

scDFN: Enhancing single-cell RNA-seq Clustering with Deep Fusion Networks

Tianxiang Liu

Dalian Maritime University

Yue Bi

Monash University

Xudong Guo

North West Agriculture and Forestry University

Quan Zou

University of Electronic Science and Technology of China

Cangzhi Jia

cangzhijia@dlnu.edu.cn

Dalian Maritime University

Fuyi Li

North West Agriculture and Forestry University

Research Article

Keywords: Single-cell RNA sequencing, Clustering, Deep Learning, Clustering

Posted Date: May 8th, 2024

DOI: <https://doi.org/10.21203/rs.3.rs-4356835/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

scDFN: Enhancing single-cell RNA-seq Clustering with Deep Fusion Networks

Tianxiang Liu¹, Yue Bi², Xudong Guo³, Quan Zou⁴, Cangzhi Jia^{1*}, Fuyi Li^{3,5*}

¹School of Science, Dalian Maritime University, Dalian, 116026, China

²Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3153, Australia

³College of Information Engineering, Northwest A&F University, Yangling, 712100, China

⁴Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, 611730, China

⁵South Australian Immunogenomics Cancer Institute (SAiGENCI), The University of Adelaide, SA 5005, Australia

***Corresponding authors:** (1) Cangzhi Jia, cangzhijia@dlmu.edu.cn, ORCID: 0000-0002-4682-2881; (2) Fuyi Li, fuyi.li@nwafu.edu.cn, ORCID: 0000-0001-5216-3213.

Abstract

Single-cell RNA sequencing (scRNA-seq) technology can be used to perform high-resolution analysis of the transcriptomes of individual cells. Therefore, its application has gained popularity for accurately analyzing the ever-increasing content of heterogeneous single-cell datasets. Central to interpreting scRNA-seq data is the clustering of cells to decipher transcriptomic diversity and infer cell behavior patterns. Although clustering plays a key role in the subsequent analysis of single-cell transcriptomics, its complexity necessitates the application of advanced methodologies capable of resolving the inherent heterogeneity and limited gene expression characteristics of single-cell data. In this study, we introduced a novel deep learning-based algorithm for single-cell clustering, designated scFCN, which can significantly enhance the clustering of scRNA-seq data through a fusion network strategy. The scFCN algorithm applies a dual mechanism involving an autoencoder to extract attribute information and an improved graph autoencoder to capture topological nuances, integrated via a cross-network information fusion mechanism complemented by a triple self-supervision strategy. This fusion is optimized through a holistic consideration of four distinct loss functions. A comparative analysis with five leading scRNA-seq clustering methodologies across multiple datasets revealed the superiority of scFCN, as determined by better Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) metrics. Additionally, scFCN demonstrated robust multi-cluster dataset performance and exceptional resilience to batch effects. Ablation studies highlighted the key roles of the autoencoder and the improved graph autoencoder components, along with the critical contribution of the four joint loss functions to the overall efficacy of the algorithm. Through these advancements, scFCN set a new benchmark in single-cell clustering and can be used as an effective tool for the nuanced analysis of single-cell transcriptomics. The source code of scFCN is publicly available at <https://github.com/11051911/scDFN>.

Keywords: Single-cell RNA sequencing, Clustering, Deep Learning, Clustering

1. Background

Single-cell RNA sequencing (scRNA-seq) technology represents a considerable advancement in high-throughput technology. It can be used to analyze the transcriptome of individual cells at the finest level [1]. This technique differs from conventional bulk RNA-Seq methodologies, which miss the intricate heterogeneity among cells [2]. The ever-increasing content of heterogeneous single-cell data highlights the need for the development of efficient tools specifically for conducting single-cell transcriptome analysis [3]. Cell clustering is an important technique that can be used to determine cell heterogeneity and perform cell development trajectory analysis [4], as well as, the KEGG pathway analysis and Gene Ontology (GO) analysis. By grouping cells based on their expression matrices, clustering facilitates the elucidation of internal structural information and molecular profiles, thus, influencing the quality of downstream single-cell transcriptomics analyses. Single-cell clustering, however, is hindered by the high heterogeneity and low gene expression rates of single-cell transcriptomics data.

Conventional clustering methods, such as k -means clustering [5], hierarchical clustering [6], principal component analysis (PCA), and KNN (K-Nearest Neighbors), have been used to develop cell clustering approaches. For example, SC3 [7] uses genetic filtering along with PCA and Laplace transforms, enhancing k -means with hierarchical clustering to refine the clustering outcome by integrating the initial value and condition variations. This approach addresses the limitations of greedy algorithms by ensuring consistency. Additionally, approaches like PARC [8] and Seurat [9] utilize KNN to allocate cells to their nearest clusters based on computational distances and neighbor counts, which can vary significantly. TSCAN is an unsupervised approach that links gene expression to the temporal or spatial orientations of cells, enabling cell clustering via trajectory inference analysis [10]. However, when dealing with large datasets, conventional clustering techniques struggle with scalability, and the efficacy of methods like KNN depends heavily on the choice of the definition of computational distance and the number of nearest neighbors.

Deep learning has introduced various sophisticated methods for clustering and analyzing scRNA-seq data. These advanced single-cell data clustering approaches are categorized into five main groups based on their network optimization objectives. These groups include generative adversarial network-based methods (*e.g.*, scGPCL [11] and scDECL [12]), subspace clustering-based methods (*e.g.*, scBGEDA [13]), Gaussian mixture model-based methods (*e.g.*, scSSA [14]), spectral clustering-based methods (*e.g.*, Secuer [15] and scDSSC [16]), and self-optimization-based methods (*e.g.*, scziDesk [17], scDeepCluster [18], DESC [19], and GraphSCC [20]). The methods are described in **Table 1**. A summary of deep learning-based single cell clustering methods.. For example, scDSSC [16] applies an autoencoder for noise reduction and dimensionality reduction and performs spectral clustering through a sparse self-expressive matrix (SEM). However, the model uses different hyperparameters for different datasets while training. DESC [19] obtains parameters through an automatic encoder and learns the spatial structure by iteratively optimizing the clustering objective function. Additionally, DESC uses the traditional MSE loss as the data reconstruction error and ignores the

distance between similar cells; thus, it cannot maintain the global and local structure of the data. DCA [21] uses a deep count autoencoder network to denoise the scRNA-seq data. In scDeepCluster [18], autoencoders are trained to denoise the data by simultaneously reducing reconstruction loss and separation loss, which helps in learning transcriptome expression and clustering features simultaneously. However, as it does not pre-select highly variable genes as input features, it has low clustering accuracy, is time-consuming, and requires a lot of memory. The scSSA [14] uses a semi-supervised autoencoder and fastICA for dimensionality reduction and constructs a Gaussian mixture model that yields accurate clustering results. However, different epochs may be required to minimize the loss in different datasets. Because it uses a semi-supervised autoencoder, at least some cells need to be labeled while training. Otherwise, the clustering performance may be greatly affected. The scziDesk [17] model combines deep learning techniques with denoising autoencoders to characterize scRNA-seq data, and a soft self-training k -means algorithm is used to cluster cell populations in the potential learning space. The scBGEDA [13] performs single-cell clustering by a dual denoising autoencoder and bipartite graph ensemble clustering algorithm. Most of the models mentioned above use k -means clustering for initialization, and the results are optimized based on clustering loss. However, they ignore the structural information among cells and face difficulties while dealing with large datasets. In scTAG [22], a topologically adaptive graph convolutional autoencoder is developed to learn cell-cell topological representations. A hypothesis-free cell learning framework, known as scGNN [23], uses graph neural networks to characterize the relationship between cells. To preserve the topological structure and attribute information of the scRNA-seq data, multi-task-oriented graph autoencoders are applied by scGAE [24] for dimensionality reduction. GraphSCC [20] combines graph convolutional networks (GCNs) and denoising autoencoders to integrate the structural information of scRNA-seq data and a dual self-supervised module is designed to optimize the latent representation. In scMGCA [25], a graph-embedded autoencoder is constructed to learn cell-cell topological representations and cluster assignments simultaneously. Notably, Ding et al [26] provides an end-to-end toolkit DANCE, which supports 3 modules and 8 tasks with 32 state-of-art methods on 21 benchmark datasets. However, graph autoencoders often miss key patterns in gene expression data, which is a major limitation of these methods.

Based on these approaches, several deep graph clustering methods have been developed, which combine autoencoder and graph encoder to extract the structural and attribute information of graphs. This combination has advantages over an autoencoder or a graph autoencoder used alone. SDCN [27] integrates multiple representations and structures of the data based on a dual self-supervised combination of autoencoder and GCN. In this process, the autoencoder and GCN help each other to improve clustering performance. DFCN [28] captures the shared representation of local and global information in input data by an autoencoder and a graph encoder. These results suggest that combining classic autoencoder and graph autoencoder is an effective method to extract cell information and cell-cell topology information of scRNA-seq data.

In this study, we propose a self-optimization deep single-cell clustering algorithm known as scFCN, which can extract the attribute information and structure information of scRNA-seq data through a fusion network. The scFCN has three main modules, including a data processing module, an information coding module, and an information fusion module. The data processing module is used to select highly variable genes and reduce the dimension of the features of the input scRNA-seq data. The information coding module extracts the cell representation information through the autoencoder and the improved graph autoencoder. This is the first study to apply the augmented graph network for the topological representation of data at the single-cell level. The information fusion module combines the extracted attribute and topology information based on the cross-network information fusion mechanism and the triple self-supervision strategy. Quadruple joint losses are adopted to optimize the cell clustering representation in the whole model. The results of a comprehensive evaluation and analysis confirmed that scDFN can improve scRNA-seq clustering effectively. Compared to five state-of-the-art scRNA-seq clustering methods, scDFN achieves better NMI and ARI on multiple datasets. Additionally, scDFN can be used to analyze multi-cluster datasets effectively. Ablation studies have not only shown the necessity of an autoencoder and an improved graph autoencoder but also revealed the importance of four joint losses. Our scDFN method also exhibits excellent robustness during clustering and effectively eliminates batch effects.

Table 1. A summary of deep learning-based single cell clustering methods.

Method Classification	Tool	Year	Model	Objective	Autoencoder	Hyperparameters consistency
Generative adversarial network-based	scGPCL	2023	Graph prototypical contrastive learning	Clustering	GNN-based autoencoder	No
	scDECL	2023	Contrastive learning and pairwise constraints	Clustering	Deep autoencoder	No
Subspace clustering-based	scBGEDA	2023	Dual denoising autoencoder with bipartite graph ensemble clustering	Clustering and interpretable analysis	Denoising autoencoder	Yes
Gaussian mixture model-based	scSSA	2022	Semi-supervised autoencoder	Clustering	Deep autoencoder	Yes
Spectral clustering-based	Secuer	2022	Construct weighted bipartite graph, k-means clustering	Clustering	None	No
	scDSSC	2022	Autoencoder, subspace clustering	Clustering and interpretable analysis	Denoising autoencoder	No
Self-optimization-based methods	scDeepCluster	2019	Autoencoder	Clustering	Denoising autoencoder	No
	DCA	2019	Deep count autoencoder	Clustering and interpretable analysis	Deep autoencoder	No

scziDesk	2020	Autoencoder, soft K-means clustering	Clustering and interpretable analysis	Deep autoencoder	No
DESC	2020	Stacked autoencoder, iterative clustering	Clustering and interpretable analysis	Stacked autoencoder	Yes
GraphSCC	2020	Graph convolutional network, Denoising Autoencoder	Clustering and interpretable analysis	Denoising Autoencoder	Yes
scGNN	2021	Graph neural networks, left-truncated mixture Gaussian model	Clustering and interpretable analysis	Deep autoencoder and graph autoencoder	Yes
scGAE	2021	Multitask-oriented graph autoencoder	Clustering and interpretable analysis	Graph autoencoder	Yes
scTAG	2022	Graph embedding clustering	Clustering	Deep graph convolutional autoencoder	Yes
scMGCA	2023	Graph-embedded autoencoder	Clustering and interpretable analysis	Graph autoencoder	Yes
scDEFR	2023	Fusion Cell Topology Encoder, Transcriptomics Profile-based Graph Encoder	Clustering	Deep autoencoder and graph autoencoder	Yes

2. Results and Discussion

2.1 Overall framework

The scDFN method we developed mainly included three parts for optimizing clustering results by learning a low-dimensional representation of cells, whose overall framework is illustrated in **Figure 1**. The first part involved data processing using the Python Scanpy package [29]. We selected M , a highly variable gene, and plotted a K-Nearest Neighbor (KNN) graph as the input of the improved graph autoencoder (IGAE). The second part involved the establishment of the framework of AE and IGAE to obtain a low-dimensional potential representation of cells. In this part, AE was used to integrate the attribute information of cells and obtain low-dimensional potential embedding. IGAE may be used to extract the main cell information and the topological structure of the cells by learning the expression matrix and the cell graph. The third part was the most important; in the third part, the zero-inflated negative binomial model (ZINB) distribution was used to not only simulate the distribution of cells but also to act as a decoder to reconstruct the representation of genes. The Mean Square Error (MSE) reconstruction loss was used to reconstruct the cell representations of AE and IGAE. The self-optimizing clustering task based on the Kullback-Leibler (KL) divergence was performed on the embedded representation, which we called the triple self-monitoring mechanism. By

simultaneously correcting clustering loss, ZINB-based loss, and cell graph and expression matrix reconstruction loss, we captured the optimal attribute information and topology information of cells.

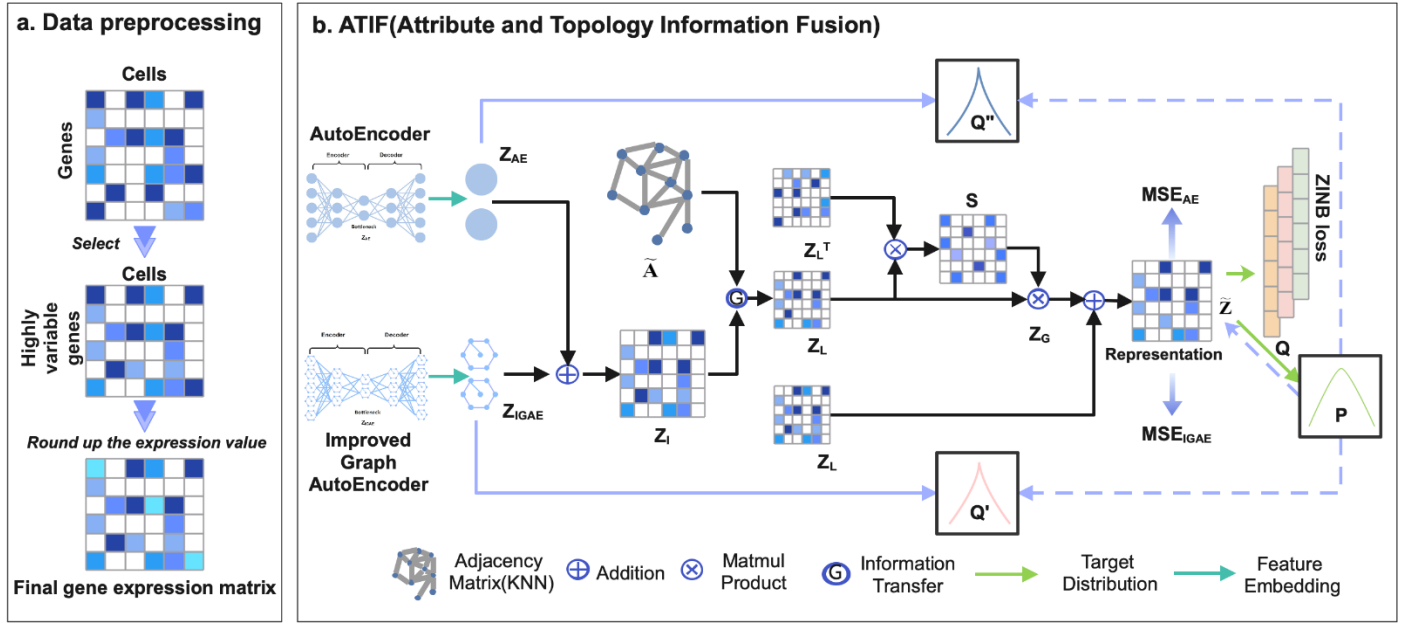


Figure 1. Network architecture of scDFN. **a.** The data preprocessing module. **b.** Attribute and topology information fusion (ATIF).

2.2 Performance comparisons with five state-of-the-art scRNA-seq clustering methods

We compared scDFN with five advanced single-cell clustering methods using 32 real scRNA-seq datasets originating from multiple platforms to evaluate the clustering performance of scDFN. These methods, including scMGCA, scDEFR, scTAG, scDeepCluster, and scDSSC, are the most advanced methods developed in recent years and have achieved good clustering performance. To ensure that the comparison was fair, the method of the baseline model was analyzed and found to be consistent with the original publication. The results of the comparison are summarized in **Figure 2** and Supplementary Tables S2 and S3. We found that scDFN achieved the highest average scores of NMI (0.8131) and ARI (0.7518) on all 32 datasets, and scDFER achieved the second highest scores (NMI = 0.8161 and ARI = 0.7518) (Figures 2A and 2B). Specifically, the NMI score of scDFN performed well on 15 datasets, with the highest score of 0.9693 recorded for the QS_Diaphragm dataset [30], and scDFER performed well on seven datasets, with the best score of 0.9637 for the QS_Limb_Muscle dataset (**Figure 2g**). The ARI value of scDFN performed well on 16 datasets, with the highest score of 0.9693 for the QS_Diaphragm dataset; the ARI value of the performance of scDSSC on six datasets was the second highest (**Figure 2h**). We also compared the overall clustering performance (NMI and ARI) of the methods on various data platforms, including plate-based platform, flow cell-based platform, Smart-Seq 2, SMARTer, 10X Genomics, and inDrop (Figures 2c-f). The experimental results showed that the clustering performance of the scMGCA algorithm on cross-platforms was better than the performance of the other five clustering algorithms, which confirmed

the effectiveness and accuracy of the scMGCA algorithm in cross-platform clustering.

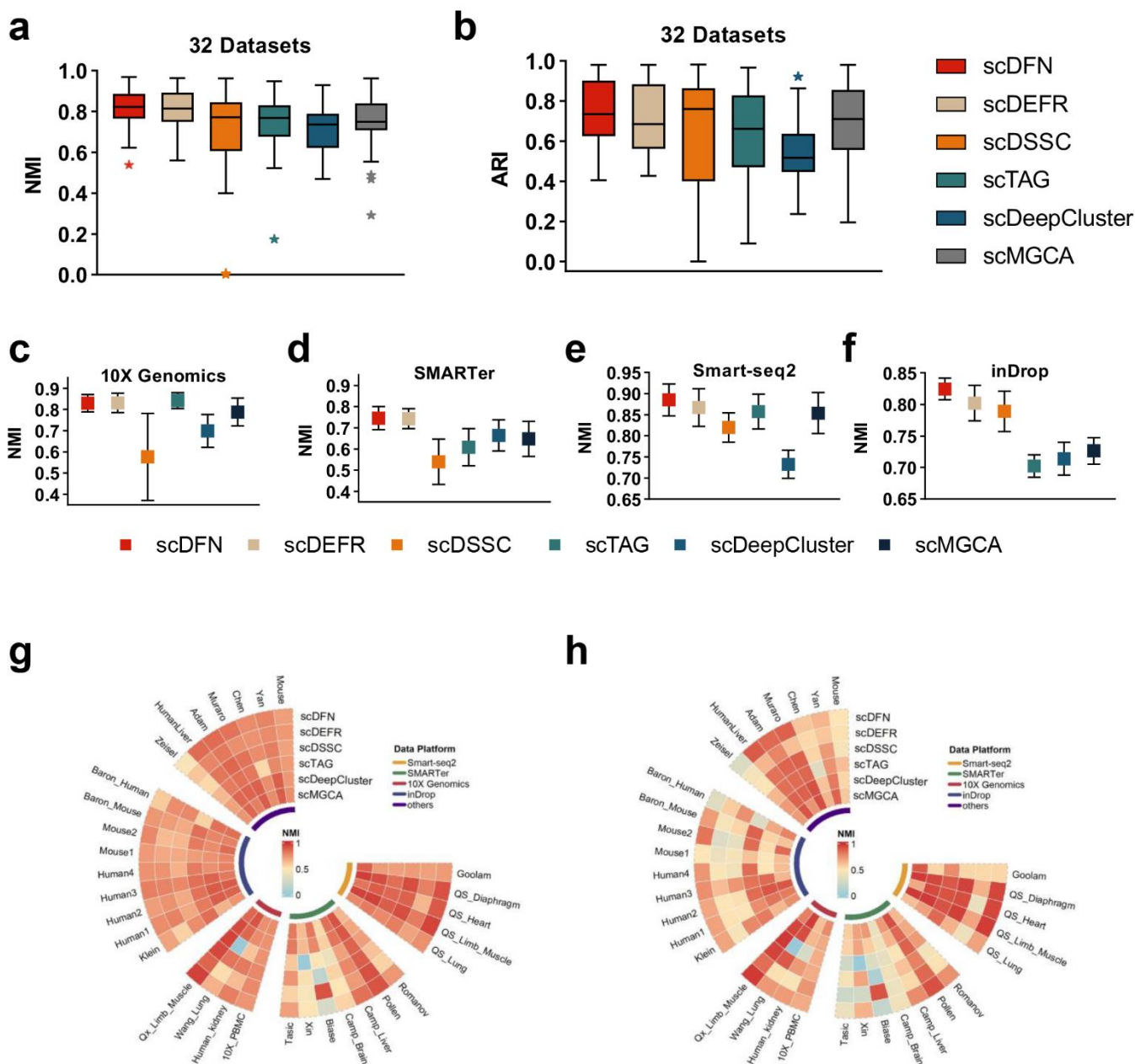


Figure 2. (a and b) A comparison of the average value of NMI and ARI on the 32 real scRNA-seq datasets among six clustering methods. (c-f) A comparison of the mean value of NMI on various data platforms among six clustering methods. (g and h) Specific numerical values for each method on each dataset.

2.3 Performance on multi-cluster datasets

The analysis of single-cell RNA sequencing (scRNA-seq) datasets with a large number of cell clusters is challenging because unsupervised learning methods are affected by the number of cell clusters [31]. We found that scDFN can achieve better NMI and ARI performance on datasets with a large number of clusters (Supplementary Tables S2 and S3). For example, for the Chen dataset (46 clusters) and the TASIC dataset (49 clusters), the NMI scores of scDFN were 0.7655 and 0.8178, and the ARI scores were 0.6420 and 0.5689, respectively. The high performance of scDFN occurred mainly because the newly constructed interactive potential space can capture the subtle differences in different cell

clusters. For example, the TASIC dataset included 49 clusters, among which nine clusters included less than 15 cells per cluster (Supplementary Table S10). The original cell expression matrix of TASIC is visualized in **Figure 3a**, which demonstrates that several clusters were mixed. After the interactive potential space was reconstructed in scDFN, most clusters were separated (**Figure 3b**). To evaluate how scDFN distinguishes the rare cell clusters contained in TASIC, we selected the rare cell clusters to make the t-SNE projection and found that the cell clusters were well-separated in the potential space (**Figure 3c**). The relationship between real cell clusters and the number of cells they contain is presented in Supplementary Table S12.

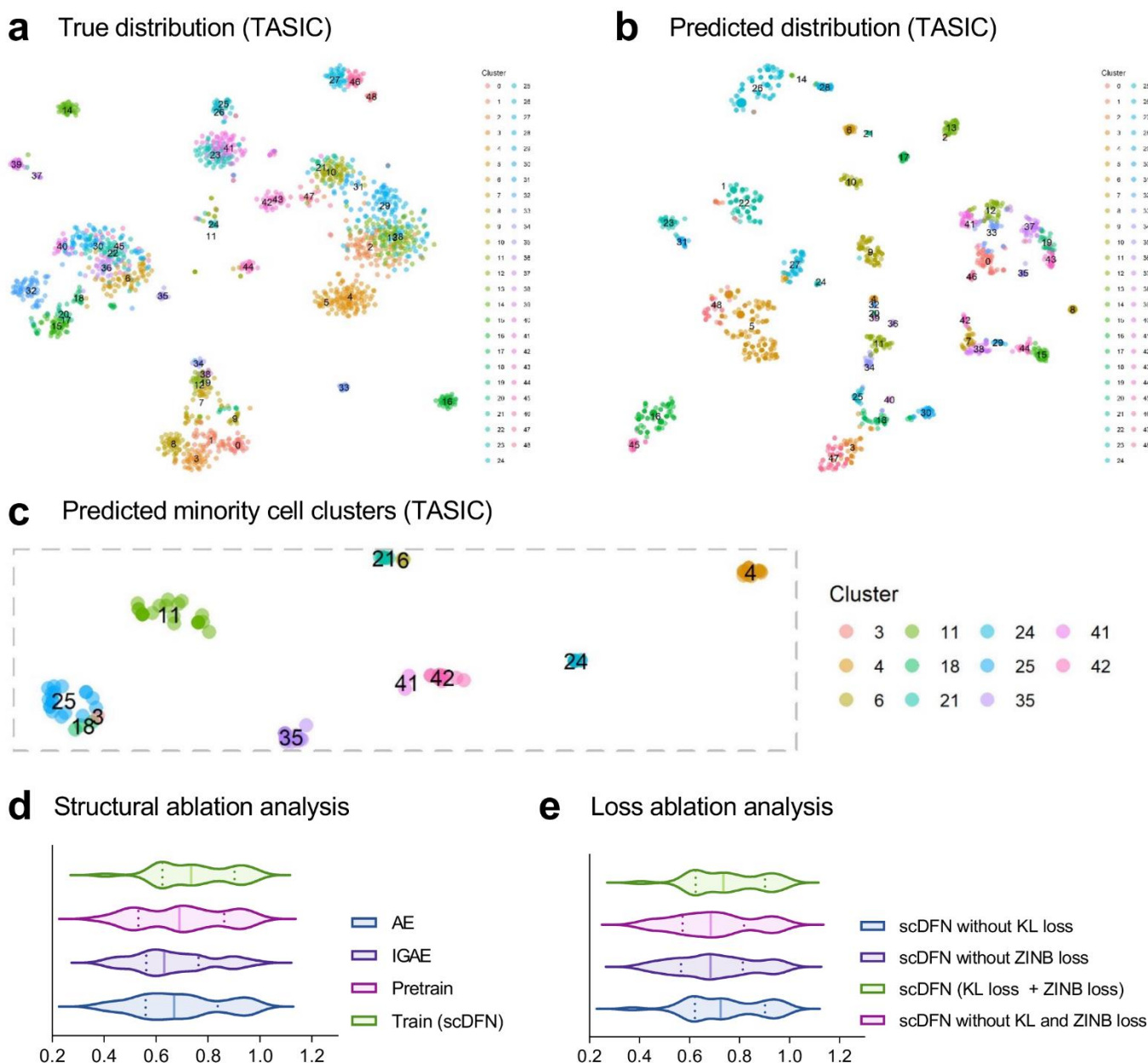


Figure 3. (a-c) The scDFN separated different cell types in the TASIC dataset: (a) Author's noted labels on latent space; (b) The scDFN prediction of 32 cell types; (c) The projection of scDFN on small cell clusters. (d-e) Ablation studies: (d) Comparison of ARI values for the 32 real scRNA-seq datasets with different modules ((i) AE only, (ii) IGAE only, (iii) the pre-training part after the fusion of AE and IGAE, and (iv) the fine-training part); (e) Comparison of ARI values for the 32 real scRNA-seq datasets with different losses.

2.4 Superiority of information fusion

To assess the effectiveness of our proposed model at various stages and the contribution of each unit to the performance of scDFN, we conducted an ablation study containing (i) AE only, (ii) IGAE only, (iii) the pre-training part after the fusion of AE and IGAE, and (iv) the fine-training part. We evaluated the performance of different modules, the distribution of ARI values are shown in **Figure 3d**, and the results of the NMI values and ARI values of different modules are shown in Supplementary Tables S4 and S5, respectively. The pre-training stage achieved an average NMI value of 0.7956 and an average ARI value of 0.6987 on 32 scRNA-seq datasets; these values were better than the average NMI and ARI values (0.7797 and 0.6912, respectively) obtained using only AE. Since we added triple self-supervised loss during the training stage, the average NMI and ARI values were further improved to 0.8161 and 0.7518, respectively. All results obtained indicated that the information fusion module can effectively extract the attribute information and topological information of single-cell data, thus improving the clustering performance.

2.5 Analysis of the influence of different losses on the experimental performance

In our proposed scDFN, the decoder network used four loss functions to learn the latent representation, and thus, it collaboratively optimized the ZINB reconstruction loss, the MSE reconstruction loss of AE, the MSE reconstruction loss of IGAE, and the KL triple self-supervised clustering loss. To examine the reasoning behind adopting ZINB loss and self-supervised triple clustering loss, we compared the performance of scDFN, scDFN without ZINB loss, scDFN without KL loss, and scDFN without both of them on 32 real scRNA-seq data. The distribution of ARI values for different losses is shown in **Figure 3e**, and the detailed results of NMI values and ARI values are shown in Supplementary Tables S6 and S7, respectively. We found that the performance of scDFN without KL and ZINB losses decreased considerably, with an average decrease in NMI of 0.0224 and an average decrease in ARI of 0.0542, whereas the performance of scDFN without KL loss decreased slightly, with an average decrease in NMI of 0.0032 and an average decrease in ARI of 0.0091. Using four types of losses, the NMI value improved on 14 datasets, while the ARI value improved on 16 datasets. Based on these findings, we concluded that ZINB loss and clustering loss in auto-encoders positively affected the clustering performance of scDFN.

2.6 Impact of cell cluster number and highly variable genes on scDFN

To determine whether the choice of the number of cell clusters affects the clustering performance of scDFN, we changed the number of clusters, i.e., we set the number of experimental clusters to $\{k - 2, k - 1, k, k + 1, k + 2\}$, where k indicates the true number of clusters. The NMI results of scDFN on 32 datasets with different clusters are shown in **Figure 4a**. The best NMI and ARI scores were obtained when k was set as the true number of clusters, such as on 10X_PBMC [32], Romanov [33], Human [34], etc. Even when the best NMI value was not obtained under the actual

number of clusters, changing the number of clusters had a negligible effect on the NMI value. For the datasets Wang_Lung ($K=2$), Biase ($K=4$), and Xin ($K=4$), the NMI values were more affected by interference, which occurred probably because the datasets contained few cell clusters. We observed that scDFN was not sensitive to the number of clusters (Figure 4a). By comparing the average values, we found that changing the number of clusters had a greater impact on the ARI value than the NMI value (Supplementary Tables S8 and S9). Overall, changing the number of clusters slightly affected the NMI and ARI values. The effects of different numbers of selected genes (M) on clustering performance are presented in Supplementary Tables S10 and S11. The results showed that scDFN had the best overall clustering performance for most datasets when M was 2,000, compared to other conditions.

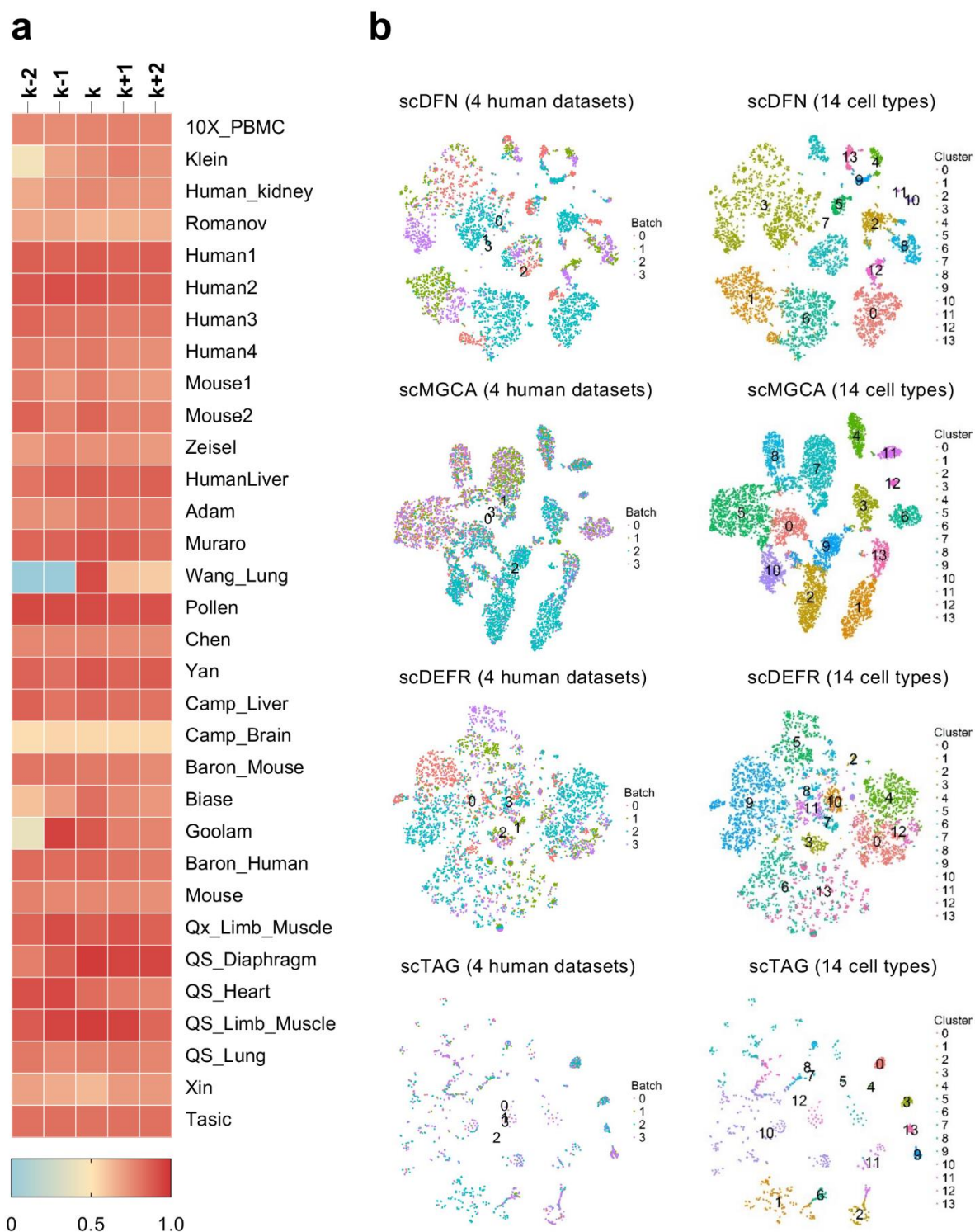


Figure 4. (a) The impact of different numbers of cell clusters on clustering performance (k represents the real number of cell clusters. (b) The t-SNE plots were used to compare the effectiveness of batch effect correction using four methods, including scDFN, scMGCA, scDEFR, and scTAG. The dataset used is a fusion of Human1 (Batch0), Human2 (Batch1), Human3 (Batch2), and Human4 (Batch3).

