BMC Genomics

RESEARCH

Open Access

DGM: deep graph clustering with mincut for analysis of single-cell transcriptomics

Xi Liu[1], Xiaolin Chen[1], Wenqian Yang[1] and Yong Yu[1*]

Abstract

Objective Single-cell RNA sequencing (scRNA-seq) is pivotal for deciphering cellular heterogeneity in biomedical research, from developmental biology to cancer studies. However, accurate cell type identification via clustering remains challenging. Current graph-based methods are limited by ignoring higher-order topologies.

Methods We present Deep Graph MinCut (DGM), an end-to-end deep learning framework designed to overcome these limitations. Its core innovation is the joint optimization of three objectives: (1) a higher-order MinCut loss for capturing complex cellular communities; (2) an inter-cluster orthogonality loss to enforce clear separation between cell populations; and (3) a reconstruction constraint to maintain global graph structure.The model jointly optimizes autoencoder denoising, graph learning, and cluster assignment.

Results Evaluated on ten public scRNA-seq datasets, DGM achieved average improvements of 2.03% in NMI and 3.11% in ARI, outperforming baseline methods. Crucially, downstream biological analysis demonstrated alignment with known marker genes and developmental trajectories, verifying its ability to produce biologically meaningful and interpretable clusters.

Conclusion DGM provides a robust and biologically interpretable framework for cell type identification, with potential for applications in cancer research and developmental biology. By addressing key computational challenges, DGM generates reliable outcomes that may contribute to downstream research, such as biomarker discovery or characterization of tumor microenvironment heterogeneity.

Keyword Single-cell RNA-seq, Graph neural networks, MinCut, Clustering, Cell type identification

Introduction

Single-cell RNA sequencing (scRNA-seq) technology has become a core tool for analyzing cellular heterogeneity, bringing significant progress to developmental biology and the study of cancer studies and other disease mechanisms [1–3]. The precise identification of distinct cell types through clustering is the critical first step in transforming this complex data into actionable biological insights. This technology is particularly crucial for identifying novel cell types, understanding tumor microenvironments in cancer research, and guiding personalized medicine approaches. However, the inherent high dimensionality, sparsity, and technical noise of scRNA-seq data make its precise clustering—that is, accurately identifying different cell types or states—always face severe challenges [4–7], which also complicate downstream analyses such as the accurate identification of differentially expressed genes [8].

Early methods (such as PCA combined with k-means [9, 10]) assume that the data is located on a linear

*Correspondence:
Yong Yu
yongyu@hbmu.edu.cn
[1]HealthCare BigData Center, School of Public Health, Hubei University of Medicine, Shiyan, Hubei, China

manifold and the clusters have a spherical distribution, which is difficult to capture complex nonlinear relationships and sensitive to high noise and dropout events [11, 12].Subsequently, deep autoencoder-based methods, including those built on autoencoder [13, 14] or jointly optimizing deep learning with k-means objectives [15], improve the robustness by nonlinear dimensionality reduction and denoising, but they usually ignore the inter-relationship between cells [16], and the learned high-dimensional embedding may still collapse. In recent years, Graph neural networks (GNNS) have shown great potential in this field by explicitly modeling the relationships between cells [17, 18]. Among them, the differentiable MinCut method [18], which draws on the idea of spectral clustering, has achieved remarkable results by optimizing the goal of "compactness within clusters and alienation between clusters". However, the existing GNN-based clustering methods have three common defects that have not been solved by the system: (1) they generally rely on first-order adjacency relationships [17, 18], and cannot capture higher-order topological structures such as triangle motifs, which are crucial for defining stable cell communities [19]; (2) The column Spaces of the soft assignment matrix output by the model are easy to overlap, which makes the cluster boundaries fuzzy and the embedding vectors of different cell types difficult to distinguish; (3) When optimizing the MinCut objective, excessive pursuit of the minimum cut value is easy to cause over-cut, destroy the global topology of the graph, and cause the biologically coherent cell population to be split. These problems not only make the clustering worse but also make the results harder for biologists to understand. For example, missing complex cell connections can hide true cell groups, and unclear group boundaries can cause rare cell types to be put into the wrong groups. This ultimately leads to wrong conclusions in the biological research that follows.

It is worth noting that even the latest researches (e.g. DESC [14] and scBGEDA [20]) still focus on the optimization of first-order graph convolution and fail to solve the above fundamental problem. Recent studies have attempted to improve the performance by graph contrastive learning or knowledge injection. However, these works still have not systematically proposed solutions to these three defects due to the inherent mechanism of graph clustering, which provides a clear innovation space for our research.

To tackle these three interrelated challenges simultaneously, we propose the Deep Graph MinCut (DGM) model. DGM is a deep learning framework designed to enhance the accuracy and interpretability of single-cell clustering.

Its core innovation is to collaboratively optimize three loss functions: differentiable high-order MinCut loss (which is the first to explicitly model high-order topology by introducing the admittance of high-order structures such as triangle motifs into the clustering loss in a differentiable form); Inter-cluster orthogonality loss (explicitly separating the distance between clusters by enforcing the column orthogonality of the soft assignment matrix to solve the problem of fuzzy cluster boundaries) And topology reconstruction constraints (regularizing the clustering process through the adjacency matrix reconstruction loss to prevent over-segmentation and maintain the global topological consistency of the graph). The overall architecture of the proposed framework is shown in Fig. 1, which collaboratively optimizes autoencoder noise reduction, graph structure learning, and cluster assignment.

Our contributions are three-fold: (1) We introduce a differentiable higher-order MinCut loss, enabling the capture of complex cellular communities that are crucial for defining robust cell types; (2) We enforce inter-cluster orthogonality, ensuring clear separation between cell populations to prevent misclassification and aid in the discovery of rare subsets; (3) We regularize the cut objective with topology reconstruction, preserving the global tissue architecture and preventing biologically implausible over-cut. Collectively, these innovations make DGM a effective tool for generating biologically trustworthy clustering results.

The following chapters of this paper are arranged as follows: The second part details the methodology of DGM; Sect. 3 shows the experimental results and analysis. Section 4 provides discussion and conclusion. Overview of our approach is shown in Fig. 1 .

Methods

The overview our approach is shown in Fig. 1, which consists of autoencoder, GNN encoder, end-to-end clustering assignment network, joint loss function, etc. In the following sections, we will introduce each part of the model in detail. The overall pipeline begins with data processing of the raw scRNA-seq data. Specifically, the expression profile matrix X undergoes normalization to adjust for technical variations and gene filtering to select informative genes, resulting in a processed matrix X. This processed data is then fed into an autoencoder for denoising and dimensionality reduction, producing a low-dimensional representation Z. Subsequently, the representation Z is used to construct a cell-cell graph with adjacency matrix A. Graph Neural Network (GNN) encodes this graph into node embeddings H, which are subsequently fed into a clustering assignment network to produce the soft assignment matrix S. Finally, the model parameters are optimized by jointly minimizing a total loss function that combines reconstruction, clustering, and topology-preserving objectives.

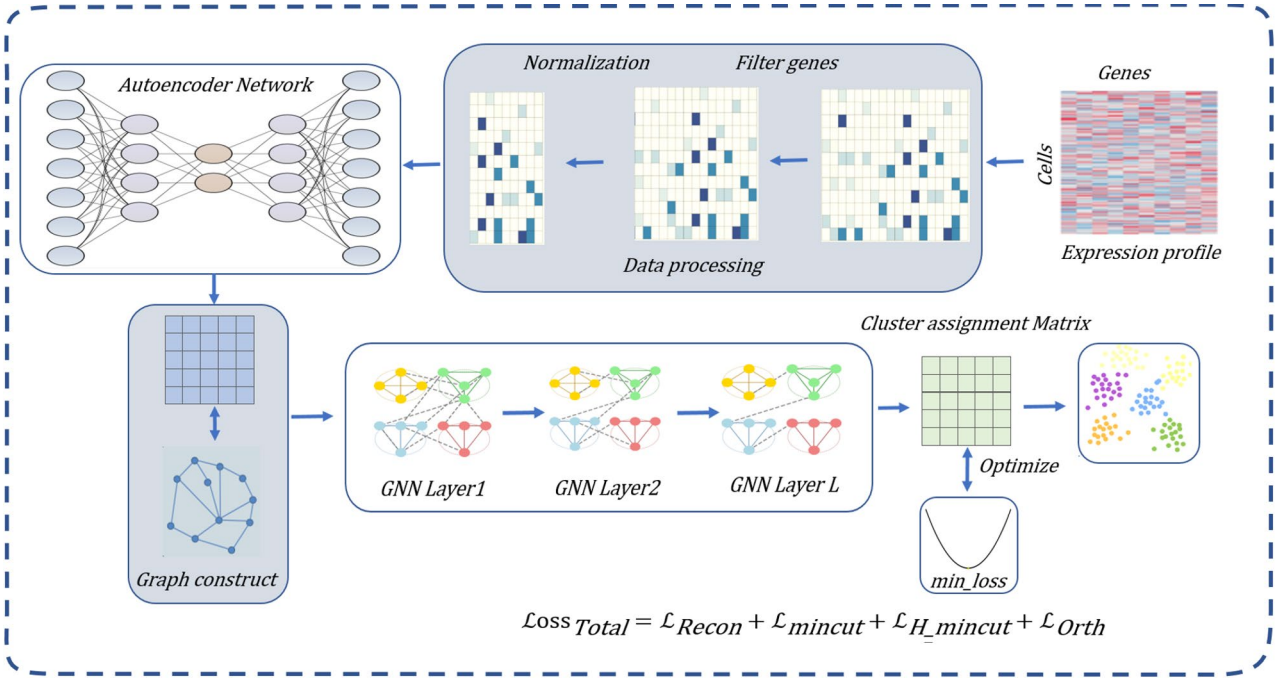

Fig. 1 Overview of our approach

Problem definition

Given single-cell transcriptome data, it is represented as a matrix $X \in \mathbb{R}^{n \times m}$, where n is the number of cells and m is the number of genes. The gene expression vector of each cell is denoted as X_i. Our goal is to assign these cells into different cell types by clustering such that cells of the same type have a higher similarity in gene expression patterns.

Preliminaries

Define the adjacency matrix of an undirected graph $G = \{V, E\}$ as $A \in R^{n \times n}$ and the node degree matrix D=diag(A). The goal of graph partitioning is to divide V into K mutually exclusive subsetsThe gusive subsets $C_1, C_2, \ldots, C_K$. Traditional mincut minimizes the sum of edge weights between clusters [21, 22]:

$$\text{mincut}\,(C_1, C_2, \ldots, C_K) = \sum_{i=1}^{k} \text{cut}\left(C_i, \overset{?}{C_i}\right) \quad (1)$$

To avoid cutting all nodes to one cluster, normalized mincut is used:

$$\text{Norm\_mincut} \sum_{i=1}^{k} \frac{\text{cut}\left(C_i, \overset{?}{C_i}\right)}{\text{vol}\,(C_i)} \quad (2)$$

In the formula, $\text{vol}\,(C_i) = \sum_{u \in C_i} \sum_{v \in V} A_{uv}$, is the sum of all the nodes in the cluster C_i degree nodes [23, 24].

Figure 2 shows three common local structures in network analysis: (a) Simple edge connection (2-node); (b) The triangular closed structure (3-node) reflects the local cluster characteristics and is the basic unit for measuring the transitivity and clustering of the network. (c) The cyclic structure (4-node) represents the cyclic path in the network. These structures are the fundamental units for understanding the principles of complex networks.

Higher-order mincuts extend the traditional mincut problem by considering higher-order connection patterns in graphs (such as triangles, 4-node cycles, etc.). The objective of motif conductance is to minimize the number of inter-group motif cuts while maximizing the number of intra-group motifs. The high-order derived degree is defined as follows.

$$Norm\_mincut_M = \sum_{i=1}^{k} \frac{\text{cut}_M(C_i, \overset{'}{C_i})}{\text{vol}_M\,(C_i)} \quad (3)$$

Here: $cut_M(C_i, \bar{C}_i)$ is a Motifs Cut: it represents the number of motifs when two nodes belong to different groups. $\text{vol}_M\,(C_i)$ is the Motif Volume and represents the number of nodes of the motif whose nodes are in the group. $cut_M(C_i, \bar{C}_i)$and $\text{vol}_M\,(C_i)$.

represent the number of motif instances across clusters and within clusters, respectively.

The traditional MinCut objective [21] aims to minimize the weight of connections between clusters, but its

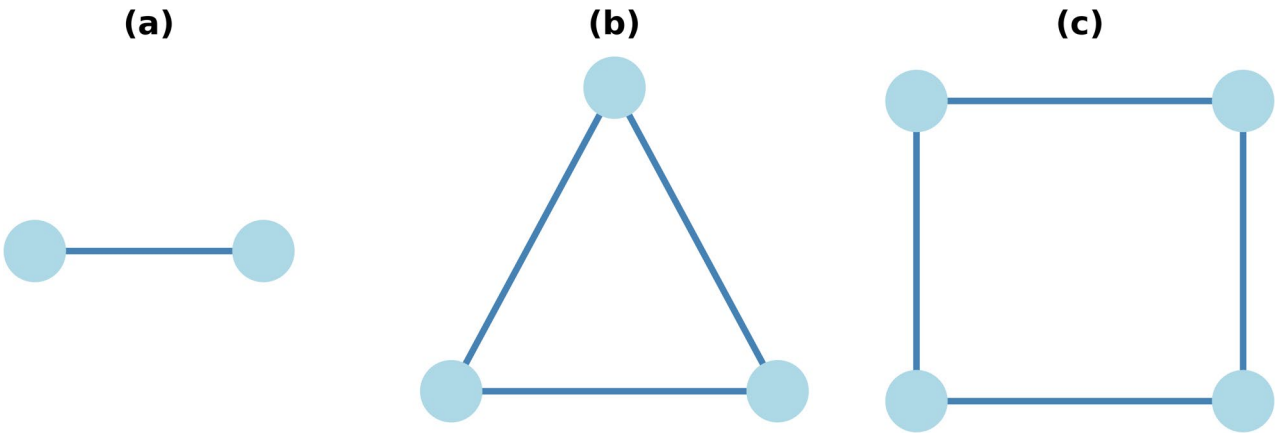

Fig. 2 Three representative motifs in biological networks, (**a**) edge, (**b**) triangle motif, (**c**) 4-node cycles

discrete optimization is NP-hard. Spectral clustering [25, 26] provides a relaxation via eigen-decomposition, while local graph partitioning methods [27] offer an alternative perspective. However, traditional spectral clustering requires the eigen-decomposition of the Laplacian matrix [25, 26, 28], and the computational complexity increases with n^3 and it is easy to fall into local minima. Dimensionality reduction techniques, such as trace ratio optimization [29, 30], can reduce computational costs by projecting data into a lower-dimensional space, but they may not fully address the fundamental limitations of spectral clustering in capturing complex graph structures. To overcome these challenges, our deep clustering strategy is inspired by recent advances in graph neural networks (GNNs) [31, 32], which highlight the advantages of GNNs in graph pooling and incorporate ideas from trace ratio optimization [29] and dimensionality reduction [30] into the optimization process. Specifically, in this paper, we adopt the continuous relaxation strategy from [33] to relax Eq. (3) as a differentiable loss and embed it into the end-to-end training of the GNN; details are provided in subsequent sections. Since the discrete optimization of Eq. (3) is NP-hard, this approach transforms the hard assignment matrix into a probabilistic simplex, enabling differentiable optimization and seamless integration with GNN training via backpropagation. By leveraging continuous relaxation and backpropagation in the GNN framework, node embedding and the soft assignment matrix are learned synchronously, thereby avoiding the explicit calculation of feature vectors [34]. Further details are elaborated in subsequent sections.

Dimensionality reduction with autoencoder

We employed a deep autoencoder [35] to denoise the data and learn its salient features [36], a strategy commonly used in scRNA-seq data analysis [37], which is optimized by minimizing the MSE reconstruction error:

$$min_{\theta_e,\theta_d} L_{AE} = \frac{1}{n}\sum_{i=1}^{n} \parallel X_i - \widehat{X}_i \parallel_2^2 \tag{4}$$

Here, $\widehat{X}_i$ is the output of the decoder and represents the reconstruction of the input data X_i. By minimizing the reconstruction error, the autoencoder is able to learn important features of the data while removing part of the noise.

Construction of the cell adjacency matrix

Using the hidden feature data Z after dimension reduction, we construct the adjacency matrix A between cells. Specifically, we use a Gaussian kernel function to compute the similarity between cells:

$$A_{ij} = \exp(-\frac{\parallel Z_i - Z_j \parallel_2^2}{2\sigma^2}) \tag{5}$$

Here, σ is the width parameter of the Gaussian kernel, which controls the rate of decay of the similarity. For each cell, the top K nearest neighbors are selected based on this similarity measure. The selection of σ and K will be discussed in detail in the parameter analysis and simulation data analysis sections. To ensure the connectivity of the graph, we usually normalize the adjacency matrix and add self-connections.

Node embedding with graph neural networks

Graph Neural Networks have shown great potential in this domain [38, 39]. We employ topologically adaptive graph convolution TAGConv [40] to learn cell embeddings. TAGConv captures high-order path patterns by aggregating 0-K order neighbor information [41], and introduces attention weights to automatically learn the importance of different orders. Its propagation rule for the l-th layer is

$$\mathbf{H}^{(l+1)} = \sigma\left(\sum_{k=0}^{k} \alpha^{(l,k)} \mathbf{A}^k \mathbf{H}^{(l)} \mathbf{W}^{(l,k)}\right) \tag{6}$$

Where: $H^{(l)}$ is the node feature matrix of the l th layer, where each row represents the feature vector of a node A is the adjacency matrix of the graph, A^k denotes the adjacency matrix of order k, Where A^k records the number of paths of length k, $\alpha^{(l,k)}$ is the attention coefficient, which is calculated by the learnable vector $\alpha^{(l,k)}$ and LeakyReLU activation [40]. We take K = 3 in all experiments, and the final output H^l is used as the cell representation for subsequent clustering and reconstruction constraints.

End to end cluster assignment

After obtaining the node embedding H_L, we directly input the two-layer MLP to obtain the soft assignment.

matrix: $S = \mathrm{Softmax}\left(MLP\left(H^{(l)}\right)\right)$ (7)

Each row S_i represents the probability that cell I belongs to each cluster. Different from the classical spectral clustering that requires re-feature decomposition of new samples [25, 26], this mapping is updated through back propagation together with GNN parameters in the training phase to realize the end-to-end joint optimization of "embedding learning-clustering assignment" [34]. During training, S is used to compute mincut, higher-order mincut, and orthogonal loss, as detailed in the following sections.

First-order mincut loss

To reduce the connection weights between different cell clusters, we adopt normalized mincut as the clustering loss [18].

$$\mathcal{L}_{\mathrm{mincut}} = -\frac{\mathrm{Tr}\left(S^{\mathrm{T}}AS\right)}{\mathrm{Tr}\left(S^{T}DS\right)} \tag{8}$$

Where S is the node-to-cluster assignment matrix, and each row represents the probability that a node belongs to each cluster. A is the adjacency matrix of the graph; Tr() denotes the trace of a matrix. $\mathrm{Tr}\left(S^{\mathrm{T}}AS\right)$ represents the sum of weights of inter-cluster connections, and $\mathrm{Tr}\left(S^{\mathrm{T}}DS\right)$ represents the sum of weights of intra-cluster connections.

Higher-order minicut loss

While the first-order MinCut loss in Eq. (10) only considers pairwise connections, it fails to capture higher-order structures like triangles, which are crucial for defining robust cell communities [19, 42]. To address this, we introduce a higher-order MinCut loss that operates on triangle motif-based adjacency matrices. The formulation of triangle motif is

$$\mathbf{A}_M = \mathbf{A} \circ \mathbf{A}^2 \tag{9}$$

$\mathbf{A}$ is the adjacency matrix of the network, with $A_{ij} = 1$, if nodes i and j are connected, and 0 otherwise. $\mathbf{A}^2$ represents the number of length-2 paths between node pairs, $\circ$ denotes the Hadamard (element-wise) product. $\mathbf{A}_M$ encodes triangle structures, where $(\mathbf{A}_M)_{ij} > 0$ indicates that nodes i and j participate in triangles. This matrix-based representation provides a computationally efficient framework for analyzing higher-order network structures.

We choose the triangular motif as the representative of high-order structures mainly based on the following considerations: Firstly, the triangular motif is the simplest high-order connection mode in the network, and its computational complexity is relatively low. Secondly, the triangular motif is closely related to the local aggregation characteristics of the network and serves as the basis for measuring the clustering coefficient and transitivity of the network.

After the motif adjacency matrix $A_{\mathcal{M}}$ is constructed where each element $(A_{\mathcal{M}})_{\mathrm{ij}}$ counts the number of motifs involving both node I and j, the higher-order mincut loss is defined as follows.

$$\mathcal{L}_{\mathrm{H\_mincut}} = -\frac{\mathrm{Tr}\left(S^{\mathrm{T}}A_{\mathcal{M}}S\right)}{\mathrm{Tr}\left(S^{\mathrm{T}}D_{\mathcal{M}}S\right)} \tag{10}$$

Where S is the node-to-cluster assignment matrix, and each row represents the probability that a node belongs to each cluster. A is the adjacency matrix of the graph; Tr() denotes the trace of the matrix, $\mathrm{Tr}\left(S^{\mathrm{T}}A_{\mathcal{M}}S\right)$ denotes the sum of node weights of participating motifs between different clusters, and $\mathrm{Tr}\left(S^{\mathrm{T}}D_{\mathcal{M}}S\right)$ denotes the sum of node weights of participating motifs within a cluster.

Inter-cluster orthogonality loss

The orthogonality constraint encourages the column vectors of S be orthogonal. This not only separates cluster centroids in the embedding space but also implicitly balances the cluster sizes, preventing the degeneration where all cells are assigned to a few large clusters. To prevent embedding space overlap and balance cluster sizes, we force the cluster assignment matrix to be approximately column-orthogonal. Inter-cluster orthogonal loss is defined as follows.

$$\mathcal{L}_{\mathrm{Orth}} = \left\| \frac{\mathbf{S}^T\mathbf{S}}{||\mathbf{S}^T\mathbf{S}||_F} - \frac{\mathbf{I}_K}{\sqrt{K}} \right\|_F \tag{11}$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm, S is cluster assignment matrix, $\mathbf{I}_K$is the identity matrix, and K is the number of clusters. Inter-cluster orthogonal loss can effectively prevent the overlap of clusters in the

Table 1 Summary of scRNA-seq datasets in this study

Dataset	Class	Cells_Num	Organ	Platform	Ref.
Adam	8	3660	Kidney	Drop-seq	[47]
Bach	8	23,184	Mammary Gland	10x	[48]
Klein	4	2717	Embryonic Stem Cell	inDrop	[49]
Muraro	9	2122	Pancreas	CEL-seq2	[50]
QS_Diaphragm	5	870	Diaphragm	Smart-seq2	[51]
QS_Heart	8	4365	Heart	Smart-seq2	[51]
QS_Lung	11	1676	Lung	Smart-seq2	[51]
Qx_Bladder	4	2500	Bladder	10x	[51]
Tosches_turtle	15	18,664	Turtle Brain	Drop-seq	[52]
Wang_Lung	2	9519	Lung	10x	[53]

embedding space, balance the number of nodes between clusters, and make the clustering results more clear and interpretable.

Topology reconstruction loss

The pure cut objective function may over-partition the graph, so the inner product is used to reconstruct adjacencies to constrain the global structure [43]. Cross-entropy is used to ensure that A' is consistent with the original A distribution, formally, $A' = \text{sigmoid}\left(\mathbf{H}_L^T \mathbf{H}_L\right)$,we use cross-entropy loss to calculate the difference between the newly constructed adjacency matrix A' and the original matrix A, and minimize this loss as a constraint in the learning process of graph neural network:

$$\mathcal{L}_{\text{Recon}} = -\frac{1}{\text{n}} \sum_{i=1}^{n} \sum_{j=1}^{n} [A_{ij} \log A'_{ij} + (1 - A_{ij}) \log(1 - A'_{ij})] \quad (12)$$

Total loss function

The overall training objective is to minimize the following total loss function (Eq. 14), which is a sum of all previously described loss components, we define the total loss function as follows.

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Recon}} + \mathcal{L}_{\text{mincut}} + \mathcal{L}_{\text{H\_mincut}} + \mathcal{L}_{\text{Orth}} \quad (13)$$

By jointly optimizing the total loss function, we can achieve dimensionality reduction and denoising of data, graph structure learning and cluster assignment optimization at the same time, thereby improving the performance of clustering.

Table 2 Summary of baseline methods and their capabilities

Method	Category	Core Idea / Technical Feature
PCA + k-means [10]	Traditional Linear Model	Performs k-means clustering after dimensionality reduction via Principal Component Analysis (PCA)
Seurat [44]	Classic scRNA-seq Tool	Clusters a nearest-neighbor graph using community detection algorithms (e.g., Louvain)
Scanpy [45]	Classic scRNA-seq Tool	Provides various graph clustering and Leiden algorithm options; a mainstream tool in the Python ecosystem
DCA [43]	Deep Learning Method	Uses Denoising Autoencoders for dimensionality reduction and imputation, followed by clustering
scDeepCluster [13]	Deep Learning Method	Jointly optimizes an autoencoder and a clustering loss for end-to-end clustering
scziDesk [46]	Deep Learning Method	Handles dropout noise using a zero-inflated negative binomial model and autoencoders
DESC [14]	GNN-based Method	Learns embeddings using a graph autoencoder and employs a self-training mechanism to refine clustering
scBGEDA [20]	GNN-based Method	Guides clustering by combining graph autoencoders with biological pathway prior knowledge
Our DGM	GNN-based Method	Coordinates high-order MinCut loss, orthogonality loss, and topology reconstruction constraint

Evaluation metrics

To evaluate the clustering performance, we employ the following commonly used evaluation metrics, with higher values indicating better clustering. Normalized Mutual Information (NMI): measures the mutual information between the clustering results and the true labels, Adjusted Rand Index (ARI): This measures the agreement between the clustering results and the true labels, with higher values indicating better clustering. These evaluation metrics are able to evaluate the clustering performance from different perspectives, providing us with a comprehensive performance evaluation.

Experimental results and analysis

To evaluate the performance of our proposed DGM model, we evaluate the model on 10 public scRNA-seq datasets covering a variety of organizations and platforms (e.g., 10x, Drop-seq, Smart-seq2). All datasets are publicly available and no ethical approval is required. Table 1 summarizes the details of the different datasets. We compare it with eight representative baseline methods covering different technological routes. The comparison methods include traditional methods (PCA, Seurat [44], Scanpy [45]) and deep learning methods scBGEDA [20], DCA [43], scDeepCluster [13], scziDesk [46], DESC [14]. Etc.). