $A_{n\times n}$ as its inputs and produces a reconstructed feature matrix $Y'_{D\times n}$. Finally, the self-optimizing clustering module measures the similarity between cells and clustering centers by a membership matrix $Q_{c\times n}$, and redistributes the membership by an optimized matrix $P_{c\times n}$. Then, $Q_{c\times n}$ is utilized to update clustering centers and provide a clustering result

matrix, and σ is a non-linear activation function. α_{ij} is the weight coefficient, representing the importance of cell j to cell i.

To measure the importance of a cell to another cell, the original GAT (Veličković *et al.*, 2018) concatenates features of them to obtain the attention coefficient e_{ij}, which can be expressed as:

$$e_{ij} = \mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i||\mathbf{W}\mathbf{h}_j]), \tag{3}$$

where LeakyReLU is a non-linear function, $\mathbf{a}$ is a learnable weight vector, and $||$ is the concatenation operation.

Different from GAT, we explicitly integrate similarity information into the attention coefficient, which is calculated by transforming the distance between two cells by a Gaussian kernel:

$$e_{ij} = \exp(-|\mathbf{a}_1^T\mathbf{W}\mathbf{h}_i - \mathbf{a}_2^T\mathbf{W}\mathbf{h}_j|^2), \tag{4}$$

where $\mathbf{a}_1$ and $\mathbf{a}_2$ are the learnable weight vectors for cell i and its neighbor j, respectively.

In general, the attention coefficient will be normalized by a softmax function to be comparable across different cells, which is expressed as:

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in N_i}\exp(e_{ik})}. \tag{5}$$

Similar to GAT, we employ multi-head attention to stabilize the learning process. G independent attention modules are used to jointly learn the representation, which can be formulated as:

$$\mathbf{h}'_i = \oplus_{g=1}^{G}\sigma\left(\sum_{j\in N_i}\alpha_{ij}^{g}\mathbf{W}^{g}\mathbf{h}_j\right), \tag{6}$$

where $\oplus$ is an aggregation operation. The concatenation and averaging function are used as aggregation operation for the first three layers and the last layer of the autoencoder, respectively.

In implementation, $Y_{D\times n}$ and $A_{n\times n}$ are fed into the graph attentional autoencoder, providing gene-level expression information and cell-level topological information, respectively. To constrain the learning process, MAE between the reconstruction feature matrix $Y'_{D\times n}$ and the original input matrix $Y_{D\times n}$ is calculated as the reconstruction loss:

$$L_r = \sum_{i=1}^{D}\sum_{j=1}^{n}|Y_{ij} - Y'_{ij}|. \tag{7}$$

2.4 Self-optimizing clustering

After pre-training the graph attentional autoencoder, the latent representation matrix, $Z_{d\times n}$, can characterize cells in the aspects of both cell–cell relationship and gene expression. A simple clustering result can be obtained by performing k-means (Hartigan and Wong, 1979) on $Z_{d\times n}$. However, this result can be suboptimal, since the clustering module does not interact with the feature learning module. Hence, we adopt an iteratively self-optimizing clustering method to allow the two modules to benefit from each other and finally improve the result.

First, we measure the similarity between cells and cluster centers (initialized by k-means) based on student's t-distribution like in DEC (Xie *et al.*, 2016). After normalized in cellular level, a membership matrix, $Q_{c\times n}$, can be calculated as:

$$q_{ij} = \frac{(1+\|\mathbf{z}_i - \mathbf{u}_j\|^2)^{-1}}{\sum_{k=1}^{c}(1+\|\mathbf{z}_i - \mathbf{u}_k\|^2)^{-1}}, \tag{8}$$

where $\mathbf{z}_i$ is the embedding of cell i, $\mathbf{u}_j$ is the embedding of cluster center j, and c is the number of cell types.

Then, an optimized membership matrix, $P_{c\times n}$, is constructed based on $Q_{c\times n}$, which is defined as:

$$p_{ij} = \frac{q_{ij}^2/\sum_{i=1}^{n}q_{ij}}{\sum_{k=1}^{c}(q_{ik}^2/\sum_{i=1}^{n}q_{ik})}. \tag{9}$$

It is used as the target for $Q_{c\times n}$, to optimize and redistribute the membership. After the redistribution, membership distribution of cells will be more deterministic due to the square operation. Additionally, clusters with lower total membership ($\sum_{i=1}^{n}q_{ij}$) will obtain relatively higher membership.

