


# scMUG: deep clustering analysis of single-cell RNA-seq data on multiple gene functional modules

De-Min Liang and Pu-Feng Du 

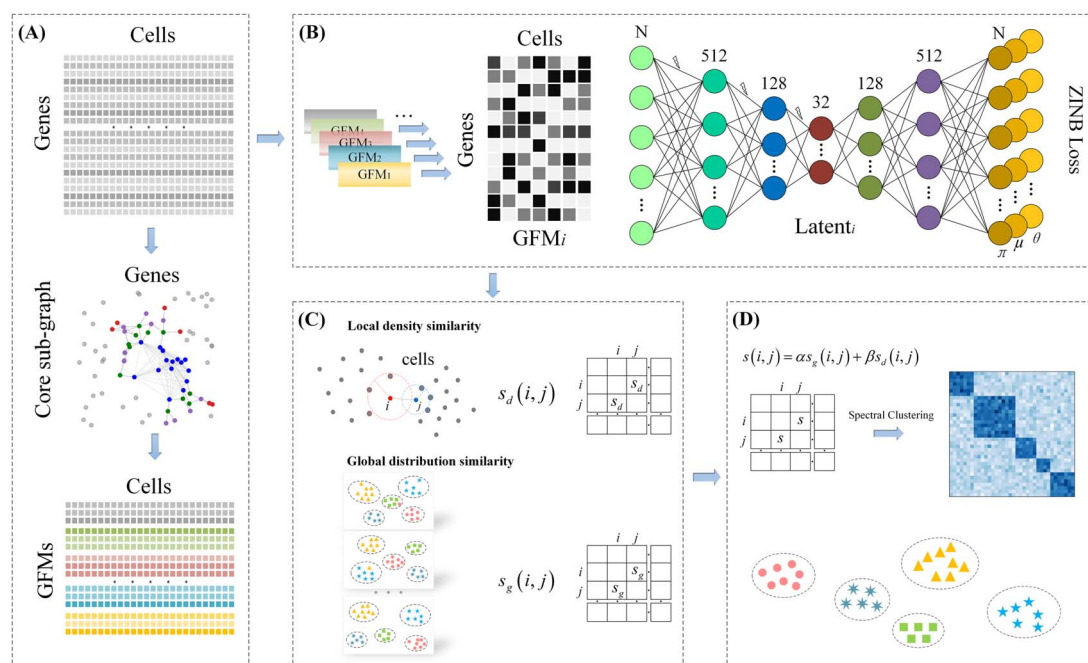
College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

\*Corresponding author. College of Intelligence and Computing, Tianjin University, Tianjin 300350, China. E-mail: pdu@tju.edu.cn

## Abstract

Single-cell RNA sequencing (scRNA-seq) has revolutionized our understanding of cellular heterogeneity by providing gene expression data at the single-cell level. Unlike bulk RNA-seq, scRNA-seq allows identification of different cell types within a given tissue, leading to a more nuanced comprehension of cell functions. However, the analysis of scRNA-seq data presents challenges due to its sparsity and high dimensionality. Since bioinformatics plays an important role in the analysis of big data and its utility for the welfare of living beings, it has been widely applied in analyzing scRNA-seq data. To address these challenges, we introduce the scMUG computational pipeline, which incorporates gene functional module information to enhance scRNA-seq clustering analysis. The pipeline includes data preprocessing, cell representation generation, cell-cell similarity matrix construction, and clustering analysis. The scMUG pipeline also introduces a novel similarity measure that combines local density and global distribution in the latent cell representation space. As far as we can tell, this is the first attempt to integrate gene functional associations into scRNA-seq clustering analysis. We curated nine human scRNA-seq datasets to evaluate our scMUG pipeline. With the help of gene functional information and the novel similarity measure, the clustering results from scMUG pipeline present deep insights into functional relationships between gene expression patterns and cellular heterogeneity. In addition, our scMUG pipeline also presents comparable or better clustering performances than other state-of-the-art methods. All source codes of scMUG have been deposited in a GitHub repository with instructions for reproducing all results (<https://github.com/degiminnal/scMUG>).

## Graphical Abstract



**Keywords:** scRNA-seq; autoencoder; clustering analysis; gene functional modules

Received: November 8, 2024. Revised: February 11, 2025. Accepted: March 9, 2025

© The Author(s) 2025. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact [journals.permissions@oup.com](mailto:journals.permissions@oup.com)

## Introduction

Single-cell RNA sequencing (scRNA-seq) has been applied widely in revealing hidden gene expression patterns at single-cell level [1–3]. Traditionally, bulk RNA-seq extracts gene expression profiles at sample level, which are essentially average gene expression counts of many cells [4]. Gene expressions in different cells of a bulk sample are neither synchronous temporally nor consistent spatially [5]. Accurate analysis of cellular heterogeneities is difficult with bulk RNA-seq data [6]. scRNA-seq is a revolution technology that enables deep investigation on gene expressions at single-cell level [7, 8]. Different types of cells can be identified in a given tissue, allowing us to have a deeper and a more precise understanding of their biological roles [9, 10].

Annotating cell types based on scRNA-seq is a fundamental task in biological and medical studies [11]. With the scRNA-seq data, it is still difficult to annotate or discover cell types using traditional computational methods [4]. scRNA-seq profiles are always very sparse, usually containing over 60% zero values [12, 13]. These zero values may only be missing values, which are dropped by the sequencing process, but not indicating a completely silenced gene. This is due to the low capture rate in the scRNA-seq experiments. Moreover, scRNA-seq profiles are usually high dimensional vector data with massive noises. These facts construct a major challenge for analyzing scRNA-seq data using computational methods.

Bioinformatics is an interdisciplinary field that plays an important role in the study of protein targeting sequencing, drug development, and the identification of gene functions, regulatory elements, and functional regions [14–16]. In this emerging field, we utilize computational tools, statistical models, and algorithms to analyze and interpret large datasets in biological and health sciences [17–20]. Unsupervised machine learning algorithms, mainly clustering algorithms, have been introduced in annotating or discovering different cell types from scRNA-seq data. Traditional clustering methods do not work well on scRNA-seq data. Efforts have been made by introducing clustering algorithm enhancements in many ways [21–24]. For example, the SC3 constructs a consensus matrix by performing multiple rounds of clustering on various data projections, followed by hierarchical clustering, to enhance the robustness of the clustering [24]. Since raw scRNA-seq gene expression profiles are always in a high-dimensional space, autoencoders are widely applied in generating cell representations in a lower-dimensional space. For example, deep count autoencoder (DCA) employs autoencoders in data preprocessing, using deep learning methods to reduce noise and dimensionality in single-cell gene expression profile data [25]. For another example, scBGEDA further utilizes k-means for pre-clustering in the latent space, constructs a bipartite graph based on these clusters, and employs bipartite graph ensemble clustering for single-cell clustering analysis [26]. The scziDesk model adopts a weighted soft k-means clustering algorithm in the latent space [27]. The scMAE method introduced masked autoencoders that are specifically designed for scRNA-seq data to extract more information of gene–gene associations [28]. Graph neural networks were introduced to enhance cell representations recently. scGNN proposed a cell–cell network construction, which was used as a basis to extract graph-based cell representations [29]. Graph-sc constructs a gene-cell association network. Cell representations were obtained from this bipartite graph with graph autoencoders [30]. Existing methods provided deep insights in applying advanced algorithms to produce cell representations, and to produce better and better clustering results. Since

scRNA-seq data is intrinsically with a lot of noise, recent algorithms tend to apply deep machine learning models to denoise scRNA-seq data. Although gene–gene correlations had been considered in existing methods [31], this information is extracted solely from the scRNA-seq expression profiles. It represents essentially the same information as the expression matrix.

In biology, genes are correlated not only superficially by their expression values, but also intrinsically by their functional implications. Functional correlations between genes have been explored for a very long time. It is a fundamental knowledge that genes perform their functions in groups. Therefore, given a cell type, we believe that a set of genes with functional relationships must express with a correlated pattern. However, the functional relationships between genes are still missing in scRNA-seq clustering analysis. Databases for gene functional modules (GFMs) have been established [32–36]. We introduce this external information of functional relationships between genes to filter scRNA-seq data before clustering analysis.

To this end, we propose the scMUG pipeline, which performs scRNA-seq clustering analysis with the consideration of gene functional correlations. The scMUG pipeline has the ability to establish relationships between cell types and functional gene modules. It also provides hints to find potential new cell types. Besides, to further enhance the power of clustering algorithm, we introduced a novel similarity measure between cells, which combines a local density measure and a global distribution measure in the latent cell representation space. As a result, scMUG provides comparable or better clustering results than other state-of-the-art methods. As far as we can tell, this is the first attempt to incorporate external knowledge of gene functional relationships in scRNA-seq clustering analysis.

## Materials and methods

### Dataset curation

We curated nine human scRNA-seq datasets [37–44] from Hemberg's collection (<https://hemberg-lab.github.io/scRNA.seq.datasets>). These datasets cover a wide range of biological diversity (brain, pancreas, liver, embryonic cells, and various tumor microenvironments), technical heterogeneity (Smart-seq2, 10X Genomics, and inDrop) and complexity of data (ranging from 90 to over 8000 cells and 6–16 cell types), ensuring a comprehensive evaluation of our scMUG pipeline. The scMUG pipeline incorporated GFM information in human genome to assist the clustering analysis. Since there is no clear source of mouse GFMs, we only use human scRNA-seq dataset in this study. We listed the number of cells, number of genes and the platforms for obtaining the data in Table 1. We list the details of these nine datasets in Supplementary Table S1, showing the source organ, the platform, the number of annotated cell types, the number of cells and the zero percentage. The Manno dataset is used as a benchmarking dataset for the first time in developing scRNA-seq clustering analysis pipeline. Since the number of cells of different cell types are highly imbalanced in this dataset, we choose to use a subset containing seven largest cell types (eProg1b, eProg1a, hRgl2a, hPeric, eProg2a, eSCa, and eNb1).

### Overview of scMUG

The whole pipeline of scMUG can be separated into four steps: data preprocessing, generating cell representations, constructing cell–cell similarity matrices, and cell clustering analysis (Fig. 1).