2.7 scDFN can effectively eliminate the batch effect

Our scDFN method not only exhibited excellent performance in clustering but also showed its ability to effectively eliminate batch effects. Human1, Human2, Human3, and Human4 are single-cell RNA sequencing datasets from four human donor pancreatic cells. The number of cells in these datasets is 1,937, 1,724, 3,605, and 1,303, respectively. The number of genes is 20,215. Each dataset has 14 consistent cell types. These cells were derived from different batches[34]. As the cells were derived from different pancreatic tissues, there may be subtle differences between them related to batch effects. To evaluate the performance of scDFN in eliminating batch effects, we used R to integrate four datasets (Human1, Human2, Human3, and Human4). We obtained a combined dataset containing 8,569 cells and 14 different cell clusters. We evaluated the ability of three state-of-the-art scRNA clustering methods, including scMGCA, scDEFR, and scTAG, to eliminate batch effects, determined by cluster visualization and batch correction results. The experimental results are shown in **Figure 4b**. We found that scTAG could not completely separate some cell types, such as clusters 8, 10, and 12, scDEFR could not separate most cells, and scMGCA could effectively separate cells. Overall, scTAG and scDEFR performed poorly in eliminating batch effects. Both scDFN and scMGCA efficiently eliminated batch effects, demonstrated by clustering a mixture of cells of the same type but obtained from different batches. However, although scMGCA effectively eliminated batch effects, its clustering performance was not as good as that of scDFN because scDFN showed larger distances between clusters. To summarize, scDFN can mix cells of the same type but in different batches and accurately distinguish different cell types.

2.8 Finding marker genes

Single-cell RNA sequencing data are often annotated using marker genes of the cell type, which can be predefined or defined de novo. Successful artificial interpretation requires high expression levels of closed cell clusters with distinct marker genes, and these genes are mainly concentrated in specific cell clusters. Finding marker genes related to cell clusters is a prerequisite for cell annotation, and the selection of marker genes strongly depends on the accuracy of the final results. In this study, we selected the Goolam dataset and used it to find the marker genes of each newborn [35]. For the Goolam dataset, we calculated the ranking of highly different genes in each cluster and only performed logistic regression analysis for validation (see Figure S1). Then, the top two genes from each cluster were selected for visualizing the results. The visualization results of Goolam are shown in **Figure 5**. We used bubble plots and violin plots to display the expression of marker genes in each cell cluster. The marker genes for each cell cluster in the Goolam dataset are shown in **Figure 5**. The results were visualized using the Python package Scanpy.

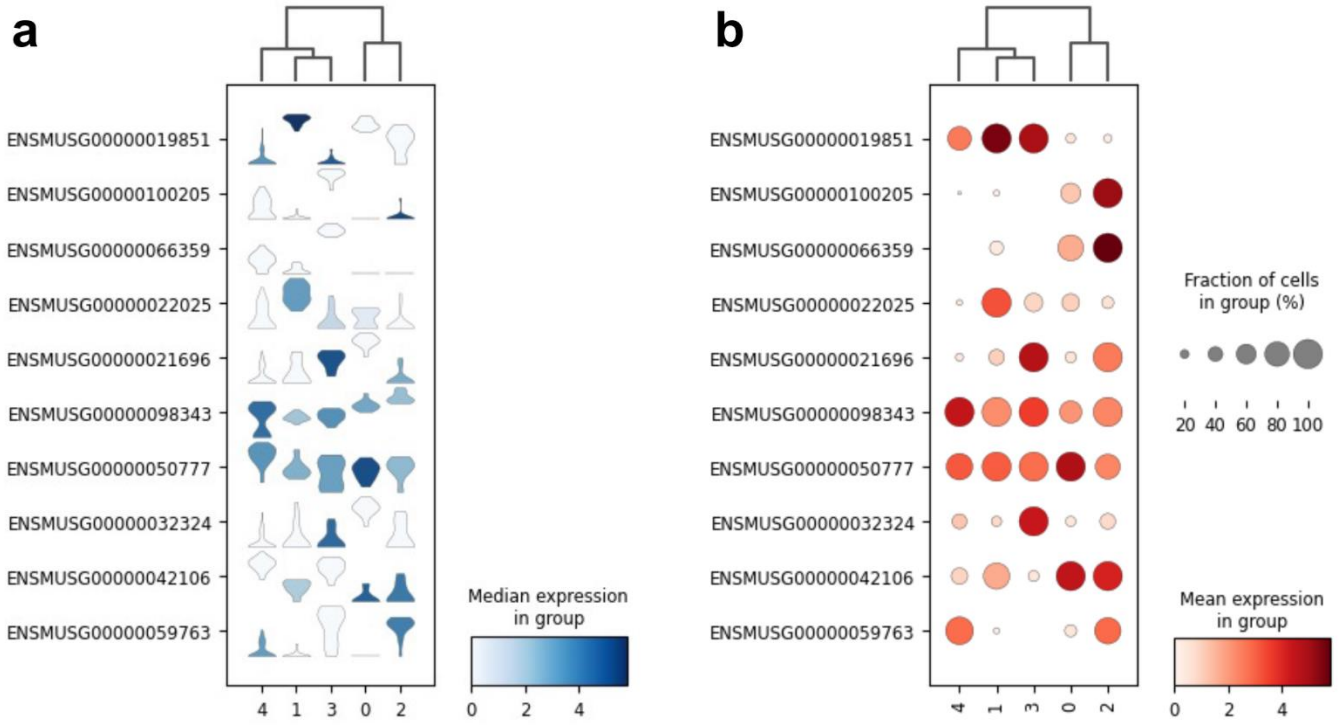


Figure 5. Visualization of the results of the Goolam marker genes. (a) A violin plot and (b) A bubble chart; the abscissa represents marker genes, and the ordinate represents cell clusters. The color depth of the violin plot reflects the degree of expression of the marker genes, while the shape of the violin plot shows the distribution of the expression of the marker genes in different cell types. The distribution of expression of the marker genes was different in different types of cells. The marker genes for each cluster were prominent.

3. Conclusion

In this study, a deep single-cell clustering model was developed through the fusion of AE and IGAE to identify cell populations in scRNA-seq datasets. First, high-dimensional scRNA-seq data were preprocessed, and the top highly variable genes were selected to eliminate redundant genes with low levels of expression that could interfere with the clustering results. These results indicated significant improvements in clustering performance. Next, AE and IGAE were combined through an information fusion mechanism to learn the attribute information and topological information of scRNA-seq data. Then, clustering was optimized using a quaternion loss function to identify cell types from the learned latent cell representations. To validate our model, we conducted a comprehensive study, in which we compared our model to other benchmark methods for cell type identification and characterization from different perspectives. We found that scDFN was better than the currently used methods. With the development of advanced high-throughput technologies for scRNA-seq and emerging cell atlases, we aim to investigate the performance of the proposed scDFN method on larger-scale scRNA-seq data in the future. One reason why scDFN did not consistently outperform other models was that it used hyperparameter settings uniformly across all datasets. However, this approach, in turn, decreased the problem of overfitting, which commonly occurs in methods trained on a specific dataset. This view was supported by the fact that the other five methods had lower performance rankings on average and showed larger differences than

scDFN for the 32 datasets analyzed. In the future, we aim to expand the scope of scDFN and integrate different single-cell omics datasets, such as scATAC-seq and scMethyl-seq, to comprehensively understand cellular heterogeneity and epigenetic regulation.

4. Materials and Methods

4.1 Data collection

We collected 32 real scRNA-seq datasets of mouse and human tissues. The cell types ranged from 2 to 49, and cell numbers ranged from 56 to 12,089 [30, 32, 34-51]. These datasets were obtained from four sequencing platforms, including inDrop, 10X, SMARTer, and Smart-seq2. These datasets had known labels for each cell and had been used to test performance in previous studies. Detailed information on the collected scRNA-seq datasets is summarized in Supplementary Table S1.

4.2 Details regarding model implementation

Our model was implemented using the PyTorch platform. While constructing graph neural networks, we used the KNN algorithm to construct the cell graph, and the number of nearest neighbor k was set to 10, the same as that set in another study [52]. Our model training strategy was as follows. We first trained AE and IGAE independently and achieved potential embedding $\tilde{Z}$. Then, we integrated these two modules into a unified framework for pre-training to quickly obtain the optimal combination of hyperparameters. Finally, the triple self-supervised strategy was used to fine-tune the hyperparameters and obtain clustering results. Specifically, we used the classical symmetric encoder structure in AE and set the dimension reduction dimension to M -1024-512-128-20, the learning rate to 1e-3, and the epoch to 50 in the encoder part. We used a symmetrical graph encoder structure in IGAE and set the dimension reduction dimension to M -1024-512-128-20, the learning rate to 1e-7, the fusion parameter to 0.1, and the epoch to 50.

The module of attribute and topology information fusion consisted of pre-training and fine-training processes, whose learning rate of pre-training was 1e-5, fine-training was 1e-8, epoch of pre-training was 100, and fine-training was 50, respectively. The weight vector $\{\gamma_1, \gamma_2, \gamma_3, \gamma_4\}$ of the objective function was set to $\{0.3, 0.3, 3, 0.2\}$. For the parameters, the Adam optimizer was used uniformly, while learning rate attenuation was fixed. We did not set the hyperparameters for each dataset separately; instead, all test datasets used the same hyperparameters to assess the robustness of scDFN in the following sections.

4.3 Data processing module

The scDFN model uses the original gene expression matrix $X \in R^{N \times d}$ as input, in which N represents the number of cells and d represents the number of genes. The raw gene expression matrix was processed by the Scanpy package. We

removed genes with non-zero counts in less than three cells and calculated the size factor of each cell, which is the median of the ratio of the gene expression to the geometric mean of the gene and is standardized by the cell library; thus, the total count was the same between cells. If the library size (the number of total read counts) of cell i is represented as s_i , the cell size factor can be represented as $s_i/\text{median}(s)$. Since the single-cell expression matrix represents the information on gene expression levels in each cell transcriptome, all elements in the matrix have non-negative values. Therefore, we performed a logarithmic transformation to eliminate the influence of the differences in the value range of different interval data. Finally, to discard genes with low recognition and description information, the function in the Scanpy package was used to select the first M ($M = 2000$) highly variable genes. After standardizing and compressing the data, we obtained the gene expression matrix $X' \in R^{N \times d'}$, $d' < d$.

Given an undirected graph $G = \{V, E\}$, $V = \{v_1, v_2, \dots, v_N\}$ represents the set of nodes, E represents the set of edges, and N represents the number of cells. The original adjacency matrix of G is defined by:

$$A = (a_{ij})_{N \times N} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{else} \end{cases} \quad (1)$$

The degree matrix is represented by D , where $D = \text{diag}(d_1, d_2, \dots, d_N) \in R^{N \times N}$, and $d_i = \sum_{v_j \in V} a_{ij}$. Then, A can be normalized as follows:

$$\tilde{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}} \quad (2)$$

Here, $I \in R^{N \times N}$ represents an identity matrix. Thus, the gene expression matrix X' and the standard adjacency matrix $\tilde{A}$ of the cell graph are the input of IGAE.

4.4 Autoencoder

As learning effective cell representation is a critical step for clustering scRNA-seq data [53], we used an autoencoder to learn the representation of scRNA-seq data. We assumed that the autoencoder has l layers, in which l represents the layer number. Then, the l -th layer of a specific encoder process can be represented as follows:

$$H^{(l)} = \sigma(W_e^{(l)}H^{(l-1)} + b_e^l) \quad (3)$$

Here, σ represents a linear activation function, such as the Relu and Sigmoid function, $W_e^{(l)}$ and b_e^l represent the weight and bias, respectively, and the initial $H^{(0)}$ represents the gene expression matrix $H^{(0)}$.

The function of the decoder is to pass the data from encoding to the output layer and restore the extracted information to its original structure. The structure of encoding and decoding is symmetrical. We used the decoder to reconstruct the scRNA-seq data, which can be expressed as follows:

$$H^{(l)} = \sigma \left(W_d^{(l)} H^{(l-1)} + b_d^l \right) \quad (4)$$

Here, $W_d^{(l)}$ and b_d^l represent the weight and bias of the l -th layer. The output of the decoder is the reconstruction of the gene expression matrix, which is optimized by the MSE loss function:

$$L_{AE} = \frac{1}{2N} \sum_{i=1}^N \|x_i - \hat{x}_i\|_2^2 = \frac{1}{2N} \sum_{i=1}^N \|X' - \hat{X}\|_2^2 \quad (5)$$

4.5 Improved Graph Autoencoder

The classical autoencoder is usually symmetric, but the graph convolutional network is usually asymmetric. A general solution is to reconstruct the adjacency matrix; however, it ignores the structure-based attribute information. To compensate for the deficiency, we used a novel symmetric improved graph autoencoder (IGAE) to enhance stability by simultaneously reconstructing the attribute matrix and the adjacency matrix. In IGAE, the encoder and decoder were modified as follows:

$$\begin{aligned} Z^{(l)} &= \sigma \left(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}Z^{(l-1)}W^{(l)} \right) \\ \hat{Z}^{(h)} &= \sigma \left(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}\hat{Z}^{(h-1)}W^{(h)} \right) \end{aligned} \quad (6)$$

Here, $W^{(l)}$ and $W^{(h)}$ represent the parameters that can be learned by the l -th coding layer and the h -th decoding layer, respectively, and σ represents a linear activation function. For satisfactory cell representation, the mixed loss function was defined by the following formula:

$$L_{IGAE} = L_w + \gamma L_a \quad (7)$$

Here, γ represents a hyperparameter that needs to be predefined to balance the weights of the two losses. In this study, we set $\gamma = 0.1$. The loss L_w and L_a can be expressed as follows:

$$L_w = \frac{1}{2N} \| \tilde{A}X - \hat{Z} \|_F^2 \quad (8)$$

$$L_a = \frac{1}{2N} \| \tilde{A} - \hat{A} \|_F^2 \quad (9)$$

Here, $\hat{Z}$ represents the reconstructed weighted attribute matrix, and $\hat{A}$ represents the reconstructed adjacency matrix. The goal of the network is to minimize the reconstruction loss of the weighted attribute matrix and the adjacency matrix simultaneously.

4.6 Attribute and Topology Information Fusion (ATIF)

The information fusion module was used to fully mine the feature attribute and topology information of the gene expression matrix, and an information fusion module of attribute information and topology information was proposed. The module consisted of two parts, including the cross-network information fusion mechanism and the triple self-supervision strategy. The overall structure of ATIF is shown in **Figure 1b**.

First, the potential embedding $Z_{AE} \in R^{N \times d'}$ obtained by the AE and the potential embedding $Z_{IGAE} \in R^{N \times d'}$ obtained by the IGAE underwent initial information fusion; in the expressions, d' represents the dimension of the potential embedded gene. The above-mentioned terms were combined as follows:

$$Z_I = \alpha Z_{AE} + (1 - \alpha) Z_{IGAE} \quad (10)$$

Here, α represents a learnable parameter. The value of α was initialized to 0.5, and then, automatically adjusted by the gradient descent method. Using a message-passing operation, we obtained $Z_L = \tilde{A} Z_I$, which was equivalent to a local data augmentation of the initial fused information.

Based on local data enhancement, we used an autocorrelation method to investigate the non-local relationship between cells through the autocorrelation matrix. Specifically, we first calculated the normalized autocorrelation matrix S , and the element S_{ij} was defined as follows:

$$S_{ij} = \frac{\exp((Z_s)_{ij})}{\sum_{k=1}^N \exp((Z_s)_{ik})} \quad (11)$$

where $Z_s = Z_L Z_L^T$.

Based on the autocorrelation matrix, we used Z_L to consider the global connection between cells and found $Z_G = S Z_L$. Finally, we linked the global relationship and the local relationship matrices to ensure the smooth transmission of the initial fusion information in the information fusion module $\tilde{Z} = \beta Z_G + Z_L$. In the equation, β is a learnable parameter. The information fusion mechanism can be used to extract the global information and local information of cells from the aspects of local correlation and global correlation. The attribute information obtained from AE and the topological relationship of cells obtained from IGAE can achieve a better potential cell representation than that achieved by using either one alone.

4.7 Decoder based on ZINB

To capture the structure of the scRNA-seq data, we employed ZINB loss to the decoder. Specifically, ZINB is used to model the dropout event of scRNA-seq data based on zero components and the distribution of NB [21]. This method connects three independent connection layers with the last layer, and then, estimates the parameters of ZINB, including the dropout rate π , dispersion degree θ , and mean μ . The output of the network is as follows:

$$\begin{aligned} \Pi &= \text{sigmoid}(W_\pi X) \\ M &= \exp(W_\mu X) \\ \Theta &= \exp(W_\theta X) \end{aligned} \quad (12)$$

Here, W represents the learnable weight of the loss function, and uses a ZINB-based decoder to reconstruct scRNA-seq data; NB and ZINB can be expressed as follows:

$$NB(X | \mu, \theta) = \frac{\Gamma(X+\theta)}{X!\Gamma(\theta)} \left(\frac{\theta}{\theta+\mu}\right)^\theta \left(\frac{\mu}{\theta+\mu}\right)^X \quad (13)$$

$$ZINB(X | \pi, \mu, \theta) = \pi\delta_0(X) + (1 - \pi)NB(X) \quad (14)$$

The reconstruction loss function of the gene expression matrix X is defined as the negative log-likelihood of the ZINB distribution and is expressed as follows:

$$L_{ZINB} = -\log(ZINB(X | \pi, \mu, \sigma)) \quad (15)$$

4.8 Triplet Self-supervised Strategy

We integrated the autoencoder and the improved graph autoencoder in the neural network architecture, the combination of which helped us extract the attribute information and topology information of the matrix. We proposed a triple self-supervised module, which combined the autoencoder and the improved graph encoder. We trained these two modules to achieve end-to-end single-cell clustering. In the first step, to generate a more reliable clustering network for guiding training, we used a more stable clustering embedding $\tilde{Z} \in R^{N \times d'}$. For the i -th cell and the j -th cluster, $\tilde{Z}$ combined the information from AE and IGAE to generate the target distribution. We used t-distribution [54] as the kernel to determine the similarity between the data representation $\tilde{z}_i$ and the cluster center vector u_j . The process can be expressed as follows:

$$q_{ij} = \frac{(1 + \|\tilde{z}_i - \mu_j\|/\nu)^{-\frac{\nu+1}{2}}}{\sum_j (1 + \|\tilde{z}_i - \mu_j\|/\nu)^{-\frac{\nu+1}{2}}} \quad (16)$$

Here, ν represents the degree of freedom of the Student's distribution, q_{ij} represents the probability of the i -th node being assigned to the j -th center (i.e., a soft assignment), and $Q = [q_{ij}]$ represents the distribution of all samples. In the second step, we optimized data representation by learning from high-confidence assignments. Specifically, we made the data representation closer to the cluster center, thereby improving the cohesion of the cluster. Thus, we introduced the target distribution P , which was calculated as follows:

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \quad (17)$$

Here, $0 \leq p_{ij} \leq 1$ represents the element that generates the target distribution $P \in R^{N \times K}$, which represents the probability of the i -th sample belonging to the j -th center. Each assignment in Q was normalized by the sum of squares, so that the assignment had a higher confidence level, and the following objective function was obtained:

$$L_{clu} = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (18)$$

Similar to the generated target distribution Q , we used equation (17) to calculate the soft allocation distribution of AE and IGAE. On the potential embedding of the two sub-networks, we expressed the soft allocation distribution of IGAE and AE as $Q' = [q'_{ij}]$ and $Q'' = [q''_{ij}]$, respectively.

To train the network under a unified standard and improve the representation ability of each component, we designed a triple clustering loss using the following formula:

$$L_{KL} = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{\frac{(q_{ij} + q'_{ij} + q''_{ij})}{3}} \quad (19)$$

Through this loss, we aligned AE, IGAE, and their fusion distribution with the target distribution. The objective function had two advantages. (1) Compared to the traditional multi-classification loss function, the KL divergence updates the entire model in a more reasonable way to avoid serious interference with the data representation. (2) The AE module, IGAE module, and fusion representation are unified on the same optimization objective, causing their results to be consistent during the training process. Since the target of the AE module, IGAE module, and fusion representation is the approximate target distribution P , these three modules are strongly connected, giving rise to the triple self-monitoring mechanism.

4.9 Quadruple joint losses

The learning goal of the model has four main parts, including the reconstruction loss of AE, the double reconstruction loss of IGAE, the clustering loss of the triple self-monitoring mechanism, and the ZINB loss. These four losses are combined to jointly optimize the clustering representation of cells. The joint clustering loss is expressed as follows:

$$L = \gamma_1 L_{AE} + \gamma_2 L_{IGAE} + \gamma_3 L_{KL} + \gamma_4 L_{ZINB} \quad (20)$$

Here, $\gamma_1, \gamma_2, \gamma_3$ represent predefined hyperparameters that balance clustering optimization and reconstruction losses, and γ_4 represents a hyperparameter that controls the data structure that captures the distribution of the scRNA-seq data. Various losses play certain roles in optimizing single-cell clustering representation (explained in the subsequent ablation analysis).

4.10 Performance evaluation indicators

To evaluate the results of each clustering method, we selected two evaluation metrics for commonly used clustering. Based on the clustering results of the model, the Normalized Mutual Information (NMI) [55] and the Adjusted Rand Index (ARI) [56] can be used to evaluate the advantages and disadvantages of the model. NMI is calculated as follows:

$$NMI = \frac{I(\Omega; C)}{H(t) + H(d)} \quad (21)$$

$$I(\Omega; C) = \sum_i \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} \quad (22)$$

Here, $H(t)$ and $H(d)$ represent the entropy of the true label and the detected cluster, respectively; I represents mutual information and H represents entropy. A more detailed description of the calculation of NMI is provided in another study [57].

The ARI is another commonly used measure to evaluate the similarity between real labels and detected clusters. Given n cells with s true clusters $C_t = [C_{t1}, C_{t2}, \dots, C_{ts}]$ and r detected clusters $C_d = [C_{d1}, C_{d2}, \dots, C_{dr}]$. The overlap between the two partitions can be summarized as a contingency table, where each entry $n_{ij} = [C_{ti} \cap C_{dj}]$ represents the number of elements shared between the two clusters C_{ti} and C_{dj} . Based on the above-mentioned information, ARI can be defined as follows:

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \frac{[\sum_i \binom{a_i}{2}] \sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2} [\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - \frac{[\sum_i \binom{a_i}{2}] \sum_j \binom{b_j}{2}}{\binom{n}{2}}} \quad (23)$$

Here, a_i represents the sum of the row i of the contingency table, b_j represents the sum of the row j of the contingency table, and $\binom{n}{2}$ represents the binomial coefficient.

Funding

This work is supported by the National Natural Science Foundation of China (No. 62071079 and 62202388), Hainan Normal University, Ministry of Education (JSKX202203), the National Key Research and Development Program of China (No. 2022YFF1000100), the Qin Chuangyuan Innovation and Entrepreneurship Talent Project (No. QCYRCXM-2022-230), and the Talent Research Funding at Northwest A&F University (No. Z1090222021).

References

1. Regev A, Teichmann SA, Lander ES, Amit I, Benoist C, Birney E, Bodenmiller B, Campbell P, Carninci P, Clatworthy M, et al: **The Human Cell Atlas**. *Elife* 2017, **6**.
2. Ben-Dor A, Shamir R, Yakhini Z: **Clustering gene expression patterns**. *J Comput Biol* 1999, **6**:281-297.
3. Zheng R, Liang Z, Chen X, Tian Y, Cao C, Li M: **An Adaptive Sparse Subspace Clustering for Cell Type Identification**. *Front Genet* 2020, **11**:407.
4. Papalexi E, Satija R: **Single-cell RNA sequencing to explore immune cell heterogeneity**. *Nat Rev Immunol* 2018, **18**:35-45.

5. Petegrosso R, Li Z, Kuang R: **Machine learning and statistical methods for clustering single-cell RNA-sequencing data.** *Brief Bioinform* 2020, **21**:1209-1223.
6. Johnson SC: **Hierarchical clustering schemes.** *Psychometrika* 1967, **32**:241-254.
7. Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, Natarajan KN, Reik W, Barahona M, Green AR, Hemberg M: **SC3: consensus clustering of single-cell RNA-seq data.** *Nat Methods* 2017, **14**:483-486.
8. Stassen SV, Siu DMD, Lee KCM, Ho JWK, So HKH, Tsia KK: **PARC: ultrafast and accurate clustering of phenotypic data of millions of single cells.** *Bioinformatics* 2020, **36**:2778-2786.
9. Satija R, Farrell JA, Gennert D, Schier AF, Regev A: **Spatial reconstruction of single-cell gene expression data.** *Nat Biotechnol* 2015, **33**:495-502.
10. Ji Z, Ji H: **TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis.** *Nucleic Acids Res* 2016, **44**:e117.
11. Lee J, Kim S, Hyun D, Lee N, Kim Y, Park C: **Deep single-cell RNA-seq data clustering with graph prototypical contrastive learning.** *Bioinformatics* 2023, **39**.
12. Gan Y, Chen Y, Xu G, Guo W, Zou G: **Deep enhanced constraint clustering based on contrastive learning for scRNA-seq data.** *Brief Bioinform* 2023, **24**.
13. Wang Y, Yu Z, Li S, Bian C, Liang Y, Wong KC, Li X: **scBGEDA: deep single-cell clustering analysis via a dual denoising autoencoder with bipartite graph ensemble clustering.** *Bioinformatics* 2023, **39**.
14. Zhao JP, Hou TS, Su Y, Zheng CH: **scSSA: A clustering method for single cell RNA-seq data based on semi-supervised autoencoder.** *Methods* 2022, **208**:66-74.
15. Wei N, Nie Y, Liu L, Zheng X, Wu HJ: **Secuer: Ultrafast, scalable and accurate clustering of single-cell RNA-seq data.** *PLoS Comput Biol* 2022, **18**:e1010753.
16. Wang H, Zhao J, Zheng C, Su Y: **scDSSC: Deep Sparse Subspace Clustering for scRNA-seq Data.** *PLoS Comput Biol* 2022, **18**:e1010772.
17. Chen L, Wang W, Zhai Y, Deng M: **Deep soft K-means clustering with self-training for single-cell RNA sequence data.** *NAR Genom Bioinform* 2020, **2**:lqaa039.
18. Tian T, Wan J, Song Q, Wei Z: **Clustering single-cell RNA-seq data with a model-based deep learning approach.** *Nature Machine Intelligence* 2019, **1**:191-198.
19. Li X, Wang K, Lyu Y, Pan H, Zhang J, Stambolian D, Susztak K, Reilly MP, Hu G, Li M: **Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis.** *Nat Commun* 2020, **11**:2338.
20. Coskun M: *Graph Convolutional Networks Meet with High Dimensionality Reduction.* 2019.
21. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ: **Single-cell RNA-seq denoising using a deep count autoencoder.** *Nat Commun* 2019, **10**:390.
22. Yu Z, Lu Y, Wang Y, Tang F, Wong K-C, Li X: **ZINB-Based Graph Embedding Autoencoder for Single-Cell RNA-Seq Interpretations.** *Proceedings of the AAAI Conference on Artificial Intelligence* 2022, **36**:4671-4679.
23. Wang J, Ma A, Chang Y, Gong J, Jiang Y, Qi R, Wang C, Fu H, Ma Q, Xu D: **scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses.** *Nature Communications* 2021, **12**:1882.
24. Luo Z, Xu C, Zhang Z, Jin W: **A topology-preserving dimensionality reduction method for single-cell RNA-seq data using graph autoencoder.** *Sci Rep* 2021, **11**:20028.
25. Yu Z, Su Y, Lu Y, Yang Y, Wang F, Zhang S, Chang Y, Wong KC, Li X: **Topological identification and interpretation for single-cell gene regulation elucidation across multiple platforms using scMGCA.** *Nat Commun* 2023, **14**:400.
26. Ding JA-O, Liu R, Wen H, Tang W, Li Z, Venegas J, Su R, Molho D, Jin W, Wang Y, et al: **DANCE: a deep learning library and benchmark platform for single-cell analysis.**
27. Bo D, Wang X, Shi C, Zhu M, Lu E, Cui P: **Structural Deep Clustering Network.** In *Proceedings of The Web Conference 2020.* pp. 1400-1410. Taipei, Taiwan: Association for Computing Machinery; 2020:1400-1410.
28. Tu W, Zhou S, Liu X, Guo X, Cai Z, Zhu E, Cheng J: **Deep Fusion Clustering Network.** *Proceedings of the AAAI Conference on Artificial Intelligence* 2021, **35**:9978-9987.
29. Wolf FA, Angerer P, Theis FJ: **SCANPY: large-scale single-cell gene expression data analysis.** *Genome Biol* 2018,