The embedding of cluster centers and cells are then updated based on $Q_{c\times n}$ and $P_{c\times n}$, to achieve better clustering result. On one hand, to better characterize the clusters, embedding of cluster centers is updated by a weighted average of cells, with $Q_{c\times n}$ as the weight, which is calculated as:

$$\mathbf{u}_j = \frac{\sum_{l_i=j}q_{ij}\mathbf{z}_i}{\sum_{l_i=j}q_{ij}}, \tag{10}$$

where l_i is the cluster label of cell i in this iteration, which is defined as:

$$l_i = \arg\max_j q_{ij}. \tag{11}$$

Table 1. Details of the 16 evaluated datasets

Dataset	Description	Cells	Genes	Cell types	Platform	Reference
Yan	Human early embryos and embryonic stem cells	90	18 042	6	Tang	Yan *et al.* (2013)
Biase	Mouse embryos	49	21 489	3	Smart-Seq	Biase *et al.* (2014)
Klein	Mouse embryonic stem cells	2717	24 021	4	inDrop	Klein *et al.* (2015)
Romanov	Mouse hypothalamus	2863	18 496	7	Drop-Seq	Romanov *et al.* (2017)
Muraro	Human pancreas	2089	17 417	9	CEL-Seq2	Muraro *et al.* (2016)
Björklund	Human tonsil ILCs	647	26 087	4	Smart-Seq2	Björklund *et al.* (2016)
PBMC[a]	Human peripheral blood mononuclear cells	5356	14 219	4	Chromium	—
Zhang	Human T cells	8496	12 547	20	Smart-Seq2	Zhang *et al.* (2018)
Guo	Human T cells	9051	12 415	16	Smart-Seq2	Guo *et al.* (2018)
Brown.1	Human dendritic cells	4406	14 064	7	Chromium	Brown *et al.* (2019)
Brown.2	Human melanoma cells	8612	15 292	7	Chromium	Brown *et al.* (2019)
Chung	Single cells from breast cancer	515	27 420	5	Smart-Seq	Chung *et al.* (2017)
Sun.1	Mouse lung cells	1669	995	6	Chromium	Sun *et al.* (2019)
Sun.2	Human skin cells	4717	999	8	Chromium	Sun *et al.* (2019)
Sun.3	Human peripheral blood mononuclear cells	8197	1000	7	Chromium	Sun *et al.* (2019)
Habib	Nuclei from mouse archived brain	10 788	19 260	10	Drop-Seq	Habib *et al.* (2017)

[a]Dataset PBMC is obtained from https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc6k.

On the other hand, to supervise the learning of latent representations and enhance the underlying clustering structure, the KL divergence between $Q_{c\times n}$ and $P_{c\times n}$ is calculated as the clustering loss:

$$L_c = \sum_{i=1}^{c}\sum_{j=1}^{n} p_{ij}\log\frac{p_{ij}}{q_{ij}}. \qquad (12)$$

In the training process, we calculate silhouette score (Rousseeuw, 1987) based on latent representations to monitor the clustering performance for early stopping. The silhouette score of cell i is calculated as:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}, \qquad (13)$$

where a_i is the mean distance between cell i and cells in the same cluster, and b_i is the mean distance between cell i and cells in different clusters. Silhouette score of a clustering result is the average silhouette score across all the cells. The latent representation of cells and cluster centers will be iteratively fine-tuned until the silhouette score converges.

In summary, the total loss function is calculated as:

$$L = L_r + \gamma L_c, \qquad (14)$$

where L_r is the reconstruction loss, L_c is the clustering loss and γ is a hyper-parameter that balances two losses. The loss function integrates latent representation learning and clustering into a unified framework, thus promoting the final clustering result.

2.5 Datasets and pre-processing

As shown in Table 1, 16 real datasets are obtained for evaluation. These datasets range in size from dozens to thousands and come from different tissues and platforms. For instance, dataset Biase (Biase *et al.*, 2014) only contains 49 cells with 3 cell types, including zygotes, 2-cell mouse embryos and 4-cell mouse embryos, while dataset Habib (Habib *et al.*, 2017) contains 10 788 mouse nuclei with 10 cell types. Annotation of cell types from the original publications is used as ground truth.

Before clustering, the expression matrix $R_{M\times N}$ is pre-processed to $X_{m\times n}$ by performing quality control and normalization. First, low-quality cells are removed if the library size (total read counts of a cell) or the number of expressed genes exceeds the upper threshold [the 75th percentiles of all cells plus three times the interquartile range (IQR)] or falls below the lower threshold (the 25th percentiles minus three times the IQR). Besides, low-quality genes expressed in less than three cells are also removed. After quality control, the read counts are divided by library size, multiplied by 100 000, and log (base 2) transformed with a pseudo count as 1.

To accelerate the subsequent learning process, $X_{m\times n}$ is reduced to $Y_{D\times n}$ before inputting into the graph autoencoder. Principal component analysis (PCA) is used for dimension reduction as default. For small datasets ($n < D$), top D highly variable genes are selected instead, since PCA failed to run on them. Next, $Y_{D\times n}$ is normalized to z-score data with zero mean and unit variance.

2.6 Benchmark setting

2.6.1 Parameter setting

For scGAC, the default number of neighbors for graph construction is calculated as

$$K = \text{round}\left(\frac{N_{\text{cells}}}{10\times N_{\text{clusters}}}\right), \qquad (15)$$

where N_{cells} is the number of cells and N_{clusters} is the number of clusters. To ensure that the constructed graph is valid and reliable, K is limited to between 6 and 20. We set D, the dimensionality of input features, to 512. The graph attentional autoencoder consists of 4 heads, with each head containing a two-layer encoder (64 neurons for the first layer and 16 neurons for the second layer) and a two-layer decoder (64 neurons for the first layer and 512 neurons for the second layer). The loss coefficient γ is set to 1. The model is pre-trained by an Adam optimizer with learning rate as 0.0002, and is trained with learning rate as 0.0005.

2.6.2 Baseline methods

Seven state-of-art single-cell clustering methods, CIDR, SC3, SAME, RCSL, scDeepCluster, DESC and scGNN (Huh *et al.*, 2020; Kiselev *et al.*, 2017; Li *et al.*, 2020; Lin *et al.*, 2017; Mei *et al.*, 2021; Tian *et al.*, 2019; Wang *et al.*, 2021), are evaluated as baseline methods.

For CIDR, SAME and scDeepCluster, original data are provided for their subsequent pre-processing and clustering. For SAME, clustering result produced with BIC criterion is used as its solution. For SC3, both original data and log-transformed data were provided according to its requirements. For DESC, normalization, log transformation, gene selection and scaling were performed following its online tutorial (https://eleozzr.github.io/desc/tutorial.html) and clustering result produced with 'resolution' as 0.8 is used as its solution. For scGNN, data pre-processing and LTMG inferring (Wang *et al.*, 2021) were performed. The number of clusters is provided for SC3 and scDeepCluster. Parameters of all the methods were set as default or following their tutorials.

2.6.3 Analysis of cell embeddings

For deep learning-based methods, including scGAC, scDeepCluster, DESC and scGNN, latent representation in the bottleneck layer of the autoencoder (feature autoencoder for scGNN) is obtained as the cell embedding. For CIDR, the dissimilarity matrix is obtained. For RCSL, the block-diagonal matrix is obtained. For SAME and SC3, since there are no suitable cell embeddings, embedding analysis is not performed.

To quantify the intra-cluster purity and inter-cluster differentiation of cell embeddings, silhouette score [defined in Equation (13)] is calculated based on the annotated cell type labels.

For visualization, the obtained embeddings will be reduced to 50 dimensions by PCA if its dimensionality exceeds 50. Next, it will be further reduced to 2 dimensions by t-Distributed stochastic neighbor embedding (t-SNE) (van der Maaten and Hinton, 2008) for demonstration. Points in the 2D visualization plane will be colored according to their annotated cell types.

2.6.4 Evaluation metrics

Two metrics, Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and Normalized Mutual Information (NMI) (Strehl and Ghosh, 2002), are used to evaluate the clustering performance of scGAC and baseline methods.

ARI measures the similarity between the clustering result and the ground truth, which is calculated as:

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}, \tag{16}$$

where n_{ij} denotes the number of shared cells between cluster i in ground truth and cluster j in clustering results, n denotes the number of all the cells, $a_i = \sum_j n_{ij}$ and $b_j = \sum_i n_{ij}$.

NMI measures the normalized mutual information between the clustering result and the ground truth, which is calculated as:

$$NMI = \frac{2MI(U,V)}{H(U)+H(V)}, \tag{17}$$

where U is clustering result, V is ground truth, $MI(U, V)$ represents the mutual information between U and V, and $H(\cdot)$ is entropy function.

For fair comparison, scGAC and baseline methods were repeatedly run for 10 times to calculate the average ARI and NMI for evaluation.

3 Results

3.1 scGAC achieves excellent clustering performance and outperforms existing methods

We applied scGAC on 16 real datasets and compared it with 7 baseline methods. As shown in Figure 2 and Supplementary Figure S1 (see Supplementary Tables S1 and S2 for details), overall, scGAC has the best performance in terms of both ARI and NMI scores. On nine of the sixteen datasets, including Biase, Romanov, Björklund, PBMC, Brown.1, Brown.2, Chung, Sun.3 and Habib, scGAC outperforms all baseline methods with highest ARI scores. On six of the rest seven datasets, scGAC performs the second-best as well. Additionally, scGAC has the highest average score across all the datasets based on both ARI (Fig. 2 and Supplementary Table S1) and NMI score (Supplementary Fig. S1 and Table S2).

3.2 scGAC learns clustering-friendly embedding of cells

To see whether scGAC learns more clustering-friendly cell embeddings, we obtained the learned embeddings of scGAC and 5 baseline methods and compared their intra-cluster purity and inter-cluster differentiation with the original features of cells in the input expression matrix based on silhouette score.

As shown in Figure 3a, scGAC has the highest silhouette scores on half of the 16 evaluated datasets and outperforms the original features

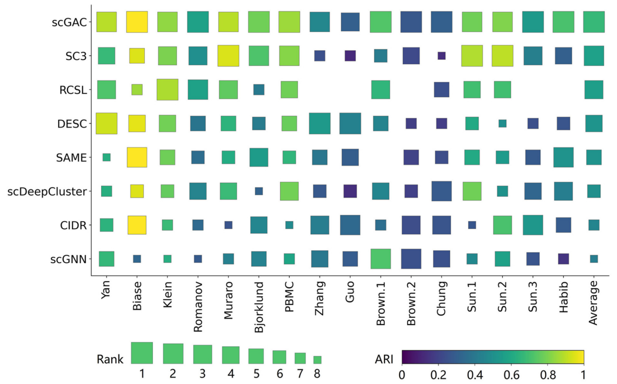

Fig. 2. ARI scores of scGAC and 7 baseline methods on 16 evaluated datasets. A block represents the performance of a method on a dataset, where size indicates the rank and color indicates the ARI score. The last column shows average ARI score of each method. In some places there is no block, since RCSL failed to finish within 5 days on 5 datasets and SAME failed to run on dataset Brown.1

on almost all the datasets. Specifically, on datasets Yan, Biase, Klein, Björklund, Sun.1, Sun.2 and Sun.3, scGAC achieves significantly higher silhouette scores than other methods. Additionally, scGAC has the highest average silhouette score across all datasets, outperforming the second best method, DESC, by 30.9%. The result suggests that cell embeddings learned by scGAC are more distinguishable across different cell types and highly coherent within the same cell type.

Notably, apart from scGAC, other two methods, DESC and scDeepCluster, which simultaneously perform clustering while learning embedding of cells, also obtain the second and third highest silhouette scores on average. It means the interaction of embedding learning and clustering effectively improves the clustering features of data.

Besides, to intuitively observe the latent representations, we visualized the cell embeddings on datasets Björklund and PBMC by t-SNE, in Figure 3b and c, respectively.

Dataset Björklund contains an NK subset and three homogenous innate lymphoid cell (ILC) subsets that are not easy to separate. As shown in Figure 3b, only scGAC gets four pure clusters with quite clear boundaries, which is much better than other methods. For scGNN, the four clusters are pure but not distinct. For RCSL, cells of the same type are divided apart while cells of different types (NK cells and ILC3) are mixed. For CIDR, scDeepCluster and DESC, different types of cells, especially the ILCs, are mixed together.

Dataset PBMC comprises four types of immune cells, including monocytes, NK cells, B cells and T cells, which are easier to cluster. As shown in Figure 3c, scGAC, scDeepCluster and DESC show relatively good visualization, while CIDR mixes cells together, and RCSL and scGNN divided NK cells or B cells into subclusters. Separating NK cells from T cells is difficult for all methods. Although scGAC produces only three clusters, with NK cells and T cells closing together, the NK cells in scGAC are purer than those in scDeepCluster and DESC.

3.3 Graph attentional autoencoder improves clustering performance

To obtain better clustering results, we use a graph attentional autoencoder to learn the embedding of cells while leveraging topological relationship between cells. The graph attentional autoencoder enables flexible information sharing between neighbors in the graph, thus making the embedding more clustering-friendly.

To explore the effect of information sharing in scGAC, we also run scGAC with the number of neighbors, K, set to 1, which means no cell can pass information to a cell except itself. Supplementary Table S3 shows ARI scores on 16 datasets for scGAC with K set to 1 (scGAC_no_neighbor) and set as default (scGAC), respectively. scGAC outperforms scGAC_no_neighbor with significantly higher ARI scores on most datasets and has a 42.3% improvement on average, suggesting that preventing information from sharing heavily hampers clustering. Therefore, leveraging the relationship between cells is important and

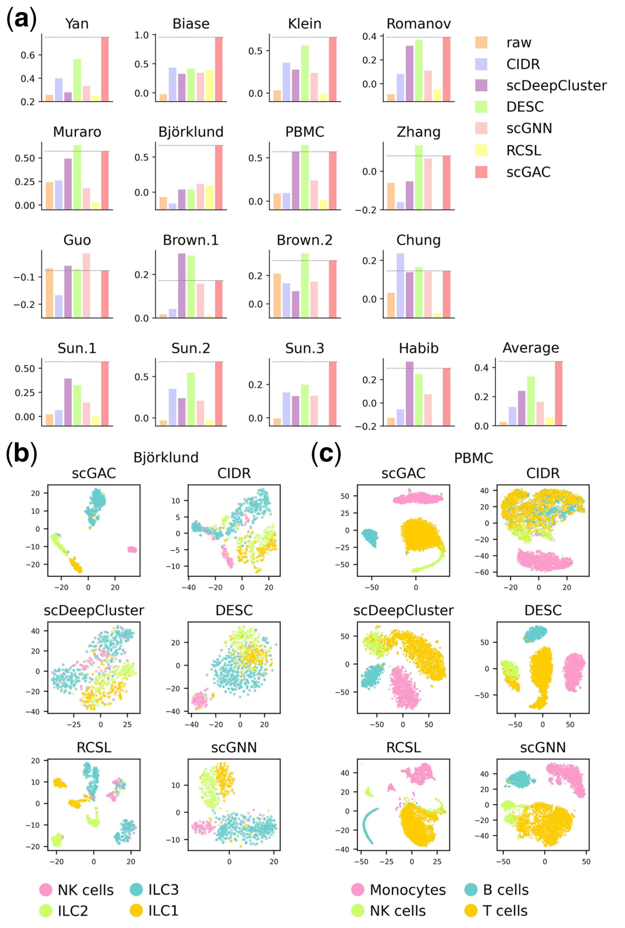

Fig. 3. Analysis of cell embeddings. (a) Silhouette scores of original features of cells in the raw expression matrix ('raw') and cell embeddings produced by six clustering methods. The last panel shows their average silhouette scores across all datasets. (b) Visualization of cell embeddings in dataset Björklund. (c) Visualization of cell embeddings in dataset PBMC

can benefit clustering. However, it is worth noting that over-sharing of information should be avoided. Sharing feature information across cells of different cell types will disturb clustering, since characteristics of different cell types will be obscured.

We use attention mechanism to ease the above problem. Because of the graph attentional layers in the autoencoder, importance of different neighbors to a cell can be automatically discriminated. Therefore, a cell can selectively receive information from different neighbors. To evaluate the benefits of attention mechanism, we also run scGAC with constant attention (scGAC_const_attention). All neighbors will be assigned to a same constant weight, thus contributing equally to feature learning process. As shown in Figure 4a, on most datasets, scGAC has higher ARI scores than scGAC_const_attention. Additionally, scGAC has a 1.4% improvement over scGAC_const_attention on the average ARI score. Furthermore, to intuitively demonstrate the importance of attention mechanism, we visualized embeddings of them on dataset PBMC. As shown in Figure 4b, without attention mechanism, the NK and T cells are more difficult to separate, suggesting that receiving all information without selection may mislead feature learning.

In addition, the implementation mechanism of attention also influences clustering results. Different from the original GAT, which implements attention mechanism with a concatenation operation, scGAC explicitly integrates similarity information between cells into attention calculation and may improve the learned features. Therefore, we also evaluated the concatenation attention mechanism (scGAC_concat_attention) for comparison. As shown in Figure 4a, scGAC outperforms scGAC_concat_attention on most

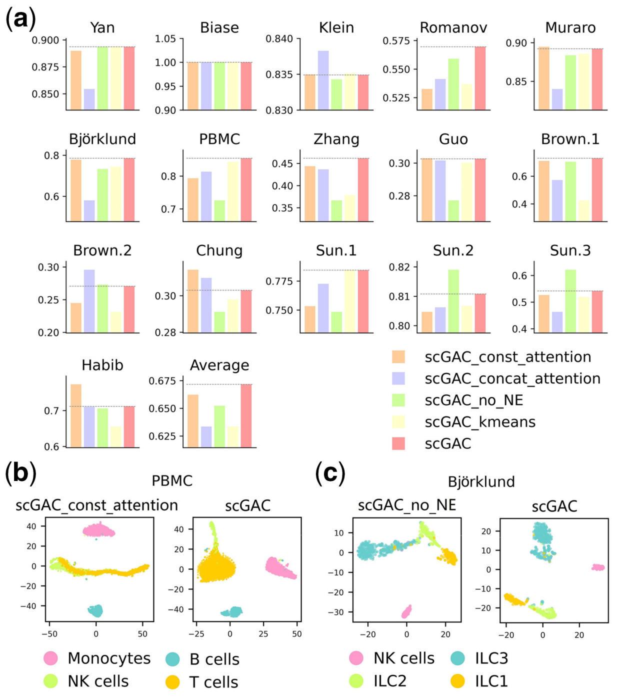

Fig. 4. Comparison of scGAC with different settings. (a) ARI scores of scGAC_const_attention, scGAC_concat_attention, scGAC_no_NE, scGAC_kmeans and scGAC on 16 datasets. The last panel shows the average ARI score over 16 datasets. (b) Visualization of scGAC_const_attention and scGAC on dataset PBMC. (c) Visualization of scGAC_no_NE and scGAC on dataset Björklund

datasets, and has a 6% improvement on the average ARI score. Therefore, integrating similarity information into attention mechanism indeed benefits single-cell clustering.

3.4 NE benefits construction of cell graph

As mentioned in the previous section, information sharing across cells is crucial to clustering. Whether it benefits or not also depends on the reliability of the constructed cell graph, since information will only be shared between cells connected in the graph. Therefore, NE is adopted to denoise and improve the cell graph.

To see whether NE benefits the construction of cell graph, we calculated the proportion of correct edges (connecting cells of the same cell type) in cells graphs constructed with or without NE, respectively (Supplementary Table S4). Higher proportion indicates higher reliability of the cell graph. On most datasets, cell graph denoised by NE has a higher proportion of correct edges. For simple datasets, such as datasets Yan and Biase, different cell types are so distinct that cell graph constructed without NE is reliable enough. However, for more complex datasets where different cell types are hard to separate, NE brings significant improvements. Specifically, NE brings an improvement of 8.6% for dataset Zhang, an improvement of 6.5% for dataset Björklund and an improvement of 5.3% for dataset Guo.

Besides, we also applied scGAC without introducing NE (scGAC_no_NE) on the evaluated datasets. As shown in Figure 4a, on most datasets, especially those with a higher improvement of proportion of correct edges (Supplementary Table S4), such as datasets Björklund, PBMC, Zhang and Guo, scGAC with NE achieves significantly higher ARI scores than scGAC without NE. As a whole, NE facilitates scGAC to achieve an improvement of 3% on the average ARI score.

Furthermore, to intuitively demonstrate the effectiveness of NE, we visualized the cell embeddings of dataset Björklund that were learned with and without NE, respectively (Fig. 4c). When compared with scGAC without NE, the ILCs can be clearly divided into three clusters in scGAC, which means NE effectively removes noisy edges in the cell graph. Therefore, NE indeed helps to construct a more reliable cell graph and improve clustering results.

3.5 Self-optimizing clustering module promotes clustering

Apart from the graph attentional autoencoder, another crucial component of scGAC is the self-optimizing clustering module, which provides the final clustering result for scGAC. It optimizes clustering centers and redistributes the membership to make it more deterministic. Meanwhile, it also interacts with the graph attentional autoencoder and helps to update the learned embeddings in return.

To see whether the self-optimizing clustering module improves clustering results, we demonstrated the clustering result before self-optimizing clustering (scGAC_kmeans) in Figure 4a, which is obtained by only performing *k*-means on the pre-trained embeddings. When compared with scGAC with only *k*-means, scGAC has higher ARI scores on almost all datasets, especially those with lower ARI scores before self-optimizing clustering, suggesting the effectiveness of self-optimizing clustering. For example, after self-optimizing clustering, the ARI score is improved by 71.8% on dataset Brown.1, 21.8% on dataset Zhang and 17% on dataset Brown.2. Therefore, the clustering solution provided by *k*-means is suboptimal and can be improved by the additional self-optimizing clustering module.

4 Discussion

Clustering is a crucial step for subsequent analysis of scRNA-seq data. However, clustering single cells suffer from the high-dimensional, highly variable and noisy data. In this article, we propose scGAC, a novel single-cell clustering method. scGAC learns clustering-friendly embedding of cells by sharing information across similar cells with attention mechanism, and improves the clustering results with NE and self-optimizing clustering.

Our experimental results show that scGAC achieves excellent clustering performance on 16 datasets and outperforms 7 state-of-art clustering methods specially designed for scRNA-seq data. Additionally, different components of scGAC have been proved to be significant and beneficial for scGAC. Besides, scGAC shows excellent generalizability on real datasets, regardless of the size, the number of cell types and the difference between cell types.

One of the limitations of scGAC is its efficiency. scGAC has a medium running efficiency on the evaluated datasets, inferior to DESC, CIDR and SC3 (Supplementary Table S5). Notably, other methods considering of pairwise cell–cell relationship, such as scGNN and RCSL, show low efficiency as well, since the run time will dramatically increase with the number of cells. Therefore, to handle larger datasets, we will consider to accelerate the learning process while leveraging relationship between cells in the future.

Acknowledgement

We acknowledge the support by High-performance Computing Platform of Peking University.

Funding

This work was supported by a grant from the National Natural Science Foundation of China [31972899].

Conflict of Interest: none declared.

Data Availability

A python implementation of scGAC is available at Github (https://github.com/Joye9285/scGAC). All datasets tested in this manuscript can be accessed through the following accession numbers or websites: Yan (GSE36552); Biase (GSE57249); Klein (GSE65525); Romanov (GSE74672); Muraro (GSE85241); Bj—rklund (GSE70580); P BMC (https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc6k); Zhang (GSE108989); Guo (GSE99254); Brown.1 and Brown.2 (GSE137710); Chung (GSE75688); Sun.1, Sun.2 and Sun.3 (GSE128066 for original data and https://github.com/CHP Genetics/BAMMSC for processed data); and Habib (GSE104525).

References

Biase,F.H. *et al.* (2014) Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing. *Genome Res.*, **24**, 1787–1796.

