Hindawi
Computational and Mathematical Methods in Medicine
Volume 2021, Article ID 6842752, 6 pages
https://doi.org/10.1155/2021/6842752

Research Article

SCMAG: A Semisupervised Single-Cell Clustering Method Based on Matrix Aggregation Graph Convolutional Neural Network

Haonan Peng, Wei Fan, Chujie Fang, Wenliang Gao, and Yuanyuan Li

School of Mathematics and Physics, Wuhan Institute of Technology, 430205 Wuhan, China

Correspondence should be addressed to Yuanyuan Li; yuanyuanli_wit@hotmail.com

Received 5 August 2021; Accepted 14 September 2021; Published 4 October 2021

Academic Editor: Sanjiban Sekhar Roy

Clustering analysis is one of the most important technologies for single-cell data mining. It is widely used in the division of different gene sequences, the identification of functional genes, and the detection of new cell types. Although the traditional unsupervised clustering method does not require label data, the distribution of the original data, the setting of hyperparameters, and other factors all affect the effectiveness of the clustering algorithm. While in some cases the type of some cells is known, it is hoped to achieve high accuracy if the prior information about those cells is utilized sufficiently. In this study, we propose SCMAG (a semisupervised single-cell clustering method based on a matrix aggregation graph convolutional neural network) that takes into full consideration the prior information for single-cell data. To evaluate the performance of the proposed semisupervised clustering method, we test on different single-cell datasets and compare with the current semisupervised clustering algorithm in recognizing cell types on various real scRNA-seq data; the results show that it is a more accurate and significant model.

1. Introduction

Analysis on the gene expression matrix of the single-cell dataset is the critical step to obtain a single-cell type [1–3]. The categories of cells are already unknown. Detecting the type of each single-cell manually will take a lot of time and money. Then, how to obtain the best results of classification through applying a semisupervised learning algorithm effectively and using the single-cell type as little as possible is a research direction worthy of exploration [4, 5].

The current common semisupervised learning algorithms mainly contain generative semisupervised models [6], self-training [7], collaborative training (Co-training) [8], semisupervised support vector machines (S3VMs) [9], and methods based on graph theory [10, 11]. Generative semisupervised models use the unlabeled data to make an attribution according to the distribution generated by the previously labeled data and modify the previous model parameters to better adjust the decision boundary [12], then iterate this process to optimize the model. Self-training uses existing label data to train a classifier and then uses this classifier to classify unlabeled data to generate pseudolabels or soft labels [13], then develops certain criteria for judging and selects the correct label data from the original pseudolabel data and adds it to the classifier for training, and finally iterates to produce the final classification results. Co-training is a kind of self-training, in which the algorithm assumes that each data can be classified from different perspectives and then uses these classifiers trained from different perspectives to classify unlabeled samples and selects those that are considered credible to be added to the training set. Since these classifiers are trained from different perspectives, they can complement each other and improve the accuracy of the classification. Supervised support vector machines use structural risk minimization for classification [14], and semisupervised support vector machines also use spatial distribution information for unlabeled data [15]. Among them, the selection of decision-making hyperplanes should focus on the place where the distribution of low-density unlabeled data and label data are consistent [16]. However, if this assumption is not true, the spatial distribution information of unlabeled data can mislead decision-making hyperplanes and result in worse performance than when only labeled data is used. In recent years, due to the

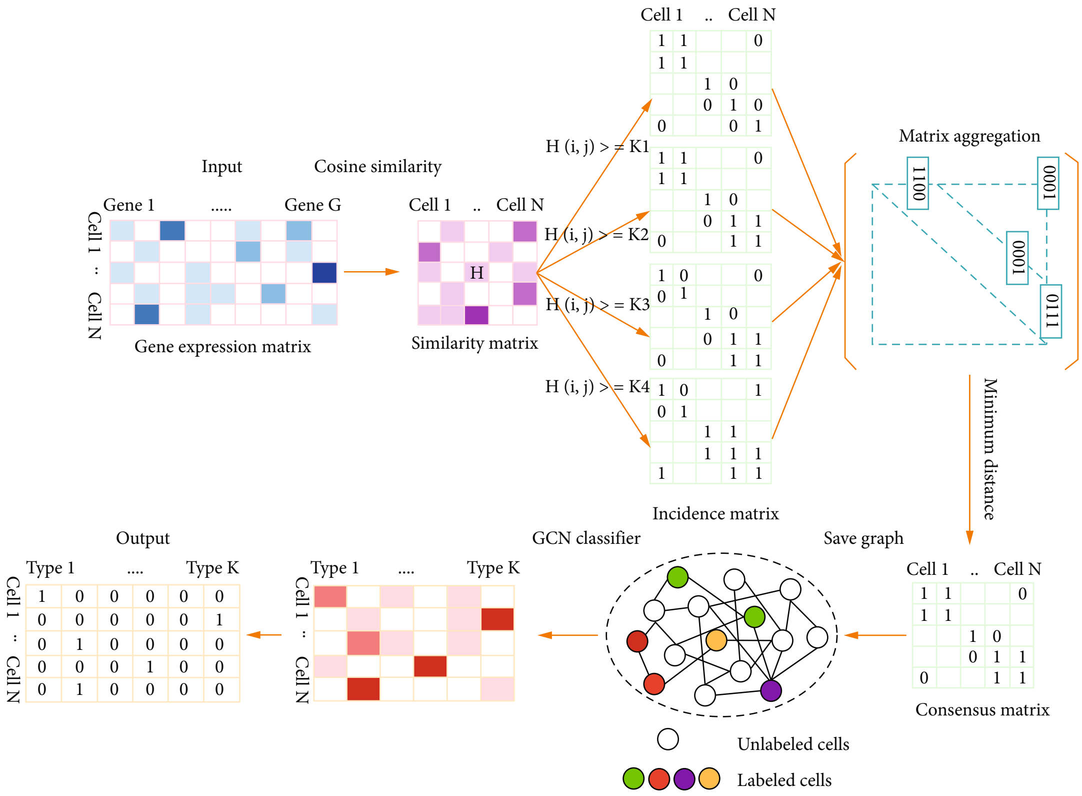


Figure 1: The workflow of the SCMAG. The input is a gene expression matrix; the algorithm includes four steps: (1) the similarity matrix is calculated by the cosine similarity formula; (2) the incidence matrix is judged by the threshold; (3) the consensus matrix is constructed by the matrix aggregation method; (4) the consensus matrix is saved as a graph; (5) lastly, the graph is used as input to the GCN classifier for training.

rise of artificial neural networks [17–19], semisupervised clustering algorithms have made breakthrough progress, among which the label propagation algorithm is one kind based on graph networks [20, 21]. In the label propagation algorithm, the connection between the labeled data and the unlabeled data is found in the training data through the construction of the graph analysis structure. Through the edge-to-edge connectivity, the labeled data flow through the unlabeled data during propagation, then use edge connections between the unlabeled data to obtain new labels and the classification results [22]. Considering that one single cell contains a large number of genes, that is to say, the characteristic dimension of each single cell is extremely high, a single classic classifier cannot learn all the high-dimensional features. Therefore, we consider using a graph convolutional neural network method to deal with high-dimensional complex connections [23–25]. The graph convolutional neural network transfers the similarity between cells to the connection relationship between the edges in the graph and then uses the convolution operation to further extract the classification features of the edges. Due to its powerful feature extraction capabilities, this algorithm shows strong performance in semisupervised clustering. However, the algorithm needs to adjust many parameters in practical applications, especially how to transform the expression matrix of genes on cells to a connection graph that can effectively reflect the similar relationship between cells is a key issue. To solve this problem, we propose SCMAG. The framework of our proposed method is presented in Figure 1. We finally demonstrate that the performance of this algorithm is better than other semisupervised clustering algorithms through tests on different datasets.

2. Materials and Methods

2.1. Data Description and Data Preprocessing. To verify the effectiveness of the method, we executed four datasets which are summarized in Table 1. These datasets are downloaded from the NCBI Gene Expression Omnibus (GEO) repository (https://www.ncbi.nlm.nih.gov/geo).

The datasets are in the form of a matrix $X(g\times n)$, which represents that there are g genes in a row and n cells in a column. Since the amount of gene expression varies greatly in each single-cell, we use min–max normalization [30] to normalize the data to (0,1):

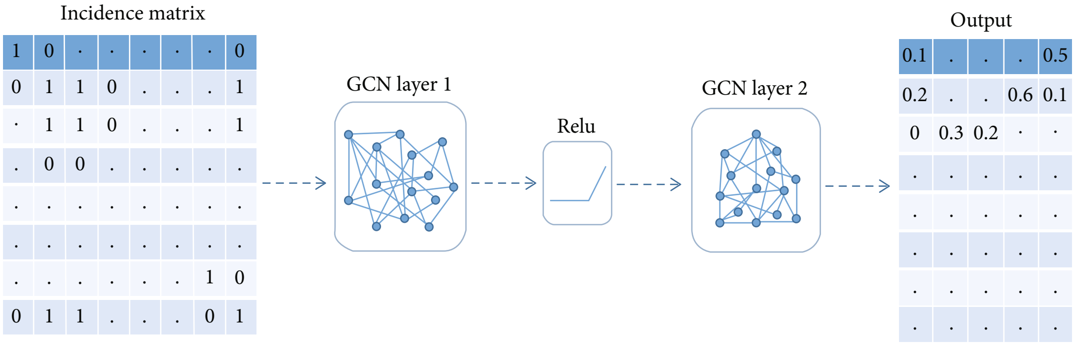
$$X_{\text{std}}(g\times n)=\frac{\left(X(g\times n)-X_{\min\ (\text{axis}=0)}\right)}{\left(X_{\max\ (\text{axis}=0)}-X_{\min\ (\text{axis}=0)}\right)}, \tag{1}$$

$$X_{\text{scaled}}(g\times n)=X_{\text{std}}(g\times n)\times(\max-\min)+\min, \tag{2}$$

where $X_{\min\ (\text{axis}=0)}$ represents the row vector composed of the minimum value in each column, $X_{\max\ (\text{axis}=0)}$ is the row

Table 1: List of datasets and their attributes.

Datasets	Number of cells	Number of cell types	Number of genes	Number of GSE	References
Chu	1018	7	19097	GSE75748	Chu et al. [26]
Patel	430	6	5948	GSE57872	Patel et al. [27]
Xin	1600	8	39851	GSE81608	Xin et al. [28]
Usoskin	622	4	25334	GSE59739	Usoskin et al. [29]



1	0	·	·	·	·	·	0
0	1	1	0	.	.	.	1
·	1	1	0	.	.	.	1
.	0	0	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	1	0
0	1	1	.	.	.	0	1

0.1	.	.	.	0.5
0.2	.	.	0.6	0.1
0	0.3	0.2	·	·
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Figure 2: Graph convolutional neural network structure.

vector composed of the maximum value in each column, max represents the maximum value of the interval to be mapped to (the default value is 1), and min represents the minimum value of the interval to be mapped to (the default value is 0). $X_{\text{std}}(g \times n)$ is the standardized result and X_{scaled} $(g \times n)$ is the normalized result, then we use cosine similarity to measure the relationship between cells [31].

$$H(i, j) = \frac{X_{\text{scaled}}(i, :) \otimes X_{\text{scaled}}(j, :)}{\|X_{\text{scaled}}(i, :)\| \times \|X_{\text{scaled}}(j, :)\|}, \tag{3}$$

where $X_{\text{scaled}}(i, \ :)$ represents the i-th row of $X_{\text{scaled}}(g \times n)$. $\otimes$ represents the inner product. $\|X_{\text{scaled}}(i, \ :)\|$ is the modulus of $X_{\text{scaled}}(i, \ :)$. $H(i, j)$ represents the value in the i-th row and j-th column of the similarity matrix $H(n \times n)$.

2.2. Data Division by Threshold. We divide H into multiple different matrices by threshold:

$$K = \{K_t = 0.1 \times t, t = 1, 2, 3, 4\}, \tag{4}$$

$$S = \{S_n, n = 1, 2, 3, 4\}, \tag{5}$$

$$S_n^{ij} = \begin{cases} 1, & H(i, j) \geq K_t, \\ 0, & H(i, j) < K_t, \end{cases} \tag{6}$$

where K_t is the threshold, S_n is the incidence matrix after threshold division, and S_n^{ij} represents the value in the i-th row and j-th column of the S_n, where 1 means that two cells are correlated and 0 means that two cells are not correlated.

2.3. Graph Convolutional Neural Network Construction. To construct a graph convolutional neural network, first of all, we should save the incidence matrix S_n as a graph $G_n(V, E$). We use the DGL package in the Python library to solve it [32]. Where the number of vertices $V_n(G)$ is equal to the number of cells, the number of edges $E_n(G)$ is equal to the number of elements in the S_n whose value is 1. Whether the two vertices in the graph are directly connected is determined by the value in the incidence matrix; the value of 1 means direct connection and 0 means no connection. Then, we build a graph convolutional neural network with two hidden layers, and its structure is shown in Figure 2.

Table 2: Accuracy under different iterations and thresholds.

K_t	Iteration			
	25	50	75	100
0.1	50.2	84.7	89.2	88.4
0.2	50.6	82.3	89.7	89.3
0.3	51.4	87.3	89.3	89.4
0.4	62.8	86.6	88.1	87.6

According to equation (4), we can get 4 initial graphs of S, and we take each S_n as the input. We randomly select 10% of the cell labels as the true labels, and the remaining 90% of the cells have no labels. In the Chu dataset, the input dimension is 1018 ∗ 1018, the activation function is ReLU, the hidden layer dimension is 256, the dimension of the final output probability matrix I_n is 1018 ∗ 7, and $I(i, j)$ represents the probability that the i-th cell belongs to the j-th type. Finally, we select $I_{\max}(i, j) = \max\{I(i, 1), I(i, 2), \cdots, I(i, j)\}$ as the output and choose j as the type of i-th cell. Table 2 shows the classification accuracy under different epochs and thresholds.

From Table 2, we can see that GCN performs well under 75 epochs. From 75 to 100 epochs, it shows the trend of convergence, and the classification accuracy is close to 90%. Then, we wonder whether there is a way to make full use of different S_n to get better performance.

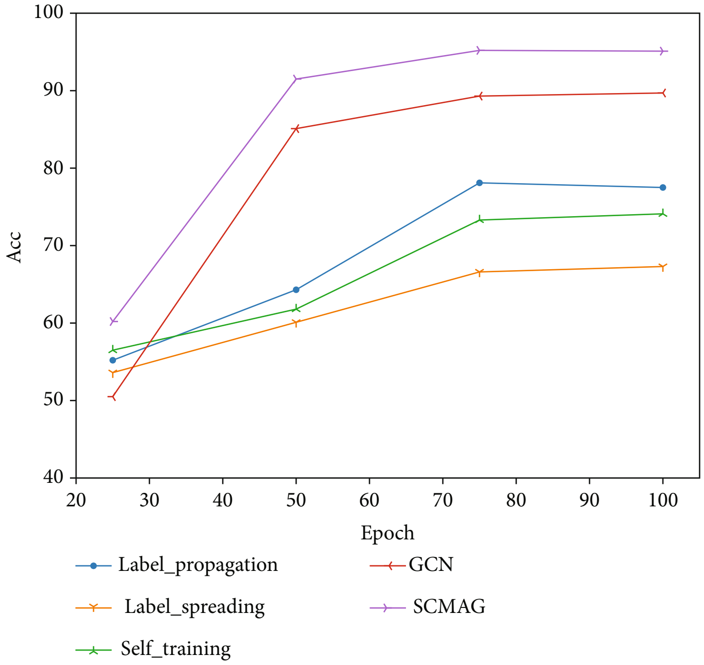


Figure 3: Accuracy under different methods.

Table 3: Performance comparison in different methods.

Dataset	Iteration	Label propagation	Label spreading	Self-training	GCN	SCMAG
	25	50.3	44.6	50.2	62.3	60.4
Patel	50	67.8	59.1	58.5	76.5	75.7
	75	69.6	65.2	61.4	78.3	79.1
	25	70.5	63.4	60.4	74.6	78.9
Xin	50	77.1	74.6	68.2	87.1	90.6
	75	79.8	75.2	72.7	89.6	91.4
	25	35.4	39.2	37.5	36.8	38.2
Usoskin	50	38.1	41.4	43.3	43.7	44.1
	75	41.6	43.2	44.8	44.9	45.2

2.4. GCN Based on Matrix Aggregation. To solve this problem, we build a consensus matrix P to minimize the distance between different thresholds [33, 34]:

$$P = \min \sum_{t=1}^{m} \sum_{j=1}^{n} \sum_{i=1}^{n} \left(P_{ij} - S_t^{ij} \right)^2, \tag{7}$$

where P_{ij} is the value of the i-th row and j-th column in the consensus matrix P. Due to the high dimension of the matrix, directly finding the minimum distance will cost a lot of time and memory. Since the values of the incidence matrix S_n^{ij} are all 0 and 1, we can convert the problem of finding the minimum distance matrix P between multiple incidence matrices S_n^{ij} into finding the number of occurrences of 0 and 1 for each S_n. We use count_0 and count_1 to count the total times of occurrences of 0 and 1.

$$P_{ij} = \begin{cases} 1, & \text{count}_1 \geq \text{count}_0, \\ 0, & \text{count}_1 < \text{count}_0. \end{cases} \tag{8}$$

We take the minimum distance matrix P as the input of graph convolutional neural network for training, then we compared it with the current commonly used semisupervised learning methods; under different epochs, the classification accuracy is shown in Figure 3.

On the Chu dataset, we found that the SCMAG showed better performance than other semisupervised methods, and we also compared it with the GCN without matrix aggregation. The result suggests that the accuracy of classification has increased by nearly 5%.

3. Experiments and Results

To further demonstrate the performance of the proposed method SCMAG, we apply the Patel, Xin, and Usoskin datasets for testing. We use label propagation, label spreading, self-training, and GCN, four classic semisupervised learning algorithms for training; then, we use SCMAG to compare with the previous four methods. After 25, 50, and 75 iterations, we get the final result, and classification accuracy is shown in Table 3.

Table 3 shows the comparison results for the Patel, Xin, and Usoskin datasets. In the Patel and Xin datasets, while the number of iterations is 25, 50, and 75, the accuracy of the GCN method is higher than that of the label propagation, label spreading, and self-training methods. When the number of iterations is small, the accuracy of the SCMAG method is lower than that of the GCN, but as the number of iterations increases, the accuracy of the SCMAG method gradually approaches and finally exceeds GCN. In the Usoskin dataset, the label spreading method has the highest accuracy after 25 iterations, followed by SCMAG. But when the number of iterations increases, the performance of GCN is better than the previous three methods. It is worth noting that SCMAG has the highest accuracy rate among the five methods. Therefore, SCMAG is the best method for cell identification.

4. Conclusion

Single-cell RNA sequencing technology has made a great contribution to the identification of single-cell types, but single-cell datasets often have a large amount of data and high dimensionality. It usually takes a lot of time to identify them. So whether other cell labels can be measured with only part of single-cell data labels is a direction worthy of research. In recent years, some semisupervised learning methods have begun to be used for single-cell data analysis.

In this study, we have proposed SCMAG for the classification of cells. Compared with the conventional graph convolutional neural network, we divide the similarity matrix by different thresholds to get different incidence matrices, and then, we construct a minimum distance matrix, and it can make full use of the high-dimensional information in the cells and better reflect the characteristics of the cells. We also test the cell classification accuracy of several commonly used semisupervised learning methods, label propagation, label spreading, self-training, and normal GCN under the same conditions. We found that SCMAG shows the best average performance in classification accuracy compared to the other four competing approaches.

Although SCMAG makes considerable improvement on identifying cell types, there remains room for improvement. Several problems are still open. For example, when the single-cell dataset contains a large number of cells, it will cost a lot of time to save the incidence matrix as a graph, and the division of threshold is also a question worth studying. In the future work, we will focus on these questions and hope to achieve more promising results.

Data Availability

The datasets supporting the conclusions of this article are available in the GEO database repository under accession numbers GSE75748, GSE57872, GSE81608, and GSE59739. The Python codes for our SCMAG method are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by the National Nature Science Foundation of China under Grant No. 12001408 and the Science Foundation of the Wuhan Institute of Technology under Grant No. K201746 and by the Graduate Innovative Fund of Wuhan Institute of Technology under Grant No. CX2020292.

References

[1] M. J. Bissell, H. G. Hall, and G. Parry, "How does the extracellular matrix direct gene expression?," *Journal of Theoretical Biology*, vol. 99, no. 1, pp. 31–68, 1982.

[2] F. Barry, R. E. Boynton, B. Liu, and J. M. Murphy, "Chondrogenic differentiation of mesenchymal stem cells from bone marrow: differentiation-dependent gene expression of matrix components," *Experimental Cell Research*, vol. 268, no. 2, pp. 189–200, 2001.

[3] Y. Li, P. Luo, Y. Lu, and F.-X. Wu, "Identifying cell types from single-cell data based on similarities and dissimilarities between cells," *BMC Bioinformatics*, vol. 22, no. S3, p. 255, 2021.

[4] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," in *A review of machine learning techniques for processing multimedia content*, pp. 9–16, 2004.

[5] R. Qi, J. Wu, F. Guo, L. Xu, and Q. Zou, "A spectral clustering with self-weighted multiple kernel learning method for single-cell RNA-seq data," *Briefings in Bioinformatics*, vol. 22, no. 4, 2021.

[6] A. Odena, "Semi-supervised learning with generative adversarial networks," 2016, https://arxiv.org/abs/1606.01583.

[7] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, pp. 29–36, Breckenridge, CO, USA, 2005.

[8] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory - COLT' 98*, pp. 92–100, Colorado, 1998.

[9] M. Chi and L. Bruzzone, "Semisupervised classification of hyperspectral images by SVMs optimized in the primal," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 6, pp. 1870–1880, 2007.

[10] F. Tian, B. Cao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *Proceedings of the*

AAAI Conference on Artificial Intelligence, pp. 1293–1299, Toronto, 2014.

[11] S. E. Schaeffer, “Graph clustering,” *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[12] Chulhee Lee and D. A. Landgrebe, “Decision boundary feature extraction for neural networks,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 75–83, 1997.

[13] H. Wu and S. Prasad, “Semi-supervised deep learning using pseudo labels for hyperspectral image classification,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1259–1270, 2018.

[14] V. Vapnik, *Principles of risk minimization for learning theory*, pp. 831–838, Advances in neural information processing systems, 1992.

[15] G. Valentine, “Images of danger: women's sources of information about the spatial distribution of male violence,” *Area*, vol. 24, no. 1, pp. 22–29, 1992.

[16] W. Fan, H. Peng, S. Luo, C. Fang, and Y. Li, “SCEC: a novel single-cell classification method based on cell-pair ensemble learning,” in *Intelligent Computing Theories and Application*, pp. 433–444, Springer, 2021.

[17] W. Wang, H. Tan, M. Sun et al., “Independent component analysis based gene co-expression network inference (ICAnet) to decipher functional modules for better single-cell clustering and batch integration,” *Nucleic Acids Research*, vol. 49, no. 9, article e54, 2021.

[18] N. H. Son, “From optimal hyperplanes to optimal decision trees,” *Fundamenta Informaticae*, vol. 34, no. 1,2, pp. 145–174, 1998.

[19] X. Yao, “Evolving artificial neural networks,” *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[20] X. Zhu and Z. Ghahraman, *Learning from labeled and unlabeled data with label propagation*, Carnegie Mellon University, 2002.

[21] S. Gregory, “Finding overlapping communities in networks by label propagation,” *New Journal of Physics*, vol. 12, no. 10, 2010.

[22] Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li, “Geometry-based edge clustering for graph visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1277–1284, 2008.

[23] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR 2017)*, pp. 1–10, Vienna, 2016.

[24] Q. Li, Z. Han, and X.-M. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3538–3545, New Orleans, 2018.

[25] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pp. 7444–7452, New Orleans, 2018.

[26] L. F. Chu, N. Leng, J. Zhang et al., “Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm,” *Genome Biology*, vol. 17, no. 1, p. 173, 2016.

[27] A. P. Patel, I. Tirosh, J. J. Trombetta et al., “Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma,” *Science*, vol. 344, no. 6190, pp. 1396–1401, 2014.

[28] Y. Xin, J. Kim, H. Okamoto et al., “RNA sequencing of single human islet cells reveals type 2 diabetes genes,” *Cell Metabolism*, vol. 24, no. 4, pp. 608–615, 2016.

[29] D. Usoskin, A. Furlan, S. Islam et al., “Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing,” *Nature Neuroscience*, vol. 18, no. 1, pp. 145–153, 2015.

[30] A. Jain, K. Nandakumar, and A. Ross, “Score normalization in multimodal biometric systems,” *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, 2005.

[31] A. Huang, “Similarity measures for text document clustering,” in *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, pp. 9–56, New Zealand, 2008.

[32] M. Wang, L. Yu, D. Zheng et al., *Deep Graph Library: towards efficient and scalable deep learning on graphs*, ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[33] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, “Distributed Kalman filtering based on consensus strategies,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622–633, 2008.

[34] V. Y. Kiselev, K. Kirschner, M. T. Schaub et al., “SC3: consensus clustering of single-cell RNA-seq data,” *Nature Methods*, vol. 14, no. 5, pp. 483–486, 2017.