Transactions on Materials, Biotechnology and Life Sciences
ISSN: 2960-1932, eISSN: 2960-2319 | Volume 7
MISBP 2024

Leiden Clustering Based on Single-cell Sequencing Data of Human Bone Marrow

Jianzhang Li, Zixuan Zhao*

School of Mathematics and Physics, Xi'an Jiaotong-Liverpool University, Suzhou, China, 215123

*Corresponding author: Zixuan.Zhao2202@student.xjtlu.edu.cn

Abstract. As single-cell sequencing technology has gradually become a popular method for obtaining effective data in biology, the experimental steps corresponding to different experimental goals are also very different, which often causes researchers to make mistakes in choosing data processing methods. In response to this situation, this paper selects a part of the NeurIPS 2021 benchmark dataset of openproblem because the data processing difficulty is moderate, the data noise is low, and the dataset is bone marrow mononuclear cell data of healthy human donors, which is representative in cell processing, and conducts an in-depth compilation and analysis of the dataset, summarizing a set of more universal single-cell sequencing experimental steps. This paper first filters cells and genes by setting different thresholds of corresponding indicators to achieve the purpose of data preprocessing; in terms of data dimensionality reduction, this paper uses principal component analysis (PCA) to reduce the dimensionality of the data and mark more than 2,000 highly variable genes. Then clustering the cells, and this paper divides the cell clusters into 27 categories through the Leiden clustering method. Meanwhile, this paper also identifies and analyzes the marker genes based on the cell clusters obtained by clustering. Through the research of this article, it is found that compared with traditional sequencing methods (such as K-means, hierarchical clustering, Louvain clustering, etc.), Leiden clustering is the most locally distributed for all subsets of all communities. Salient features, and considering that Louvain clustering's lack of community connectivity is difficult to solve, make Leiden clustering more superior to Louvain clustering in processing single-cell data. At the same time, the research of this paper not only innovates in the method of single-cell sequencing, but also achieves a driving significance in the derivation of its biological functions.

Keywords: Single-cell sequencing, Principal Component Analysis, Leiden Clustering.

1. Introduction

Single-cell sequencing technology is to sequence the genetic information carried by a given single cell at the level of a given single cell. Its principle can be briefly described as amplifying the trace whole genome DNA of a single isolated cell to obtain a complete genome with high coverage, and then high-throughput sequencing through exon capture to reveal the differences in cell populations and cell evolutionary relationships [1]. With the continuous development of molecular biology, scRNA-Seq was first cited in 2009 and has gradually become a research hotspot in recent years. Compared with other high-throughput sequencing, scRNA-Seq mainly targets the RNA of a single cell, which is amplified by PCR and then sequenced by high-throughput. The main workflow includes cell dissociation, single cell isolation, library construction, on-machine sequencing and data analysis [2]. Among them, data analysis is the most important part of the entire scRNA-Seq process. The current mainstream data analysis method flow can be divided into: data preprocessing (including data comparison, quality control and data standardization steps), interpolation, batch effect correction, dimensionality reduction analysis, cell subtype identification, differential expression analysis and reverse chronological analysis steps [3].

In the step of data preprocessing, previous researchers chose the method of cell clustering because traditional clustering methods (such as Louvain clustering) had obvious defects in cell classification and other steps, which led to low experimental data quality and large deviations from expectations. Different from previous studies, this article specifically discusses the Leiden clustering method in single-cell sequencing technology due to its high efficiency in classifying node data and relative

universality in single-cell data processing. At the same time, it also compares the Leiden clustering method. Compared with Louvain clustering, there is a significant difference in data node classification - an additional Refine step is performed to improve the partitioning results and make node classification more efficient. Therefore, this paper will carry out relevant research and analysis based on the data set obtained by single-cell sequencing technology, and pioneeringly use the Leiden clustering method to conduct in-depth research on cell data step by step. This is helpful for us to understand the cell type composition in cellular organisms and their downstream analysis, and to infer the biological functions of specific tissue cells in combination with the biological functions of the corresponding cell types, so as to better understand the cell type composition of specific tissue slices; at the same time, this article can further understand the interaction between different cells and cell differentiation, and lay the foundation for the subsequent in-depth analysis of the properties of characteristic cells, which is helpful for us to understand the functions and mechanisms of research organisms and their developmental processes.

2. The principles of Leiden clustering

In many complex networks, the network structure formed after community division is usually unknown. Therefore, when detecting communities in the network, we usually use a modular approach to try to maximize the difference between the actual number of edges in the community and the expected number of edges. Here we define e_c as the actual number of edges in community c. The expected number of edges is expressed as $\frac{{k_c}^2}{2m}$, where k_c is the sum of the degrees of the nodes in community c, and m is the total number of edges in the network. The method for defining the expected number of edges is based on the so-called configuration model [4]. The modularity is given by the following formula:

$$H = \frac{1}{2m}\sum_c \left(e_c - \gamma \frac{K_c^2}{2m}\right) \tag{1}$$

Where $\gamma > 0$ is the resolution parameter [5]. The higher the resolution, the more communities there are, while the lower the resolution, the fewer communities there are. At present, optimizing modularity is an NP-hard problem [6]. Before the Leiden algorithm was proposed, the most popular method for optimizing modularity was the Louvain algorithm [7]. It is one of the fastest and best performing algorithms in comparative analysis and one of the most cited works in the community detection literature [8-9]. Although originally defined as modularity, the Louvain algorithm can also be used to optimize other quality functions. Another quality function is the Constant Potts Model (CPM) [10], which overcomes some limitations of modularity (resolution limitations). The definition of CPM is:

$$H = \sum_C \left[e_c - \gamma \binom{n_c}{2}\right] \tag{2}$$

Where n represents the number of nodes, and γ is still used as a parameter to limit the density between communities. The density of the community should be at least greater than γ, and the density between communities should be lower than this parameter. Similarly, the relationship between resolution and community is similar to the modularity parameter.

The Louvain algorithm can be simplified into two steps: local mobile nodes (modularity optimization) and network aggregation. The article has found that the Louvain algorithm has a problem in both modularity and CPM, that is, it may produce arbitrary communities with poor connectivity or even no connectivity [4].

In the Louvain algorithm, if we move a node to another node, an internally disconnected red community will appear. And the surrounding nodes are still inside the red community. In this case, it

is better to split this part of the nodes into two communities, but the Louvain algorithm only considers the movement of a single node, which is an obvious disadvantage. At the same time, due to resolution limitations, modularity may cause smaller communities to aggregate into larger communities. This means that modularity may hide smaller communities. Similarly, when CPM is used as a quality function, poorly linked communities will also be generated [4].

The Leiden algorithm starts from a given point a and regards each node as a separate community. It then moves a single node from one community to another to improve the quality function in order to find the partition situation b. Then b is improved to c, and the community to which the node belongs does not change. The network is condensed based on the improved partition c, and the community division situation (community number) of the condensed network is used as the partition b before improvement. Therefore, we can simplify the Leiden algorithm into three steps: local node movement, improvement of the partition results, and condensation of the network based on the situation. We can intuitively see that the main difference between the two algorithms is that the Leiden algorithm adds the Refine step to improve the partition results [4].

3. Results

3.1. Data description

The data used in this experiment was collected from bone marrow mononuclear cells of healthy human donors, and the samples used were measured using the 10X Multi-Omic Gene Expression and Chromatin Accessibility Kit. The raw data dimension of the single-cell sequencing dataset used in this article is 17041×23427, which means it contains 17041 measured genes and 23427 cells. We will now use a basic pre-processing and clustering workflow to analysis the above genes. The detailed steps for data processing in this experiment are shown in the Figure 1 following mind map:

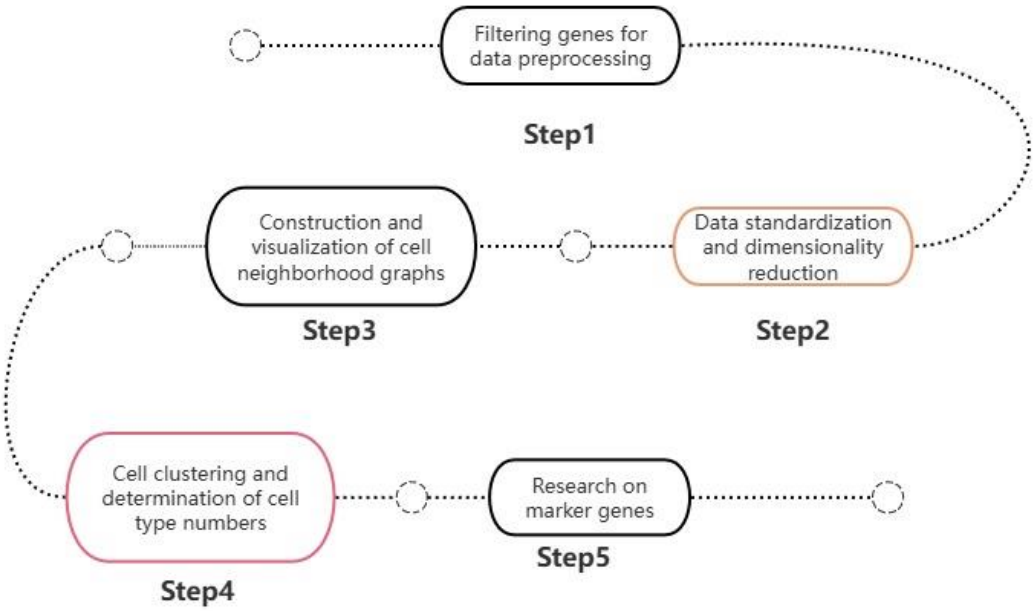

Figure 1: Data processing steps mind map.

3.2. Data preprocessing

The data set used in this paper is a high-dimensional data which used in this article was collected from bone marrow mononuclear cells of healthy human donors and was part of openproblem' s NeurIPS 2021 benchmarking dataset., and the number of cells and genes is huge. In fact, the gene expression in a considerable number of cells in this data set is close to zero, which will make our

gene-cell matrix a sparse matrix (there are a lot of zeros in the matrix composed of data), which will affect the sparsity of the data. If we directly model based on the sparse matrix, the quality of subsequent data analysis will be reduced, because we cannot extract effective data information from a large number of zero samples. Based on this, we need to preprocess the data before formal analysis, among which the filtering of genes and cells is more important: when filtering cells, we detect that the genes in some cells are not fully detected, so the extracted gene information is also less. We call these cells low-quality cells, and we will filter them out; when filtering genes, some genes have low expression levels, and we also filter these genes. Therefore, we get the following indicators in quality control (represented in the form of violin plots):

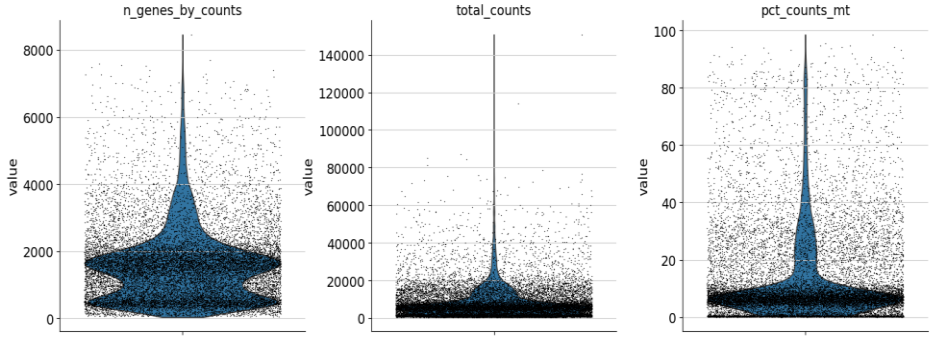

Figure 2: Violin plot of indicators in quality control.