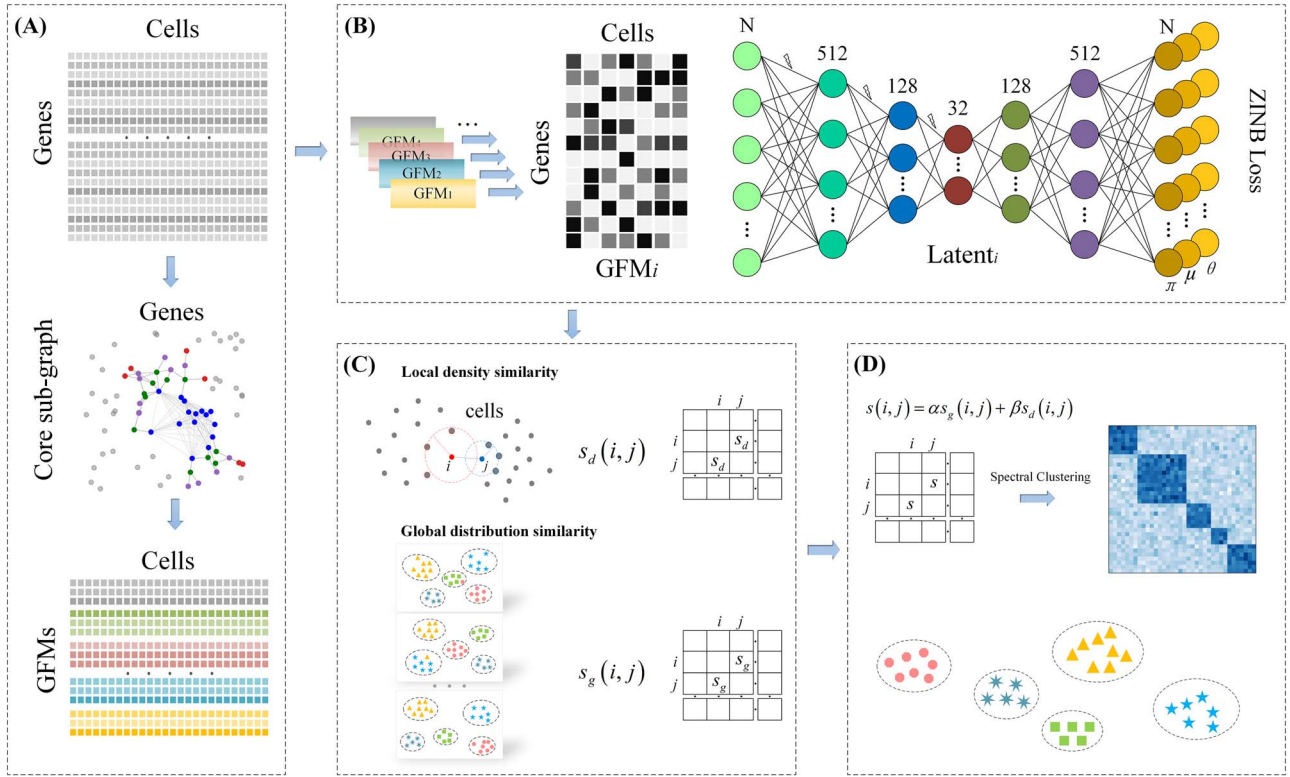


Figure 1. Flowchart of scMUG. (A) Raw scRNA-seq data were preprocessed with statistical filters. Gene co-expression network was established using filtered expression profiles. Core sub-graphs were extracted. GFMs were identified. (B) Cell representations are established by ZINB-loss based autoencoders in every GFM, respectively. (C) Constructing cell-cell global distribution similarity matrix and local density similarity matrix. (D) Cell clustering analysis by spectral clustering with the combination of global and local similarity.

Table 1. Summary of datasets utilized in scMUG analysis.

Dataset	Number of Cells	Number of Genes	Platform
Camp1	777	19 020	SMARTer
Camp2	734	18 927	SMARTer
Darmanis	466	22 088	SMARTer
Baron	8569	20 125	inDrop
Li	561	55 186	SMARTer
Yan	90	20 214	Tang
Muraro	2122	19 046	CEL-Seq2
Lake	3042	25 051	snRNA-seq
Manno	1281	20 560	STRT-Seq UMI

GFM information was incorporated in generating cell representations. Different GFM generates different cell representations. We introduced two cell-cell similarity matrices, the global distribution similarity and the local density similarity, to enhance the spectral clustering algorithm.

## Data preprocessing

Let  $X \in \mathbb{R}^{n \times d}$  be the scRNA-seq gene expression matrix containing  $n$  cells, where each cell has  $d$  gene expression values. We first excluded those genes, which have zero expression values across all cells. We note the number of remaining genes as  $d_z$ . We normalized the expression values in each cell as follows:

$$y_{ij} = \ln \left( 10^5 x_{ij} / \sum_{j=1}^{d_z} x_{ij} + 1 \right), \quad (1)$$

where  $x_{ij}$  is the raw expression value of the  $j$ -th gene in the  $i$ -th cell, and  $y_{ij}$  the normalized expression value of the  $j$ -th gene in the  $i$ -th cell. We note the normalized gene expression matrix as  $Y = \{y_{ij}\}_{n \times d_z}$ . The scanpy package [45] was used to choose top  $m$  highly variable genes from  $Y$  with all default parameter values.

We transform highly variable gene expression values with Z-transformation across all cells, as follows:

$$z_{ij} = \frac{y_{ij} - \mu_j}{\sigma_j}, \quad (2)$$

where

$$\mu_j = \frac{1}{n} \sum_{i=1}^n y_{ij}, \quad (3)$$

$$\sigma_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_{ij} - \mu_j)^2}, \quad (4)$$

and  $z_{ij}$  the Z-transformed normalized gene expression value of the  $j$ -th gene in the  $i$ -th cell. We note the gene expression matrix after all above preprocessing steps as  $Z$ . We have  $Z \in \mathbb{R}^{n \times m}$ , and  $Z = \{z_{ij}\}_{n \times m}$ .

## Gene expression correlation network construction

To generate cell representations, we first construct the gene expression correlation network from matrix  $Z$ . Pearson correlation coefficients were calculated between every two genes in  $Z$ . We use  $r_{u,v}$  ( $u, v = 1, 2, \dots, m$ ) to note the correlation between the  $u$ -th gene and the  $v$ -th gene. To correct the distribution of Pearson

correlation coefficients, Fisher's transformation was applied, as follows:

$$\rho_{u,v} = \frac{1}{2} \ln \left( \frac{1 + r_{u,v}}{1 - r_{u,v}} \right), \quad (5)$$

where  $\rho_{u,v}$  is the corrected Pearson correlation coefficient. When  $r_{u,v}$  is  $-1$  or  $1$ , a regularization factor  $\varepsilon = 10^{-6}$  was used to bound the value to  $-1 + \varepsilon$  or  $1 - \varepsilon$ . Given a cutoff value  $c$ , if  $|\rho_{u,v}| \geq c$ , we connect the  $u$ th and the  $v$ th gene to construct the gene expression correlation network in a given dataset. Intuitively, all genes will be connected in one or more isolated sub-graphs. We term the largest sub-graph as the core sub-graph. Let  $n_0$  be the size of the core sub-graph, we have  $n_0 = f(c)$ . We scanned the value  $c$  from  $1$  to  $0$  with a step of  $0.01$ . The value of  $n_0$  increases as  $c$  decreases. We choose the maximal  $c$  when  $n_0 > n_d$ , where  $n_d$  is the desired minimal size of the core sub-graph. With a given  $n_d$ , we find the value of  $c$  and a set of genes in the core sub-graph.

### Gene functional module expansion

We used the HumanBase [32] GFM identification service to find primary GFMs from the core sub-graph. Due to the restriction of HumanBase, uploading too many genes to the HumanBase online service at once can cause it to fail. Therefore, genes in the core sub-graph are partitioned into several batches before uploading to HumanBase online service. Each batch contains 2000 genes maximal. To capture GFMs that spread in more than one batch, GFMs in each batch are collected and uploaded, respectively, again with other batches to find complete GFMs in the core sub-graph. Significant GFMs with similar functions are joined. We use  $q$ -value cutoff  $10^{-4}$  to find significant GFMs.

Since the number of genes in one significant GFM may be too small to guide scRNA-seq clustering analysis, we developed a GFM expansion algorithm to expand the GFM (Supplementary Algorithm 1). Let  $n_g$  be the size of a GFM,  $n_s$  the minimal GFM size requirement,  $c_g$  a correlation cutoff value. Suppose that we have the  $u$ th gene in a GFM and the  $v$ th gene not in this GFM. If  $|\rho_{u,v}| > c_g$ , we add the  $v$ th gene to the GFM. This procedure was executed iteratively with adjustment to  $c_g$  until  $n_g \geq n_s$  is satisfied. We note the number of significant GFMs as  $n_c$ .

### Zero-inflated negative binomial distribution-based autoencoder

In each expanded GFM, we applied zero-inflated negative binomial (ZINB) distribution-based autoencoder to generate gene expression representation for each cell. The autoencoder is designed as in Fig. 1, containing seven layers. The latent layer with 32 neurons, which was further reduced to 2-D dots by UMAP [46, 47], was used as the cell representation. The loss function of this autoencoder is as follows:

$$l_{\text{ZINB}} = - \sum_{i=1}^{n_g} \ln \text{ZINB}(x_i | \pi_i, \mu_i, \theta_i), \quad (6)$$

where  $l_{\text{ZINB}}$  is the ZINB distribution-based reconstruction loss,  $n_g$  the number of genes in a GFM,  $x_i$  the input expression value of the  $i$ -th gene,

$$\mu_i = \min(\max(\exp(x'_i), 10^{-5}), 10^6), \quad (7)$$

$$\theta_i = \min(\max(\ln(1 + \exp(x'_i)), 10^{-4}), 10^4), \quad (8)$$

$$\pi_i = \exp(x'_i) / (1 + \exp(x'_i)), \quad (9)$$

$$\text{ZINB}(x | \pi, \mu, \theta) = \pi \delta_0(x) + (1 - \pi) \text{NB}(x | \mu, \theta), \quad (10)$$

$$\text{NB}(x | \mu, \theta) = \frac{\Gamma(x + \theta)}{\Gamma(x + 1) \Gamma(\theta)} \left( \frac{\theta}{\theta + \mu} \right)^\theta \left( \frac{\mu}{\theta + \mu} \right)^x, \quad (11)$$

$x'_i$  the reconstructed expression value of the  $i$ -th gene, and  $\Gamma(\cdot)$  the gamma function.

### Global distribution similarity matrix

Cell representations are used to generate global distribution similarity and local density similarity matrices for clustering analysis. Global distribution similarity was calculated by repeating  $n_k$  times  $k$ -means clustering with random initializations on cell representations in each significant GFM of every dataset.