Björklund,Å.K. *et al.* (2016) The heterogeneity of human CD127$^+$ innate lymphoid cells revealed by single-cell RNA sequencing. *Nat. Immunol.*, **17**, 451–460.

Bo,D. *et al.* (2020). Structural deep clustering network. In: *Proceedings of the Web Conference*, Taipei, pp. 1400–1410.

Brown,C.C. *et al.* (2019) Transcriptional basis of mouse and human dendritic cell heterogeneity. *Cell*, **179**, 846–863.

Chung,W. *et al.* (2017) Single-cell RNA-seq enables comprehensive tumour and immune cell profiling in primary breast cancer. *Nat. Commun.*, **8**, 15081.

Guo,M. *et al.* (2019) Single cell RNA analysis identifies cellular heterogeneity and adaptive responses of the lung at birth. *Nat. Commun.*, **10**, 37.

Guo,X. *et al.* (2018) Global characterization of T cells in non-small-cell lung cancer by single-cell sequencing. *Nat. Med.*, **24**, 978–985.

Habib,N. *et al.* (2017) Massively parallel single-nucleus RNA-seq with DroNc-seq. *Nat. Methods*, **14**, 955–958.

Hartigan,J.A. and Wong,M.A. (1979) Algorithm AS 136: a k-means clustering algorithm. *J. R. Stat. Soc. C (Appl. Stat.)*, **28**, 100–108.

Hinton,G.E. and Salakhutdinov,R.R. (2006) Reducing the dimensionality of data with neural networks. *Science*, **313**, 504–507.

Hubert,L. and Arabie,P. (1985) Comparing partitions. *J. Classif.*, **2**, 193–218.

Huh,R. *et al.* (2020) SAME-clustering: single-cell aggregated clustering via mixture model ensemble. *Nucleic Acids Res.*, **48**, 86–95.

Hwang,B. *et al.* (2018) Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.*, **50**, 1–14.

Kipf,T.N. and Welling,M. (2017). Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations*, Toulon, France, pp. 1–14.

Kiselev,V.Y. *et al.* (2017) SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods*, **14**, 483–486.

Kiselev,V.Y. *et al.* (2019) Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat. Rev. Genet.*, **20**, 273–282.

Klein,A.M. *et al.* (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**, 1187–1201.

Li,X. *et al.* (2020) Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nat. Commun.*, **11**, 2338.

Lin,P. *et al.* (2017) CIDR: ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol.*, **18**, 59.

Mei,Q. *et al.* (2021) Clustering single-cell RNA-seq data by rank constrained similarity learning. *Bioinformatics*, **37**, 3235–3242.

Muraro,M.J. *et al.* (2016) A single-cell transcriptome atlas of the human pancreas. *Cell Syst.*, **3**, 385–394.

Romanov,R.A. *et al.* (2017) Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes. *Nat. Neurosci.*, **20**, 176–188.

Rousseeuw,P.J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, **20**, 53–65.

Shalek,A.K. *et al.* (2013) Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, **498**, 236–240.

Stegle,O. *et al.* (2015) Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.*, **16**, 133–145.

Strehl,A., and Ghosh,J. (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, **3**, 583–617.

Sun,Z. *et al.* (2019) A Bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies. *Nat. Commun.*, **10**, 1649.

Tian,T. *et al.* (2019) Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nat. Mach. Intell.*, **1**, 191–198.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

Veličković,P. *et al.* (2018). Graph attention networks. In: *International Conference on Learning Representations*, Vancouver, Canada, pp. 1–12.

Wang,B. *et al.* (2018) Network enhancement as a general method to denoise weighted biological networks. *Nat. Commun.*, **9**, 3108.

Wang,J. *et al.* (2021) scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nat. Commun.*, **12**, 1882.

Xie,J. *et al.* (2016). Unsupervised deep embedding for clustering analysis. In: *International Conference on Machine Learning*, New York City, NY, USA, pp. 478–487.

Yan,L. *et al.* (2013) Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nat. Struct. Mol. Biol.*, **20**, 1131–1139.

Zhang,L. *et al.* (2018) Lineage tracking reveals dynamic relationships of T cells in colorectal cancer. *Nature*, **564**, 268–272.