Table 1: Definition of different indicators

Count the number of genes expressed in the matrix	Genes after initial filtering
Total number of genes per cell	The number of genes expressed per cell
Percentage of counts for mitochondrial genes	Indicates the degree of cell aging

As shown in Table 1, we explained three indicators, which can be considered as references when filtering genes and cells. As shown in Figure 2, we can intuitively see the base distribution of more than 8,000 genes after filtering. As shown in Figure 3, we found that the number of genes in most cells is between 10,000 and 20,000. As shown in Figure 6, we can detect aging or apoptotic cells and filter them.

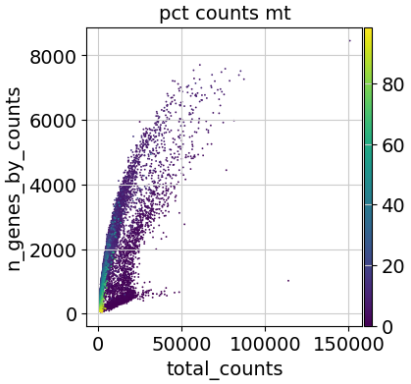

Figure 3: Colored scatter plot.

Then, we use scatter plots to visualize the above quality assessment indicators. As shown in Figure 3, the x-axis is the total number of genes in each cell, the y-axis represents the number of genes expressed in the count matrix, and the color bar on the right side represents the percentage of counts in mitochondrial genes. Based on the above figure, we can draw the following conclusions: when the percentage of mitochondrial genes is high, the corresponding cells will also have a smaller total count and less gene expression. Considering that cells with too large a percentage of mitochondrial counts

have no reference value due to their aging degree and that cells with too large a total count are mostly polyploid, we set the threshold: cells expressing more than 100 genes and more than 3 genes detected in cells. Cells with too many mitochondrial genes and too many total counts were further removed.

3.3. Standardization

The gene cell expression matrix is a sparse matrix. The expression of many detected genes is close to zero, while the expression of some genes is very high, which leads to huge differences in the expression values of different genes. If the data is directly modeled without standardization, it will cause a huge dimension in the data, which will make it difficult to capture genes with low expression. Standardizing the data can obviously solve the above problem. In this paper, we use the log plus one (log1p) transformation, that is:

$$x' = \log(x + 1) \tag{3}$$

x represents the expression level of the gene, and x' is the normalized expression value. This effectively reduces the data difference and helps us compare and convert the data.

3.4. Dimensionality reduction of data

Next, we reduced the dimensionality of the dataset to include only the most informative genes, which can also be called feature screening. In biology, we call genes with large differences in expression in a pair of different cells highly variable genes. Screening and marking highly variable genes help us determine the cell type they correspond to. Similarly, we also set a corresponding threshold when screening highly variable genes: genes with expression levels greater than 2000 are marked as highly variable genes.

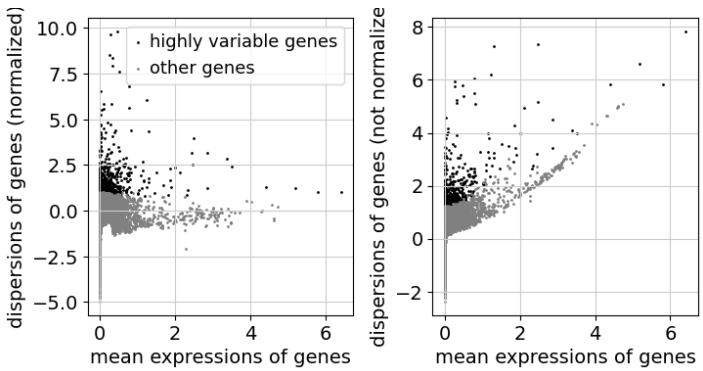

Figure 4: Comparison of highly variable genes and non-highly variable genes.

As shown in Figure 4, we can intuitively see that in the scatter plot, black is a highly variable gene and gray is a non-highly variable gene, the x-axis is the average gene expression, and the y-axis is the dispersion of the gene. We can find that the dispersion of highly variable genes is higher than that of non-highly variable genes, which means that the expression of highly variable genes varies greatly, which is consistent with our definition in the previous article. The left subplot is a graph of the standardized data, and the right subplot is a graph before standardization. By comparing the left and right graphs, we can see that before standardization, the dispersion of genes increases when the gene expression increases, but there is no such a trend after data standardization, which proves that the stability of the data is guaranteed after we standardize the data.

We used principal component analysis (PCA) to reduce the dimensionality of the data. For PCA, it projects the data from a high-dimensional space to a low-dimensional space through a projection method. In the projection process, there are two methods: "based on variance, projecting from the direction of minimum variance" and "calculating eigenvalues to find the projection direction". After running PCA, we can get multiple principal components. The principal component information

contained in it decreases step by step from the first principal component. Therefore, we usually retain the first n principal components to represent the original data information, and discard the remaining principal components. This is also our method of data denoising. Similarly, we get the principal component diagram:

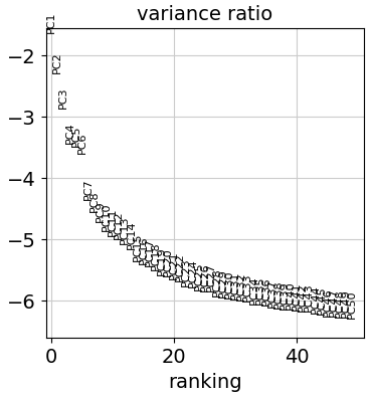

Figure 5: Principal component scree plot.

As shown in Figure 5, the figure shows the variance contribution rate of the first fifty principal components, and then we draw the principal component diagram:

3.5. Construction and visualization of nearest neighbor graphs

We used the PCA representation of the data matrix to compute the neighborhood graph of the cells, and for visualization we used the UMAP method.

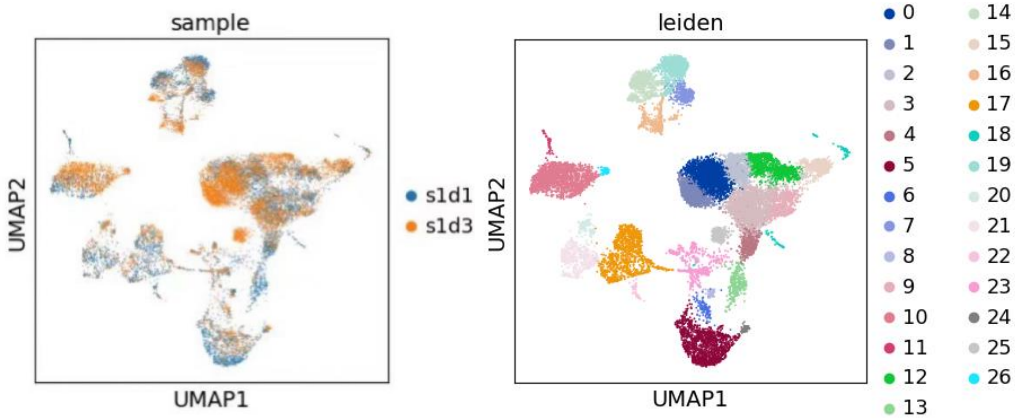

Figure 6(a): Visualizing the neighborhood graph. Figure 6(b): Leiden clustering results.

As shown in Figure 6(a), we once again find that there is only a small batch effect between samples from two different batches, and there is no need for cross-version integration or batch correction.

3.6. Clustering

In this experiment, we used the Leiden clustering method because of its good community detection function of optimizing modularity and its good performance in single-cell sequencing.

As shown in Figure 6(b), after Leiden clustering, we divide the cell clusters into 27 categories. We can find that the discrimination of the clustering results is relatively high.

3.7. Determination of cell type number

Next, we proceed to annotate the above cell sets to annotate them as known cell types. Typically, this is done using genes that are only expressed by a given cell type, in other words, these genes are marker genes for the cell type and are therefore used to distinguish the heterogeneous cell groups in

our data. Typically, cell type annotation uses those specific marker genes and then groups the cells into clusters. Here, we will still use the Leiden clustering algorithm, which will extract cell communities from our nearest neighbor graph. When defining the number of clusters, we use the '*resolution'* parameter to control its number.

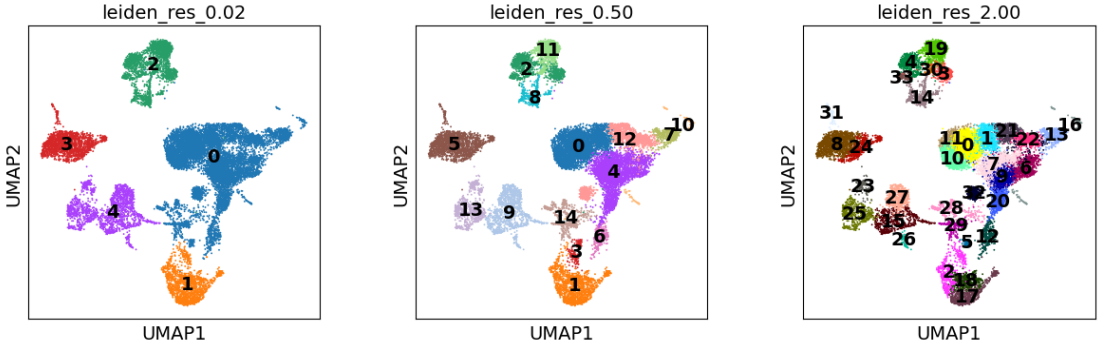

Figure 7: Cell clusters at different thresholds.

As shown in Figure 7, we set different numbers of clusters to explore the most appropriate number of cell clusters for this experiment. This study is only about the different accuracy of cell type division, and there is no right or wrong. However, too high or too low clustering will result in a certain degree of classification error. Here we can see that at the highest resolution our data is over-clustered, while at the lowest resolution, our data may group cells belonging to different cell identities.

3.8. Research on marker genes

By comparing the clustering results when we used *resolution=0.50*, we will further study the expression of specific marker genes in different cells.

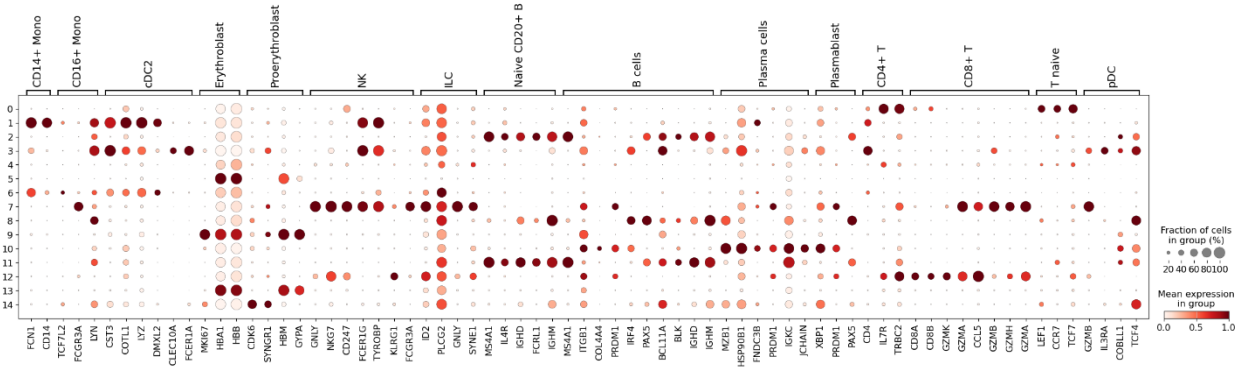

Figure 8: Comparison of the relationship between marker genes and expression in different cells.

As shown in Figure 8, the different colors in the figure above represent the average expression level in each cell cluster, and the size of the circle represents the proportion of cells containing the gene in the cluster. The gene name is at the bottom and the cell cluster name is at the top. Taking the first column as an example, we can find that FCN1 is highly expressed on CD14+Mono, so we call it a marker gene for the corresponding cells, which is an important basis for us to distinguish different cells.

From the comparison between prediction data and actual data, the BP neural network has better prediction performance and relatively small error, which can meet the demand completely, and has fast prediction speed and convenient operation.

4. Conclusions

Based on a given single-cell dataset, this paper first preprocessed the data. We set corresponding thresholds based on indicators such as the count percentage of mitochondrial genes to filter out

substandard cells, and then reduced the dimensionality of the data by screening highly variable genes and using principal component analysis. We then used the Leiden clustering method to divide the cell clusters into 27 categories. Finally, we set the resolution threshold to divide the cell clusters into 15 cell clusters again and found their corresponding marker genes.

This article describes in detail the advantages and reasons of the Leiden clustering method compared to traditional clustering methods in cell community division, and demonstrates the complete single-cell data analysis process, which will serve as a reference for future researchers in the selection of clustering methods and data analysis steps. In this experiment, we analyzed the cell type composition of human bone marrow tissue, which will help the subsequent analysis of the properties of the corresponding cell types; at the same time, we identified and mined the corresponding marker genes in the cell types obtained by clustering, and we can subsequently infer their biological functions from the perspective of marker genes; based on marker genes, we can perform cell annotation, and then we can explore the differentiation process and cell communication of bone marrow tissue in the future. These experimental data on human bone marrow cells also provide a data basis for scientific research related to human bone marrow cells (such as human bone marrow cell morphology and subtype classification).

References

[1] Wang Quan, Wang Zhu, Zhang Zhen, Li Chen, Zhang Mengmeng, Ye Yingjiang, Wang Shan, Jiang Kewei. Overview of the Technology of Single Cell Sequencing[J]. CHINESE JOURNAL OF MEDICINAL GUIDE, 2020, 22(7): 433-439.

[2] See P, Lum J, Chen J, et al. A Single-cell sequencing guide for immunologists[J]. Frontiers in Immunology, 2018, 9 ： 2425.

[3] Zhang Miao, Sun Xiang-rui, Xu Chun-ming. Research Progress of Approaches in Single Cell RNA Sequencing Data Analysis[J]. Biotechnology Bulletin, 2021, 37(1): 52-59.

[4] Traag, Vincent A., Ludo Waltman, and Nees Jan Van Eck. From Louvain to Leiden: guaranteeing well-connected communities[J]. Scientific reports, 2019, 9(1): 1-12.

[5] Newman, Mark EJ, and Michelle Girvan. Finding and evaluating community structure in networks[J]. Physical review E, 2004, 69(2): 026113.

[6] Li, Wenjun, et al. Parameterized algorithms of fundamental NP-hard problems: a survey[J]. Human-centric Computing and Information Sciences, 2020(10): 1-24.

[7] Zhang, Jicun, et al. An improved Louvain algorithm for community detection[J]. Mathematical Problems in Engineering, 2021,2021(1): 1485592.

[8] Christensen, Alexander P., et al. Comparing community detection algorithms in psychological data: A Monte Carlo simulation[M]. 2020.

[9] Smith, Natalie R., et al. A guide for choosing community detection algorithms in social network studies: The question alignment approach[J]. American journal of preventive medicine, 2020, 59(4): 597-605.

[10] Rozikov U A. Gibbs measures in biology and physics: The Potts model[M]. 2022.