With the  $c$ -th significant GFM, in the  $t$ -th time  $k$ -means clustering, we calculated a global score  $s_{t,c}(i)$  for the  $i$ -th cell, as follows:

$$s_{t,c}(i) = \sqrt{\frac{|d_{0,t,c}(i) - d_{1,t,c}(i)|}{d_{0,t,c}(i) + d_{1,t,c}(i)}}, \quad (12)$$

where  $d_{0,t,c}(i)$  is the distance between the  $i$ -th cell and the cluster center it belongs to in the  $t$ -th time  $k$ -means clustering, and  $d_{1,t,c}(i)$  the distance between the  $i$ -th cell and the closest cluster center it does not belong to in the  $t$ -th time  $k$ -means clustering. We have:

$$s_0(i, j) = \frac{\sum_{c=1}^{n_c} \sum_{t=1}^{n_k} s_{t,c}(i) s_{t,c}(j) I_{i=j}(t, c)}{n_k n_c}, \quad (13)$$

where  $s_0(i, j)$  is the unnormalized global distribution similarity between the  $i$ -th and the  $j$ th cell,  $n_c$  is the number of GFMs as we have mentioned,

$$I_{i=j}(t, c) = \begin{cases} 1 & l(i, t, c) = l(j, t, c) \\ 0 & \text{Otherwise} \end{cases}, \quad (14)$$

and  $l(i, t, c)$  the clustering label for the  $i$ -th cell in the  $t$ -th time  $k$ -means clustering with regard to the  $c$ -th significant GFM. We further normalize the  $s_0(i, j)$  as follows:

$$s_g(i, j) = \frac{1}{m_g} s_0(i, j), \quad (15)$$

where

$$m_g = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n s_0(i, j). \quad (16)$$

The optimal  $k$  value was obtained on each dataset respectively by trials.

### Local density similarity matrix

Let  $N_k(c, i)$  be the set of  $k$  nearest neighbors of the  $i$ -th cell with regard to the  $c$ -th GFM. We calculate the average distance of the  $i$ -th cell to every cell in  $N_k(c, i)$  using genes in the  $c$ -th GFM, as follows:

$$d_{c,i} = \frac{1}{k} \sum_{t \in N_k(c, i)} d_{c,i,t}, \quad (17)$$

where  $d_{c,i,t}$  is the Euclidian distance between the  $i$ -th cell and the  $t$ -th cell, which belongs to the set  $N_k(c, i)$ . By default, we set  $k = 3$ . For the  $i$ -th and the  $j$ -th cells, a local density score is calculated as follows:

$$s_d(c, i, j) = \sqrt{\min\left(\frac{d_{c,i,j}}{d_{c,i}}, \frac{d_{c,i}}{d_{c,j}}\right) \min\left(\frac{d_{c,i,j}}{d_{c,j}}, \frac{d_{c,j}}{d_{c,i}}\right)}. \quad (18)$$

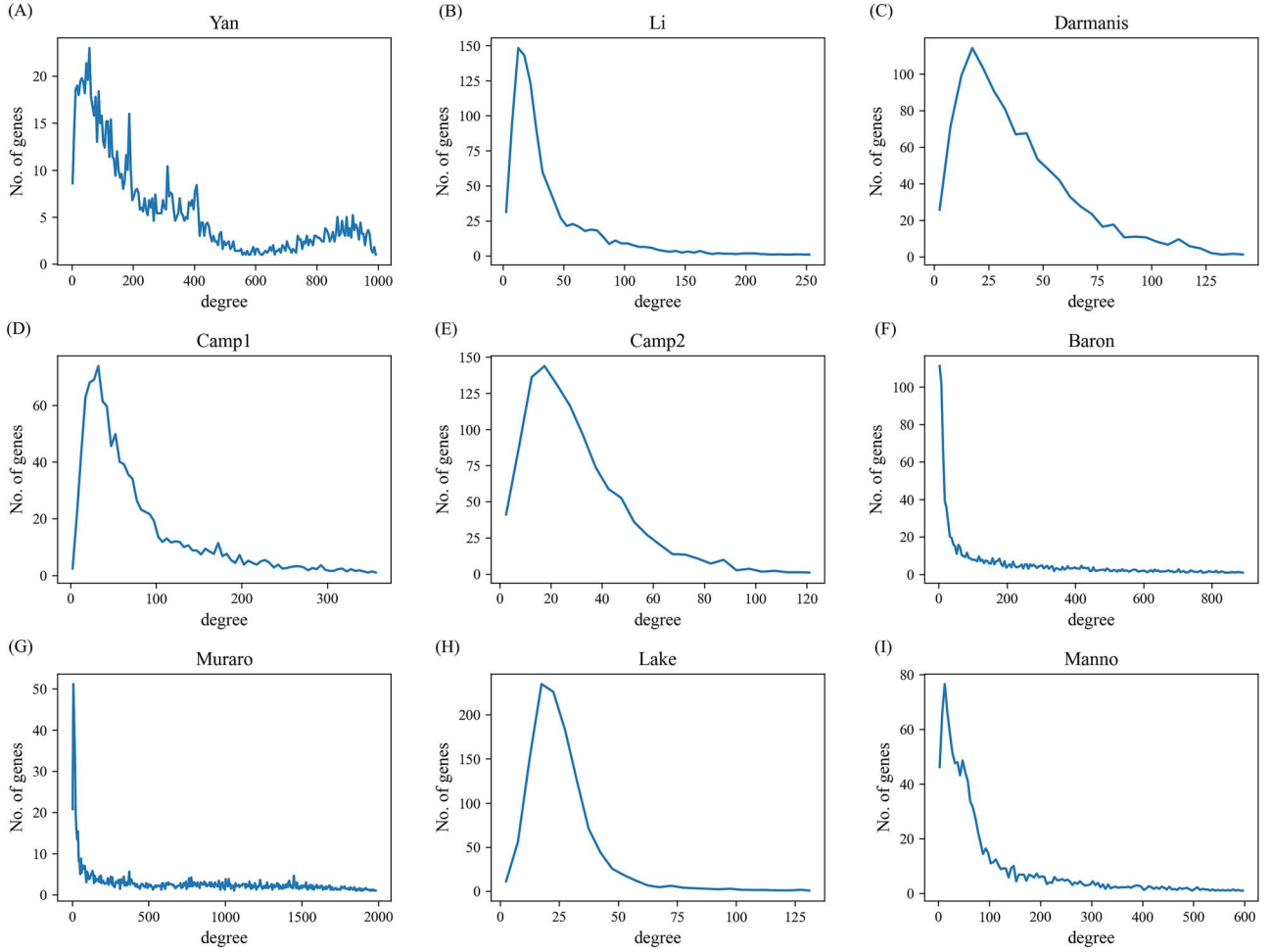


Figure 2. (A)–(I) Degree distribution of the core sub-graph in each dataset. The horizontal axis is the degree. The vertical axis is the number of nodes with the corresponding degree.

We also calculate the local dispersion measure as follows:

$$\varphi(c, i) = \sqrt{\ln \left( \frac{\text{var}_{t \in N_k(i)} d_{c,i,t} + e}{\text{var}_{t \in N_k(i)} d_{c,i,t}} \right)}, \quad (19)$$

where “var” is to calculate the variance of  $k$  distances, and  $e$  a regularization factor to ensure  $\varphi(c, i) \geq 1$ .

The local density similarity is generated as follows:

$$s_d(i, j) = \max_c \left( \frac{s_d(c, i, j)}{\varphi(c, i) \varphi(c, j)} \right). \quad (20)$$

## Clustering analysis

We combined the global distribution similarity and the local density similarity together, as follows:

$$s(i, j) = \alpha s_g(i, j) + \beta s_d(i, j), \quad (21)$$

where  $\alpha$ , and  $\beta$  are parameters for generating a linear combination. A spectra clustering was performed on  $S = \{s(i, j)\}_{n \times n}$  to output the final clustering results. The ARI (Adjusted Rand Index), NMI (Normalized Mutual Information), and ACC (Clustering Accuracy) are used as performance measures [48].

## Parameter settings

The parameters in our study are set as follows:  $m = 8000$ ,  $n_d = 5000$ ,  $n_k = 20$ ,  $n_s = 3000$ , ZINB distribution-based autoencoder contain seven fully connected layers, the dimensions of every layer are  $n_g$ , 512, 128, 32, 128, 512, and  $n_g$ , respectively. The learning rate is  $10^{-4}$ . The epoch is 50. We use the Adam optimizer. The first three layers incorporates a Gaussian noise with standard deviation 0.15. The  $\alpha$  and  $\beta$  are chosen from nine combinations:  $\{(0, 1), (10^{-3}, 1), (10^{-2}, 1), (0.1, 1), (1, 1), (1, 0.1), (1, 10^{-2}), (1, 10^{-3}), (1, 0)\}$ . The best performed combinations on every dataset are listed as [Supplementary Table S2](#).

## Results and discussions

### Statistical attributes of the core sub-graph and GFM

We established gene–gene co-expression network on each dataset. The core sub-graph was extracted on each dataset, respectively. We explore the properties of the core sub-graph of each dataset. The degree distribution of the core sub-graph was illustrated in [Fig. 2](#) for each dataset, respectively. It is interesting to see that these distributions are not exponential distributions. Exponential distributions are constantly reported in many literatures studying gene co-expression networks [49–51]. Considering that we restricted the size of core sub-graphs, the cutoff value  $c$  is smaller

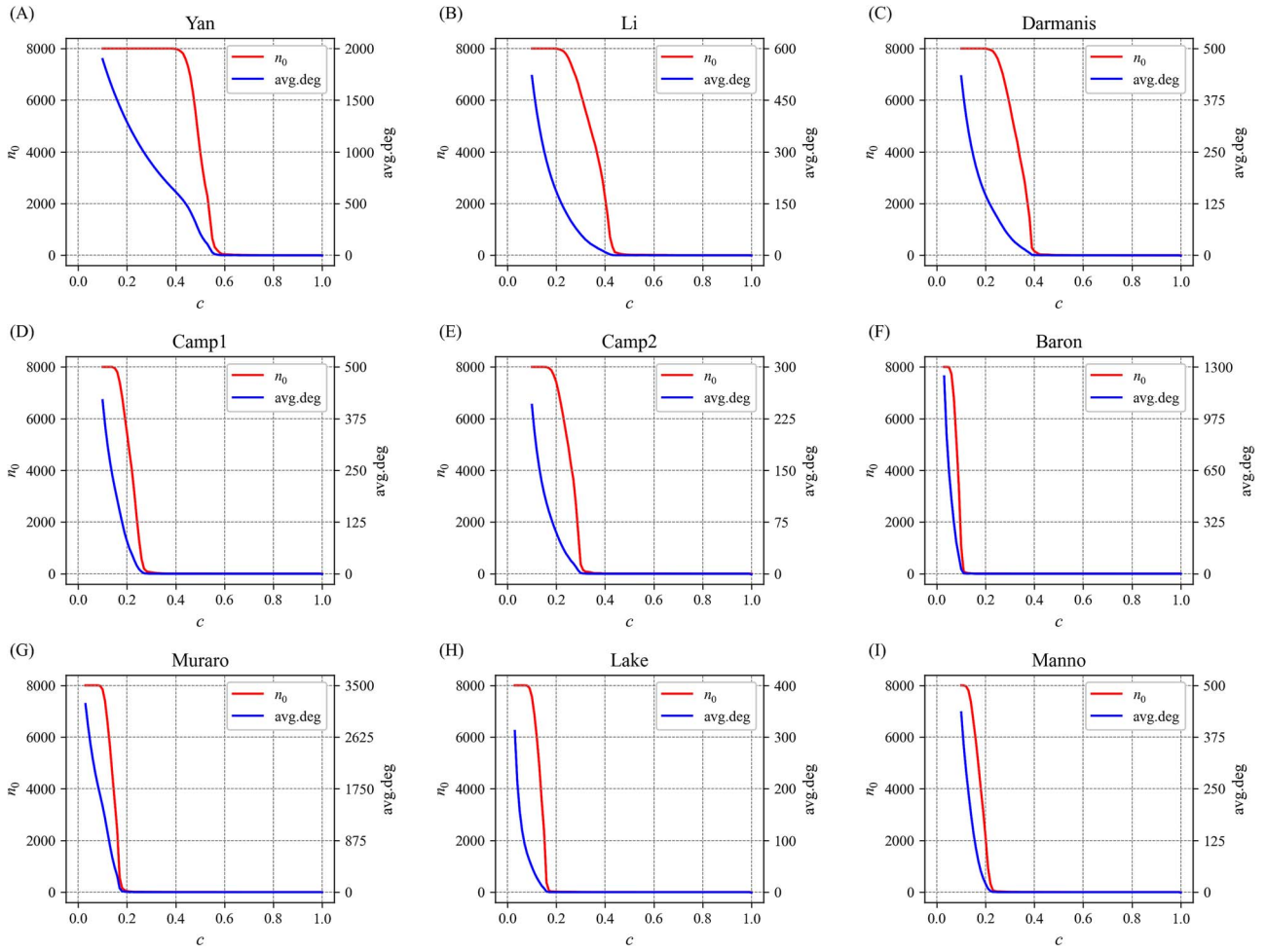


Figure 3. (A)–(I) The size of the core sub-graph and the average degree in the core sub-graph with different cutoff value  $c$  on each dataset, respectively.

than most studies concerning gene co-expression networks [51–53]. This makes our core sub-graphs more like a complete graph. Therefore, the number of nodes with low degree numbers is small. Intuitively, our degree distributions seem like Beta-distributions. Since this does not affect other analysis in our study, we did not make statistical test on this observation.

We also explored the relationship between the cutoff value  $c$  and the size of the core sub-graph. Without the restriction of  $n_d$ , the relationship between  $c$  and  $n_0$  are illustrated as Fig. 3. In Fig. 3, we also illustrated the relationship between the average degree of the core sub-graph and  $c$ . In every dataset, the  $n_0$  curve shows a cliff edge when  $c$  reaches some value. This is in line with the distribution of correlation coefficients (Supplementary Fig. S1). Most correlation coefficient values fall in a small range around zero. When the cutoff value  $c$  is beyond this range, the co-expression network collapsed quickly, making  $n_0$  a very small value. As a consequence, the whole network breaks into many very small pieces. Each piece contains very small number of nodes, which is less than four in most cases. Therefore, the average number of degrees also drops drastically. These observations guided us to choose the hyper parameter values as we have mentioned.

## Clustering performance analysis

We evaluate clustering performances of scMUG on 9 datasets, as listed in Table 1. The performance of scMUG was measured by

three indicators, including ARI, NMI, and ACC. We compared the ARI, NMI, and ACC performance values with other state-of-the-art clustering pipelines for scRNA-seq data, including scBGEDA [26], scziDesk [27], scDeepCluster [54], scGMAI [55], DCA [25], CIDR [56], SC3 [24], Scanpy [45], and Seurat [57] across all nine benchmarking datasets (Fig. 4 and Tables S3–S5 in Supplementary Materials). Ten random seeds (1111, 2222, ..., 9999, and 10,000) were applied globally in our study to validate the robustness of clustering results. The results are reported as the median value with different random seed settings. Performance differences were measured statistically (P-value in z-test, one-tail, Tables S6–S8 in Supplementary Materials).

Since we applied spectral clustering algorithms in our studies, the cluster number is a parameter that should be set. We reported and compared performances with the cluster number that generates the best NMI value. The number of clusters was scanned from 5 to 20 with a step of 1. Given the best cluster number, scMUG achieved the best NMI on 5 of 9 datasets (Li, Yan, Camp1, Baron, and Manno), and second to the best NMI on two others (Darmanis and Muraro). It also achieved the best ARI on 5 of 9 datasets (Li, Yan, Baron, Lake and Manno), and second to the best ARI on two others (Camp1 and Muraro). Particularly, it has constant superior performance on the Li dataset and the Yan dataset. Since scMUG incorporated GFM information in clustering, it looks like that a larger set of genes and a lower zero value percentage in the raw dataset help in improving its performances.

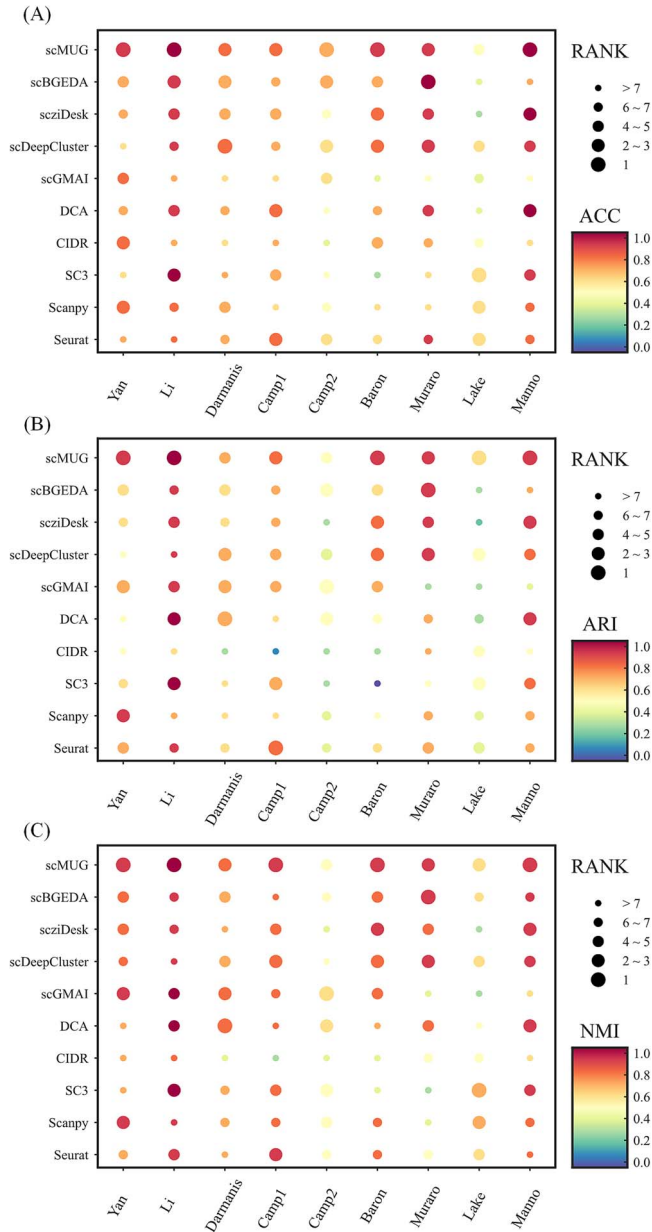


Figure 4. The ACC(A), ARI(B), and NMI(C) performance values compared with other state-of-the-art clustering pipeline for scRNA-seq data. The depth of color represents the index value, and the size of the circle represents the index rank.

To further visualize the comparison, we produced UMAP 2-D scatter plot of the cell representations with different methods in comparison on every dataset in this work (Fig. 5 and Supplementary Fig. S2). Intuitively, on the Li dataset, scMUG provides the largest margin between different cell types. On the Baron dataset, although scMUG has poor separation between ductal and acinar cells, which is similar as all methods in comparison, it provides the best separation intuitively between all other cell types, particularly the alpha, beta, delta, and gamma cells. On the Manno dataset, largest margin between hrg12a and hperic cells was achieved by scBGEDA. However, it sacrifices the margin between eprog2a, eprog1b, and eprog1a. Therefore, intuitively, the most balanced margins between cell types are still provided by scMUG.

We evaluated the running time of scMUG and other state-of-the-art methods in comparison. With the same dataset, on the same platform, the wall-clock running time of scMUG increases approximately linear along with the number of cells (Fig. 6(A)). We summed up the running time of all benchmarking dataset (Fig. 6(B)). The total running time of scMUG seems to be slightly longer than some of the other state-of-the-art methods. This is due the incorporation of GFMs. scMUG performs clustering analysis with each GFM respectively. The final results are generated after all these clustering procedures. Therefore, the total running time of scMUG is the number of GFMs times its running time for the clustering procedure. Even with this disadvantage, the total running time of scMUG is still in a comparable range to most of the methods in comparison.

## Cluster numbers and performances

Since clustering analysis aims at revealing the composition of cells and finding novel cell types, it is interesting to see what will happen if we have different settings of cluster numbers. We tried different values of clusters from 5 to 20 (Supplementary Tables S9–S11 and Supplementary Figs S3–S5). In 5 of 9 datasets (Camp2, Darmanis, Li, Yan, and Manno), we find that the best clustering performances were achieved when the number of clusters matches the true number of cell types. In three other datasets (Baron, Lake, and Muraro), we find that the best clustering performances were achieved when the cluster number was set less than the true number of cell types. Only in the Camp1 dataset, we see that the best clustering performances were achieved when the number of clusters is larger than the true number of cell types (Table 2).

To further investigate how cluster numbers affect clustering performances intuitively, we use the similarity matrix as equation (21) to produce UMAP scatter plot (Supplementary Fig. S6). In most cases, the intuitive number of clusters has no relationship with the settings. When we set the number of clusters as equal to the intuitive number of clusters, scMUG achieved the best performances. However, this intuitive number of clusters is not necessarily to be the true number of cell types. This may attribute to the unannotated cell types in these datasets. For example, in the Camp1 dataset, we see that no matter how we set the desired number of clusters, cells are clustered intuitively into nine clusters, rather than seven, which is the true number of cell types. The endothelial cell and the mesenchymal stem cell are both clearly separated into two clusters. This observation implied that these two cell types may have sub-cell types that have not been clearly annotated, which worth further investigation in life science. For another example, in the Lake dataset, the cells are better clusters when the number of clusters was set to 6, rather than 16, which is the true number of cell types. Intuitively, the ex1 cells are packed as a very tight ball, when the number of clusters was set to 6. However, it is expanded to a much wider range when the number of clusters was set to 16. This expansion reduces clustering performances.