19:15.

30. Schaum N, Karkanias J, Neff NF, May AP, Quake SR, Wyss-Coray T, Darmanis S, Batson J, Botvinnik O, Chen MB, et al: **Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris.** *Nature* 2018, **562**:367-372.
31. Lei T, Chen R, Zhang S, Chen Y: **Self-supervised deep clustering of single-cell RNA-seq data to hierarchically detect rare cell populations.** *Briefings in Bioinformatics* 2023, **24**.
32. Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, Ziraldo SB, Wheeler TD, McDermott GP, Zhu J, et al: **Massively parallel digital transcriptional profiling of single cells.** *Nat Commun* 2017, **8**:14049.
33. Romanov RA, Zeisel A, Bakker J, Girach F, Hellysaz A, Tomer R, Alpár A, Mulder J, Clotman F, Keimpema E, et al: **Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes.** *Nat Neurosci* 2017, **20**:176-188.
34. Baron M, Veres A, Wolock SL, Faust AL, Gaujoux R, Vetere A, Ryu JH, Wagner BK, Shen-Orr SS, Klein AM, et al: **A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure.** *Cell Syst* 2016, **3**:346-360.e344.
35. Goolam M, Scialdone A, Graham SJL, Macaulay IC, Jedrusik A, Hupalowska A, Voet T, Marioni JC, Zernicka-Goetz M: **Heterogeneity in Oct4 and Sox2 Targets Biases Cell Fate in 4-Cell Mouse Embryos.** *Cell* 2016, **165**:61-74.
36. Adam M, Potter AS, Potter SS: **Psychrophilic proteases dramatically reduce single-cell RNA-seq artifacts: a molecular atlas of kidney development.** *Development* 2017, **144**:3625-3632.
37. Biase FH, Cao X, Zhong S: **Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing.** *Genome Res* 2014, **24**:1787-1796.
38. Camp JG, Badsha F, Florio M, Kanton S, Gerber T, Wilsch-Brauninger M, Lewitus E, Sykes A, Hevers W, Lancaster M, et al: **Human cerebral organoids recapitulate gene expression programs of fetal neocortex development.** *Proc Natl Acad Sci U S A* 2015, **112**:15672-15677.
39. Chen R, Wu X, Jiang L, Zhang Y: **Single-Cell RNA-Seq Reveals Hypothalamic Cell Diversity.** *Cell Rep* 2017, **18**:3227-3241.
40. Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, Saadatpour A, Zhou Z, Chen H, Ye F, et al: **Mapping the Mouse Cell Atlas by Microwell-Seq.** *Cell* 2018, **172**:1091-1107.e1017.
41. Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, Peshkin L, Weitz DA, Kirschner MW: **Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells.** *Cell* 2015, **161**:1187-1201.
42. Muraro MJ, Dharmadhikari G, Grün D, Groen N, Dielen T, Jansen E, van Gurp L, Engelse MA, Carlotti F, de Koning EJ, van Oudenaarden A: **A Single-Cell Transcriptome Atlas of the Human Pancreas.** *Cell Syst* 2016, **3**:385-394.e383.
43. Pollen AA, Nowakowski TJ, Shuga J, Wang X, Leyrat AA, Lui JH, Li N, Szpankowski L, Fowler B, Chen P, et al: **Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex.** *Nat Biotechnol* 2014, **32**:1053-1058.
44. Romanov RA, Zeisel A, Bakker J, Girach F, Hellysaz A, Tomer R, Alpar A, Mulder J, Clotman F, Keimpema E, et al: **Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes.** *Nat Neurosci* 2017, **20**:176-188.
45. Tasic B, Menon V, Nguyen TN, Kim TK, Jarsky T, Yao Z, Levi B, Gray LT, Sorensen SA, Dolbeare T, et al: **Adult mouse cortical cell taxonomy revealed by single cell transcriptomics.** *Nat Neurosci* 2016, **19**:335-346.
46. Tian T, Zhang J, Lin X, Wei Z, Hakonarson H: **Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data.** *Nat Commun* 2021, **12**:1873.
47. Wang Y, Tang Z, Huang H, Li J, Wang Z, Yu Y, Zhang C, Li J, Dai H, Wang F, et al: **Pulmonary alveolar type I cell population consists of two distinct subtypes that differ in cell fate.** *Proc Natl Acad Sci U S A* 2018, **115**:2407-2412.
48. Xin Y, Kim J, Okamoto H, Ni M, Wei Y, Adler C, Murphy AJ, Yancopoulos GD, Lin C, Gromada J: **RNA Sequencing of Single Human Islet Cells Reveals Type 2 Diabetes Genes.** *Cell Metab* 2016, **24**:608-615.
49. Yan L, Yang M, Guo H, Yang L, Wu J, Li R, Liu P, Lian Y, Zheng X, Yan J, et al: **Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells.** *Nat Struct Mol Biol* 2013, **20**:1131-1139.

50. Young MD, Mitchell TJ, Vieira Braga FA, Tran MGB, Stewart BJ, Ferdinand JR, Collord G, Botting RA, Popescu DM, Loudon KW, et al: **Single-cell transcriptomes from human kidneys reveal the cellular identity of renal tumors.** *Science* 2018, **361**:594-599.
51. Zeisel A, Munoz-Manchado AB, Codeluppi S, Lonnerberg P, La Manno G, Jureus A, Marques S, Munguba H, He L, Betsholtz C, et al: **Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq.** *Science* 2015, **347**:1138-1142.
52. Cheng Y, Su Y, Yu Z, Liang Y, Wong K-C, Li X: **Unsupervised Deep Embedded Fusion Representation of Single-Cell Transcriptomics.** *Proceedings of the AAAI Conference on Artificial Intelligence* 2023, **37**:5036-5044.
53. Dou J, Jia M, Zaslavsky N, Ebeid M, Bao R, Zhang S, Ni K, Liang P, Mao H, Mao Z-H: *Learning More Effective Cell Representations Efficiently.* 2022.
54. Van der Maaten L, Hinton G: **Visualizing data using t-SNE.** *Journal of machine learning research* 2008, **9**.
55. Vinh NX, Epps J, Bailey J: **Information theoretic measures for clusterings comparison: is a correction for chance necessary?** In *Proceedings of the 26th Annual International Conference on Machine Learning.* pp. 1073-1080. Montreal, Quebec, Canada: Association for Computing Machinery; 2009:1073-1080.
56. Yu B, Chen C, Huan Y, Zheng R, Lawrence P, Wang X, Ma A, Gu H: **scGMAI: a Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder.** *Briefings in Bioinformatics* 2021, **22**:1-10.
57. Hu J, Li X, Hu G, Lyu Y, Susztak K, Li M: **Iterative transfer learning with neural network for clustering and cell type classification in single-cell RNA-seq analysis.** *Nat Mach Intell* 2020, **2**:607-618.