## GFM choices and margins between different types of cells

We incorporated the GFM information in scMUG. All above results are obtained by using all significant GFMs, which we have detected in a core sub-graph in the co-expression network. Supplementary Fig. S7 provides details on how GFM is related to clustering results. In Supplementary Fig. S7, each panel presents a UMAP scatter plot of cell representations of a specific GFM on a given dataset. The details of the GFM can be found in

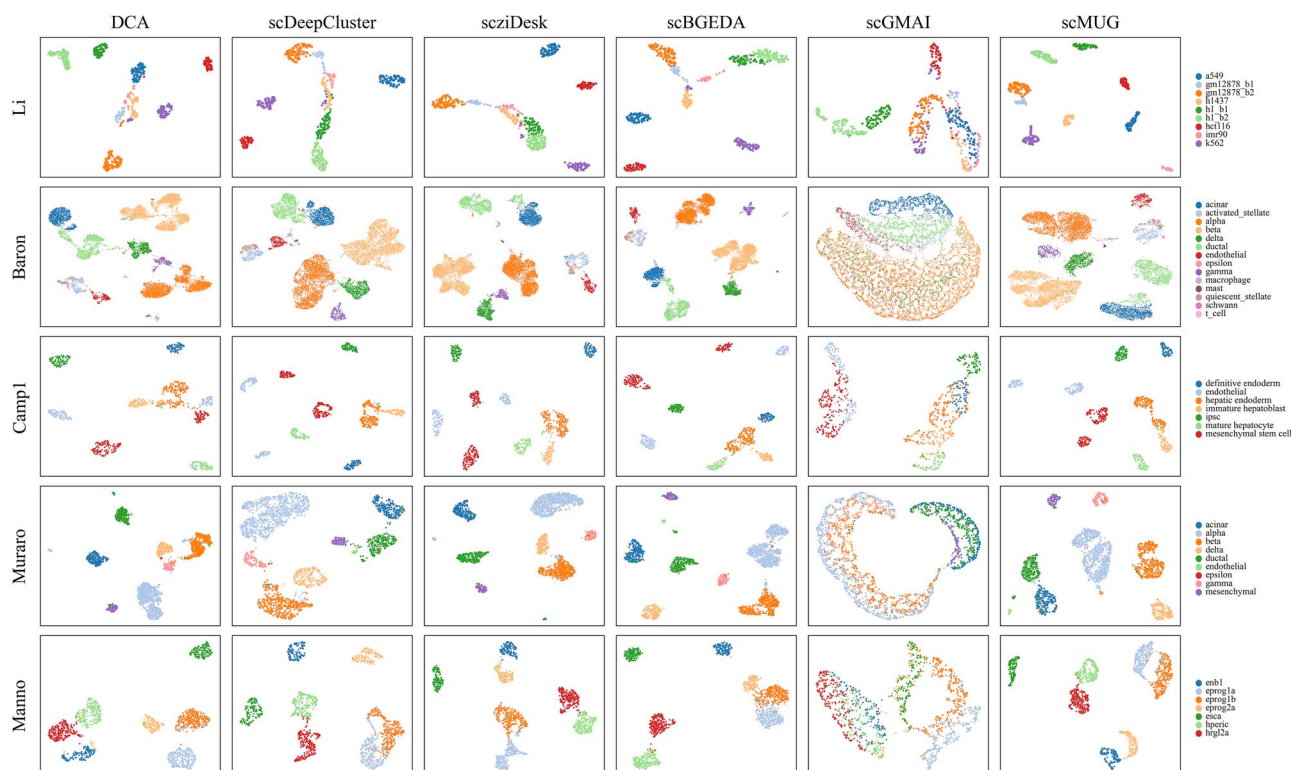


Figure 5. The UMAP scatter plot from the 32-D cell representations of scMUG and five other state-of-the-art methods in comparison. We obtained the cell representations of each method by using their own cell representation generation method. Cell representations of each method were respectively reduced to 2-D by UMAP for visualization. Each color denotes a specific cell type across all methods in a given dataset. Due to the restriction of pages, we present comparisons on five datasets (Li, Baron, Camp1, Muraro, and Manno). Figure S2 in Supplementary Materials presents the comprehensive comparisons.

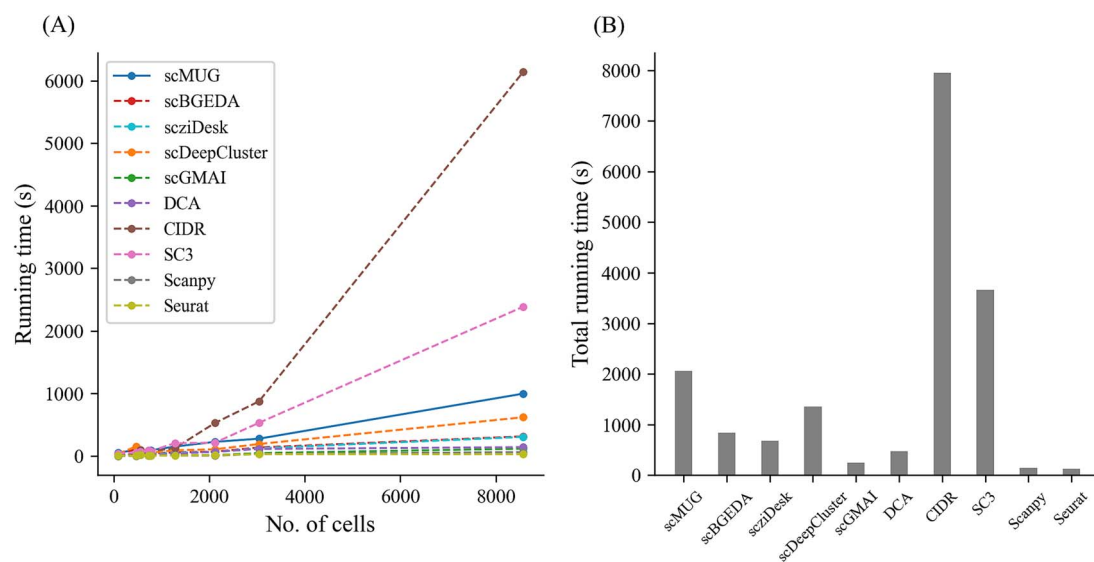


Figure 6. Comparison of running time (A) and total time cost (B) between scMUG and other clustering methods on datasets with varying cell numbers.

Table 2. Number of cell types and optimal clustering numbers of each dataset.

Dataset	Yan	Li	Darmanis	Camp1	Camp2	Baron	Muraro	Lake	Manno
True <sup>a</sup>	6	9	9	7	6	14	9	16	7
Best <sup>b</sup>	6	9	9	9	6	8	7	6	7

<sup>a</sup>True: the number of cell types that are already annotated in the dataset. <sup>b</sup>Best: the number of clusters produce the best NMI value.

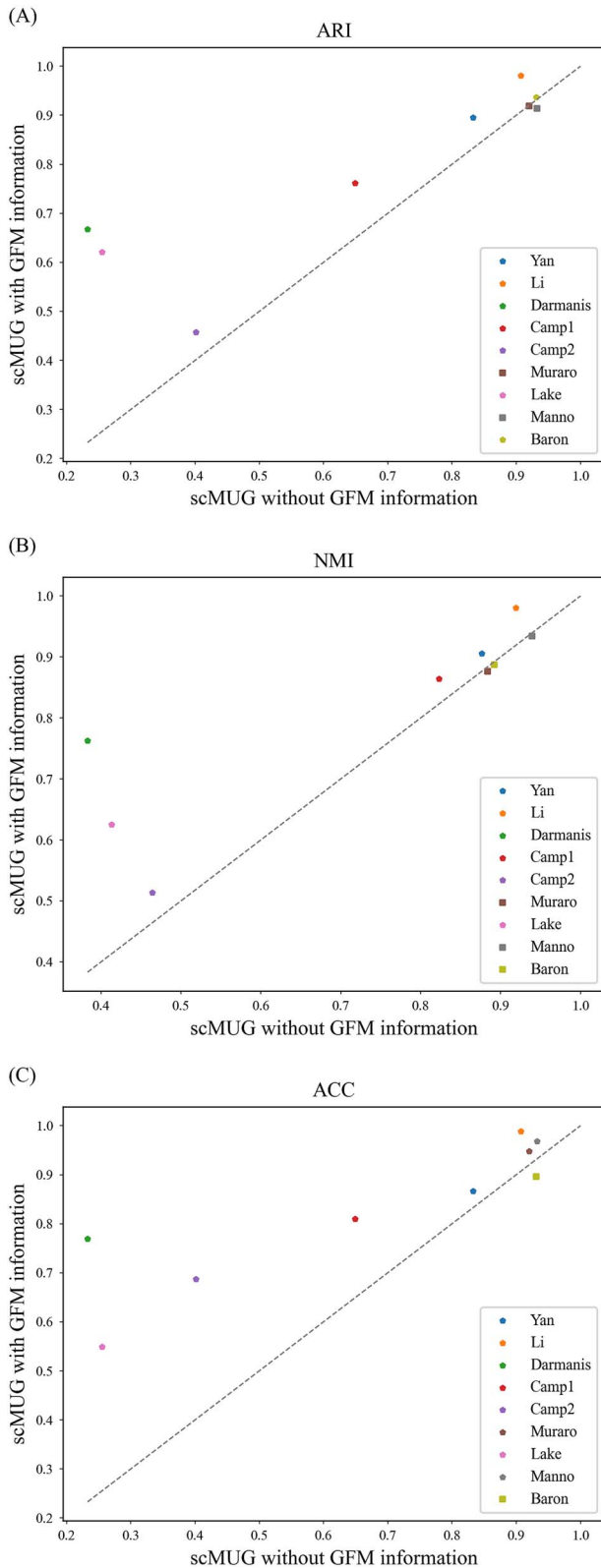


Figure 7. The performance comparison of scMUG with and without GFM information measured by ARI (A), NMI (B) and ACC (C).

[Supplementary Table S12](#). Each GFM provides a functional view of the clustering results. Different cell types have different separations and margins between them with different GFMs. This is like viewing the high dimensional clustering results from different

angles, where each angle is related to a specific set of gene functions. For example, on the Li dataset, we see that k562 cells and gm12878\_b1 cells have a minimal margin, if the first GFM is chosen. This GFM is related to the blood vessel development or angiogenesis. Therefore, scMUG provides a hint that k562 cells and gm12878\_b1 cells may have similarly roles in blood vessel development or angiogenesis functions. On the second GFM panel, genes are functionally related to cytokine-mediated signaling pathway. The k562 cells and gm12878\_b1 cells are separated, indicating their cytokine-mediated signaling pathway may function differently. In the meantime, the gm12878\_b1 cells become very close to gm12878\_b2 indicating they may share common gene expression patterns in the cytokine regulation pathways. This kind of observation happens in many cases in our study. We believe that scMUG may provide an opportunity to reveal hidden cell types from scRNA-seq data, which can be separated with others in gene functional contexts.

Since we combined all significant GFMs in performance comparisons against other state-of-the-art methods, it is interesting to see if there is any difference between our results and the results without any significant GFM identification. [Figure 7](#) presents the performance comparison of scMUG with and without GFM information. In 6 of 9 datasets (Yan, Li, Darmanis, Camp1, Camp2, Lake), combined GFMs improved the clustering performances, while no significant performance improvement can be observed in other three datasets (Muraro, Manno, Baron). Therefore, the GFM information is helpful in clustering scRNA-seq data. However, this effect is not consistent across all datasets.

## Similarity measures and clustering performances

We proposed global distribution similarity and local density similarity measures to enhance normal spectral clustering algorithms. Both similarity measures are used to establish distance matrix in the spectral clustering between cells. We compared spectral clustering result with and without these new similarity measures. [Figure 8](#) provides the ARI, NMI, and ACC violin chart of scMUG on all datasets in this work. The combination of the global distribution measure and local density measure usually provides more robust more consistent and better performance values.

To further visualize the impact of global distribution similarities and local density similarities, we use them separately as the similarity matrix between cells. UMAP scatter plots were produced from these similarity matrices ([Supplementary Fig. S8](#)). Intuitively, the global distribution similarity only provides a very tight clustering results, while the local density similarity results spread to a much larger field. For example, in the Baron dataset, the global distribution similarity mixed the acinar and ductal cells as a very tight ball. However, the local density similarities correct this by expand that pack. The combination of the global distribution and local density similarity finally provides the best separation.

## Batch effects in datasets

We used three publicly available human pancreas datasets (CelSeq [58], CelSeq2 [38], and SMART-Seq2 [59]) to explore the behavior of scMUG on a dataset with explicit or hidden batch effects. The explicit batch effects were tested by merging the three datasets directly, without any correction. The hidden batch effects were simulated by merging the three datasets and correct the batch effects before scMUG pipeline by using one of the three common correction methods (Combat [60], Harmony [61], and Scanorama [62]). Scatter plots was generated by UMAP for

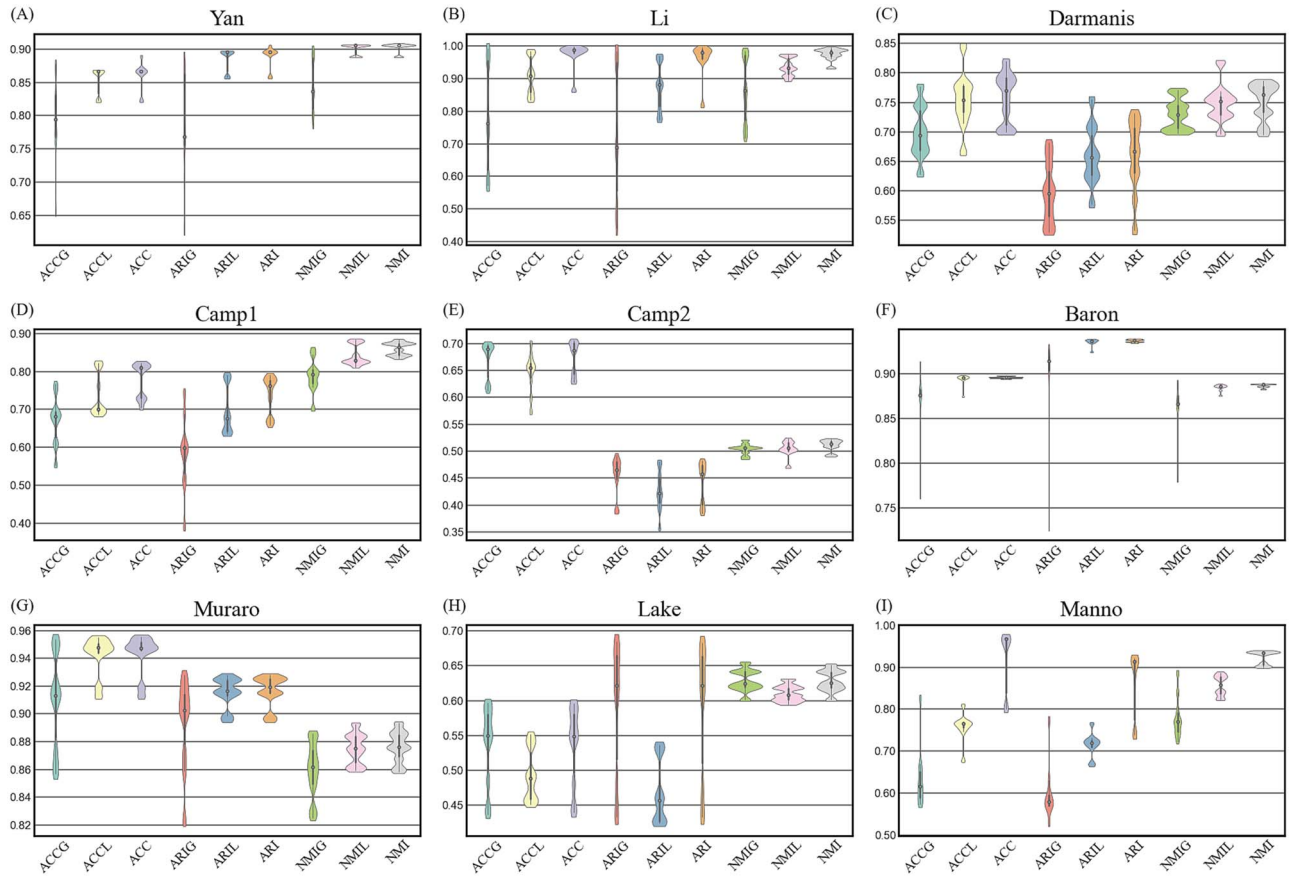


Figure 8. (A)–(I) Violin plots of ACC, ARI, and NMI of scMUG on each datasets with different types of similarity measures. We measure the similarity between cells using only global distribution similarity, only local density similarity, or both of them, respectively. The performance measure with the -G suffix is for using only global distribution similarity. The performance measure with the -L suffix is for using only local density similarity. The performance measure without and suffix is for using both local and global similarity measure.

visualization (Supplementary Fig. S9). In the input data before scMUG processing, the explicit batch effect is easy to be observed (Supplementary Fig. S9(A)). Although the correction algorithms reduced the batch effect, scatter plots still present separations between batches (Supplementary Fig. S9(B)–(D)). After the scMUG autoencoder, it seems that the dots of different colors are stirred to mix better, indicating scMUG autoencoder can partially correct batch effects (Supplementary Fig. S9(E)–(H)). However, after measuring the similarity by the local density or global distribution similarities (Supplementary Fig. S9(I)–(P)), the hidden batch effects may be revealed again. Intuitively, when using local density similarities, cells from the same batch tend to cluster more tightly, forming segments inside a single cluster. When using global distribution similarities, the batch effects are mostly corrected. Since we did not design scMUG to process datasets with batch effects, we did not optimize scMUG for this purpose. Datasets with batch effects should be better corrected by other software before entering scMUG.

### Identifying rare cell types in a dataset

We took the Baron dataset to investigate scMUG's ability to identify rare cell types. In the Baron dataset, there are 7 T cells and 25 mast cells, composing 0.04% and 0.2% of the whole dataset, respectively. In our results, both types forms clusters that can be separated from other cells. To further investigate how GFMs affect this result, we introduced small fluctuations ( $-0.02$ ,  $-0.01$ ,  $0$ ,  $+0.01$ ,  $+0.02$ ) to the expression correlation cutoff value  $c_g$ , so

that GFMs with different sizes can be generated without restrictions. When  $c_g$  increases, the GFM size decreases. The cluster of these two rare cell types becomes more distant to other cells (Supplementary Fig. S10(A)–(E)). Therefore, scMUG identify rare cell types better with smaller GFMs.

### Separating similar cells with different functional state

We took the dataset from literature [63] to see if scMUG can separate similar cell types with different functional state. The scRNA-seq expression matrix was downloaded from GEO database (GSE 120575). The cell type labels, which were generated from k-mean clustering and marker gene patterns, were obtained from literature [63]. We focused on separating the effector and exhausted T cells. The GFMs were obtained using the same algorithm as all of our benchmarking dataset. We introduced the small fluctuations ( $-0.02$ ,  $-0.01$ ,  $0$ ,  $+0.01$ ,  $+0.02$ ) to the expression correlation cutoff value  $c_g$ , so that GFMs with different sizes can be generated. We optimize scMUG with different GFM sizes. Although margins between clusters are wider than the original report, the effector and exhausted T cells were still intuitively not separated. They form two segments in a single cluster rather than two clusters (Supplementary Fig. S10(F)–(J)). These observations are in consistent with the original report [63]. Therefore, scMUG does not separate similar cells, like effector and exhausted T cells, better than other simple methods. However, with a closer look at Supplementary Fig. S10(F) and (J), we can have a vague impression

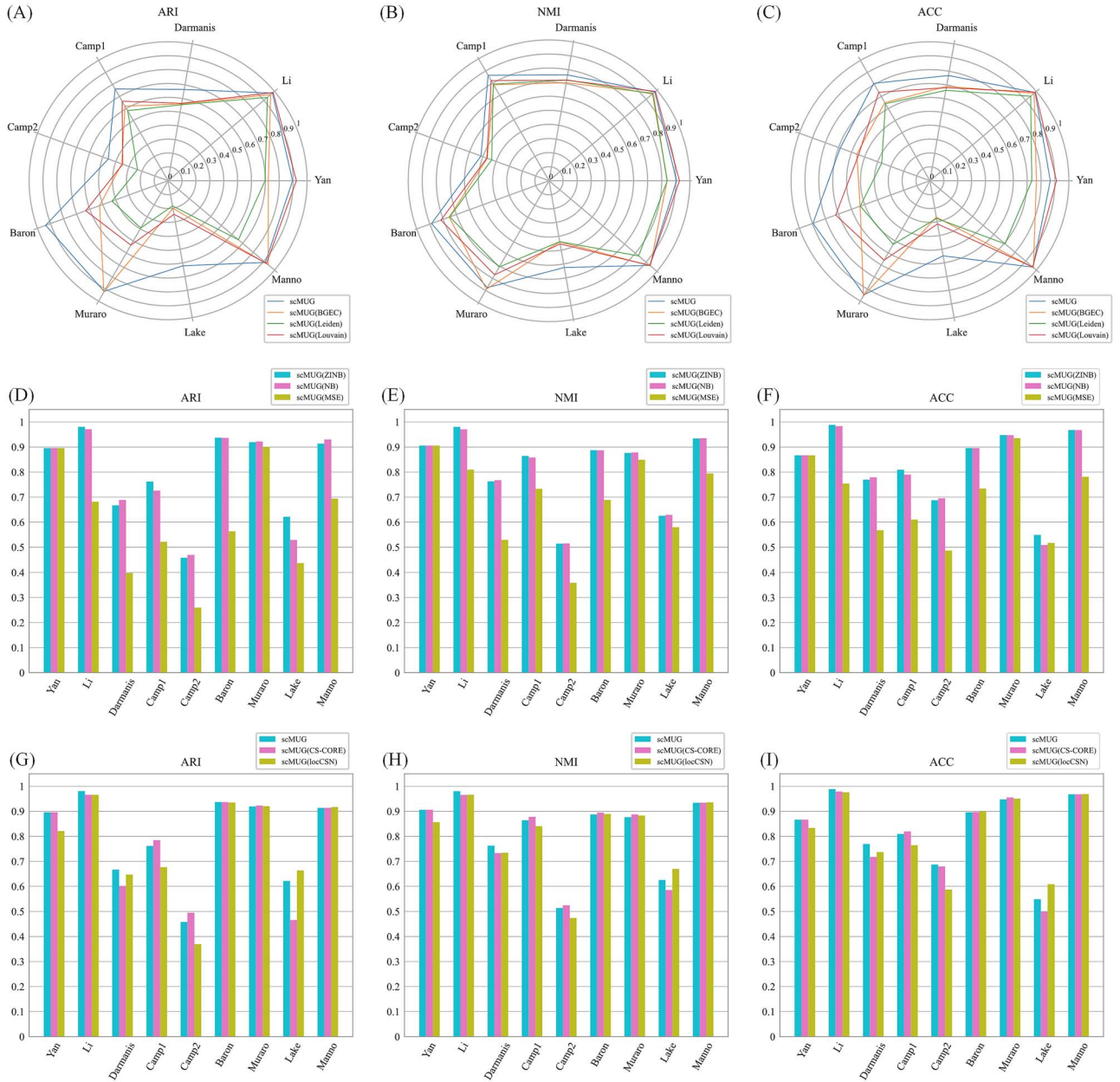


Figure 9. Ablation study. The ARI (A), NMI (B), and ACC (C) of scMUG on each dataset with different clustering algorithms (BGEC, Louvain, and Leiden) the ARI (D), NMI (E), and ACC (F) of scMUG with different loss functions (ZINB, NB, and MSE). The ARI (G), NMI (H), and ACC (I) of scMUG on each dataset with different co-expression inference algorithms (CS-CORE and locCSN).

that the boundary between the two cell types seems to be clearer when a larger GFM was introduced. Therefore, we believe that separating similar cells, like the effector and exhausted T cells, is a difficult task to scMUG. With the help of introducing sufficient gene functional information in scMUG, the results may be slightly improved.

## Ablation study

We explored the usefulness of three components in scMUG, the clustering method, the ZINB distribution-based loss function, and the co-expression inference algorithms.

We switched the clustering algorithm in scMUG to BGEC (Bipartite graph ensemble clustering) [26], Louvain [64] and Leiden [65]. In the comparison, scMUG's clustering algorithm achieved better or comparable performances across all datasets in this work (Fig. 9), supporting the design of scMUG's clustering algorithm.

We tried to use different loss function in training the autoencoder. The ZINB distribution-based loss function is compared to the NB (Negative Binomial) distribution-based loss function and the normal MSE (Mean squared error) loss function. The ZINB and NB distribution-based loss function produce consistent advantage over MSE loss function across all datasets in this work (Fig. 9). Although the margins are small, ZINB distribution-based loss function performed better than the NB distribution-based loss function on most datasets. This is in line with the fact that scRNA-seq data is better modeled with a zero-inflated distribution.

We replaced the co-expression network in scMUG with the co-expression networks that were generated by CS-CORE [66] and locCSN [67]. Figure 9(G)–(I) compared the clustering performances of scMUG with three different co-expression networks. scMUG with its original co-expression network performs better on the Li and Yan datasets. scMUG with the CS-CORE network

performs better on the Camp1 and Camp2 datasets. scMUG with locCSN network shows better results on the Lake dataset. On other datasets, all methods deliver comparable results. On the other hand, scMUG with the CS-CORE network performs poorly on the Lake dataset. scMUG with the locCSN network shows lower performance on the Yan, Camp1, and Camp2 datasets. Overall, scMUG with its original co-expression network presents the most robust performance across all datasets.

### Parameter optimization strategy

There are many parameters that can be adjusted in running scMUG. We studied the impact of these parameters, including co-expression cutoff for GFM ( $c_g$ ), number of GFMs ( $n_c$ ),  $k$ -means times ( $n_k$ ), neighborhood number ( $k$ ), weight of global distribution similarity ( $\alpha$ ), and weight of local density similarity ( $\beta$ ) (Tables S13–S17 in Supplementary Material). The whole combinatorial space of these parameters is huge, which is impossible to be enumerated. When adjusting one parameter, all other parameters are set to default values as we have stated. We suggest users of scMUG apply a similar strategy to manually optimize parameters on each individual dataset for each clustering task.

In the process of parameter optimization, we have several empirical observations. Given a desired clustering resolution, we find that the impact of  $c_g$ ,  $n_k$  and  $k$  is minimal for clustering performances. However, when the clustering resolution is increased, a lower  $c_g$  and a lower  $k$  are helpful in identifying tiny discrepancies between cells. We also find that  $n_c$  affects the performances to an unignorable level. More GFMs are more likely to produce better and more robust clustering results. Another observation is that the  $\alpha$  and  $\beta$  values should be related to the shape of clusters in UMAP visualization. If the visualization shows clusters in a curvy shape like a swiss-roll or arcs, a larger  $\beta$  is likely to give better results. This also applies when sizes of clusters vary in a large range. When visualization gives vague boundary between clusters,  $\alpha$  should be increased. Therefore, starting from some default values, and let the visualization to guide a manual optimization on  $\alpha$  and  $\beta$  may be the best choice. In addition, by sacrificing performances, reducing  $n_c$ ,  $n_k$  and forcing  $\alpha = 0$ , will significantly accelerate scMUG pipeline. This may be useful on resource limited platforms. Users of scMUG can choose their parameter optimization strategy based on our experience.

### Conclusions

We proposed the scMUG method for scRNA-seq clustering analysis. The scMUG method utilized gene functional correlations to filter gene expression profiles. Functionally correlated genes are collected as the basis for further clustering analysis. We proposed a novel similarity measure, combining local density similarity and the global distribution similarity, as the cell–cell similarities in the clustering process. The dimensionality reduction and visualization of the single-cell sequencing data using scMUG revealed clear and distinct clusters. Incorporating GFM and similarity measures into scMUG can effectively enhance the clustering performance. Comparison analysis support that scMUG has a comparable or better performance than state-of-the-art methods. Our experiments show that scMUG can partially correct batch effects in datasets, but it is recommended to apply other batch effect correction methods before using scMUG. Additionally, scMUG is effective at identifying rare cell types in a dataset, particularly when smaller GFMs are used. However, separating similar cell types, like effector and exhausted T cells, remains a challenge for scMUG, although using larger GFMs may help improve separation. Based

on the observation on clustering details, scMUG has the potential to discover hierarchical structures in the scRNA-seq dataset. It is capable of finding hidden cell types that are characterized by special gene functional groups. We hope that scMUG can be a helpful tool in single-cell life sciences.

#### Key Points

- The scMUG pipeline is the first attempt to integrate gene functional relationship in scRNA-seq clustering analysis.
- A novel cell–cell similarity measure is proposed by combining both local and global statistical features of latent cell representations.
- Evaluation results support that the scMUG pipeline can reveal hidden cell types that are related to specific gene functional groups from scRNA-seq data.

### Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

### Author contributions

DML collected the data, constructed the model, implemented the algorithm, performed experiments, analyzed results and partially wrote the manuscript. PFD supervised the whole study, conceptualized the algorithm, analyzed the results and partially wrote the manuscript.

Conflict of interest: None declared.

### Funding

This work is partially supported by National Natural Science Foundation of China [NSFC 62372320, and NSFC 61872268].

### Data availability

The code and data for reproducing results of this paper is available in a GitHub repository (<https://github.com/degiminnal/scMUG>).

### References

1. Plass M, Solana J, Wolf FA. et al. Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science* 2018;**360**:eaag1723. <https://doi.org/10.1126/science.aag1723>
2. Cao J, Spielmann M, Qiu X. et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature* 2019;**566**: 496–502. <https://doi.org/10.1038/s41586-019-0969-x>
3. Schaum N, Karkanias J, Neff NF. et al. Single-cell transcriptomics of 20 mouse organs creates a tabula Muris. *Nature* 2018;**562**: 367–72. <https://doi.org/10.1038/s41586-018-0590-4>
4. Yang F, Wang W, Wang F. et al. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nat Mach Intell* 2022;**4**:852–66. <https://doi.org/10.1038/s42256-022-00534-z>
5. Saliba A-E, Westermann AJ, Gorski SA. et al. Single-cell RNA-seq: Advances and future challenges. *Nucleic Acids Res* 2014;**42**: 8845–60. <https://doi.org/10.1093/nar/gku555>

6. Choi YH, Kim JK. Dissecting cellular heterogeneity using single-cell RNA sequencing. *Mol Cells* 2019;**42**:189–99. <https://doi.org/10.14348/molcells.2019.2446>
7. Nawy T. Single-cell sequencing. *Nat Methods* 2014;**11**:18. <https://doi.org/10.1038/nmeth.2771>
8. Gawad C, Koh W, Quake SR. Single-cell genome sequencing: Current state of the science. *Nat Rev Genet* 2016;**17**:175–88. <https://doi.org/10.1038/nrg.2015.16>
9. Kolodziejczyk AA, Kim JK, Svensson V. et al. The technology and biology of single-cell RNA sequencing. *Mol Cell* 2015;**58**:610–20. <https://doi.org/10.1016/j.molcel.2015.04.005>
10. Shapiro E, Biezuner T, Linnarsson S. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat Rev Genet* 2013;**14**:618–30. <https://doi.org/10.1038/nrg3542>
11. Zhao X, Wu S, Fang N. et al. Evaluation of single-cell classifiers for single-cell RNA sequencing data sets. *Brief Bioinform* 2020;**21**:1581–95. <https://doi.org/10.1093/bib/bbz096>
12. Bacher R, Kendzioriski C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biol* 2016;**17**:63. <https://doi.org/10.1186/s13059-016-0927-y>
13. Angerer P, Simon L, Tritschler S. et al. Single cells make big data: New challenges and opportunities in transcriptomics. *Curr Opin Syst Biol* 2017;**4**:85–91. <https://doi.org/10.1016/j.coisb.2017.07.004>
14. Sadybekov AV, Katritch V. Computational approaches streamlining drug discovery. *Nature* 2023;**616**:673–85. <https://doi.org/10.1038/s41586-023-05905-z>
15. Gupta KK, Sharma KK, Chandra H. et al. The integrative bioinformatics approaches to predict the xanthohumol as anti-breast cancer molecule: Targeting cancer cells signaling PI3K and AKT kinase pathway. *Front Oncol* 2022;**12**:950835. <https://doi.org/10.3389/fonc.2022.950835>
16. Khan S, Mosvi SN, Vohra S. et al. Implication of calcium supplementations in health and diseases with special focus on colorectal cancer. *Crit Rev Clin Lab Sci* 2024;**61**:496–509. <https://doi.org/10.1080/10408363.2024.2322565>
17. Khan S, Zakariah M, Rolfo C. et al. Prediction of mycoplasma hominis proteins targeting in mitochondria and cytoplasm of host cells and their implication in prostate cancer etiology. *Oncotarget* 2016;**8**:30830–43. <https://doi.org/10.18632/oncotarget.8306>
18. Khan S, Zakariah M, Palaniappan S. Computational prediction of mycoplasma hominis proteins targeting in nucleus of host cell and their implication in prostate cancer etiology. *Tumor Biol* 2016;**37**:10805–13. <https://doi.org/10.1007/s13277-016-4970-9>
19. Khan S, Imran A, Khan AA. et al. Systems biology approaches for the prediction of possible role of chlamydia pneumoniae proteins in the Etiology of lung cancer. *PloS One* 2016;**11**:e0148530. <https://doi.org/10.1371/journal.pone.0148530>
20. Wang Y, Imran A, Shami A. et al. Decipher the helicobacter pylori protein targeting in the nucleus of host cell and their implications in gallbladder cancer: An insilico approach. *J Cancer* 2021;**12**:7214–22. <https://doi.org/10.7150/jca.63517>
21. Satija R, Farrell JA, Gennert D. et al. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 2015;**33**:495–502. <https://doi.org/10.1038/nbt.3192>
22. Butler A, Hoffman P, Smibert P. et al. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;**36**:411–20. <https://doi.org/10.1038/nbt.4096>
23. Stuart T, Butler A, Hoffman P. et al. Comprehensive integration of single-cell data. *Cell* 2019;**177**:1888–1902.e21. <https://doi.org/10.1016/j.cell.2019.05.031>
24. Kiselev VY, Kirschner K, Schaub MT. et al. SC3: Consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**:483–6. <https://doi.org/10.1038/nmeth.4236>
25. Eraslan G, Simon LM, Mircea M. et al. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun* 2019;**10**:390. <https://doi.org/10.1038/s41467-018-07931-2>
26. Wang Y, Yu Z, Li S. et al. scBGEDA: Deep single-cell clustering analysis via a dual denoising autoencoder with bipartite graph ensemble clustering. *Bioinformatics* 2023;**39**:btad075. <https://doi.org/10.1093/bioinformatics/btad075>
27. Chen L, Wang W, Zhai Y. et al. Deep soft K-means clustering with self-training for single-cell RNA sequence data. *NAR Genom Bioinform* 2020;**2**:lqaa039. <https://doi.org/10.1093/nargab/lqaa039>
28. Fang Z, Zheng R, Li M. scMAE: A masked autoencoder for single-cell RNA-seq clustering. *Bioinformatics* 2024;**40**:btac020. <https://doi.org/10.1093/bioinformatics/btac020>
29. Wang J, Ma A, Chang Y. et al. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nat Commun* 2021;**12**:1882. <https://doi.org/10.1038/s41467-021-22197-x>
30. Ciortan M, Defrance M. GNN-based embedding for clustering scRNA-seq data. *Bioinformatics* 2022;**38**:1037–44. <https://doi.org/10.1093/bioinformatics/btab787>
31. Yin Q, Liu Q, Fu Z. et al. scGraph: A graph neural network-based approach to automatically identify cell types. *Bioinformatics* 2022;**38**:2996–3003. <https://doi.org/10.1093/bioinformatics/btac199>
32. Greene CS, Krishnan A, Wong AK. et al. Understanding multicellular function and disease with human tissue-specific networks. *Nat Genet* 2015;**47**:569–76. <https://doi.org/10.1038/ng.3259>
33. Krishnan A, Zhang R, Yao V. et al. Genome-wide prediction and functional characterization of the genetic basis of autism spectrum disorder. *Nat Neurosci* 2016;**19**:1454–62. <https://doi.org/10.1038/nn.4353>
34. Zhou J, Theesfeld CL, Yao K. et al. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat Genet* 2018;**50**:1171–9. <https://doi.org/10.1038/s41588-018-0160-6>
35. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat Methods* 2015;**12**:931–4. <https://doi.org/10.1038/nmeth.3547>
36. Chen KM, Wong AK, Troyanskaya OG. et al. A sequence-based global map of regulatory activity for deciphering human genetics. *Nat Genet* 2022;**54**:940–9. <https://doi.org/10.1038/s41588-022-01102-2>
37. Li H, Courtois ET, Sengupta D. et al. Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nat Genet* 2017;**49**:708–18. <https://doi.org/10.1038/ng.3818>
38. Muraro MJ, Dharmadhikari G, Grün D. et al. A single-cell transcriptome atlas of the human pancreas. *Cell Syst* 2016;**3**:385–394.e3. <https://doi.org/10.1016/j.cels.2016.09.002>
39. Baron M, Veres A, Wolock SL. et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *cells* 2016;**3**:346–360.e4. <https://doi.org/10.1016/j.cels.2016.08.011>
40. Lake BB, Ai R, Kaeser GE. et al. Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. *Science* 2016;**352**:1586–90. <https://doi.org/10.1126/science.aaf1204>
41. La Manno G, Gyllborg D, Codeluppi S. et al. Molecular diversity of midbrain development in mouse, human, and stem cells. *Cell* 2016;**167**:566–580.e19. <https://doi.org/10.1016/j.cell.2016.09.027>

42. Camp JG, Badsha F, Florio M. et al. Human cerebral organoids recapitulate gene expression programs of fetal neocortex development. *Proc Natl Acad Sci* 2015;**112**:15672–7. <https://doi.org/10.1073/pnas.1520760112>
43. Darmanis S, Sloan SA, Zhang Y. et al. A survey of human brain transcriptome diversity at the single cell level. *Proc Natl Acad Sci* 2015;**112**:7285–90. <https://doi.org/10.1073/pnas.1507125112>
44. Yan L, Yang M, Guo H. et al. Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells. *Nat Struct Mol Biol* 2013;**20**:1131–9. <https://doi.org/10.1038/nsmb.2660>
45. Wolf FA, Angerer P, Theis FJ. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol* 2018;**19**:15. <https://doi.org/10.1186/s13059-017-1382-0>
46. Herrmann M, Kazempour D, Scheipl F. et al. Enhancing cluster analysis via topological manifold learning. *Data Min Knowl Disc* 2024;**38**:840–87. <https://doi.org/10.1007/s10618-023-00980-2>
47. McInnes L, Healy J, Saul N. et al. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software* 2018;**3**:861. <https://doi.org/10.21105/joss.00861>
48. Liang X, Cao L, Chen H. et al. A critical assessment of clustering algorithms to improve cell clustering and identification in single-cell transcriptome study. *Brief Bioinform* 2024;**25**:bbad497. <https://doi.org/10.1093/bib/bbad497>
49. Newman MEJ. The structure and function of complex networks. *SIAM Rev* 2003;**45**:167–256. <https://doi.org/10.1137/S003614450342480>
50. Alexandre PA, Hudson NJ, Lehnert SA. et al. Genome-wide Co-expression distributions as a metric to prioritize genes of functional importance. *Genes* 2020;**11**:1231. <https://doi.org/10.3390/genes11101231>
51. Zhang B, Horvath S. A general framework for weighted gene Co-expression network analysis. *Stat Appl Genet Mol Biol* 2005;**4**:17. <https://doi.org/10.2202/1544-6115.1128>
52. Luo F, Yang Y, Zhong J. et al. Constructing gene co-expression networks and predicting functions of unknown genes by random matrix theory. *BMC Bioinform* 2007;**8**:1–17. <https://doi.org/10.1186/1471-2105-8-299>
53. van Dam S, Vösa U, van der Graaf A. et al. Gene co-expression analysis for functional classification and gene–disease predictions. *Brief Bioinform* 2018;**19**:575–92. <https://doi.org/10.1093/bib/bbw139>
54. Tian T, Wan J, Song Q. et al. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nat Mach Intell* 2019;**1**:191–8. <https://doi.org/10.1038/s42256-019-0037-0>
55. Yu B, Chen C, Qi R. et al. scGMAI: A Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder. *Brief Bioinform* 2021;**22**:bbaa316. <https://doi.org/10.1093/bib/bbaa316>
56. Lin P, Troup M, Ho JWK. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol* 2017;**18**:59. <https://doi.org/10.1186/s13059-017-1188-0>
57. Hao Y, Stuart T, Kowalski MH. et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nat Biotechnol* 2024;**42**:293–304. <https://doi.org/10.1038/s41587-023-01767-y>
58. Grün D, Muraro MJ, Boisset J-C. et al. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell* 2016;**19**:266–77. <https://doi.org/10.1016/j.stem.2016.05.010>
59. Segerstolpe Å, Palasantza A, Eliasson P. et al. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab* 2016;**24**:593–607. <https://doi.org/10.1016/j.cmet.2016.08.020>
60. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**:118–27. <https://doi.org/10.1093/biostatistics/kxj037>
61. Korsunsky I, Millard N, Fan J. et al. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat Methods* 2019;**16**:1289–96. <https://doi.org/10.1038/s41592-019-0619-0>
62. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat Biotechnol* 2019;**37**:685–91. <https://doi.org/10.1038/s41587-019-0113-3>
63. Sade-Feldman M, Yizhak K, Bjorgaard SL. et al. Defining T cell states associated with response to checkpoint immunotherapy in melanoma. *Cell* 2018;**175**:998–1013.e20. <https://doi.org/10.1016/j.cell.2018.10.038>
64. Blondel VD, Guillaume J-L, Lambiotte R. et al. Fast unfolding of communities in large networks. *J Stat Mech* 2008;**2008**:P10008. <https://doi.org/10.1088/1742-5468/2008/10/P10008>
65. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: Guaranteeing well-connected communities. *Sci Rep* 2019;**9**:5233. <https://doi.org/10.1038/s41598-019-41695-z>
66. Su C, Xu Z, Shan X. et al. Cell-type-specific co-expression inference from single cell RNA-sequencing data. *Nat Commun* 2023;**14**:4846. <https://doi.org/10.1038/s41467-023-40503-7>
67. Wang X, Choi D, Roeder K. Constructing local cell-specific networks from single-cell data. *Proc Natl Acad Sci* 2021;**118**:e2113178118. <https://doi.org/10.1073/pnas.2113178118>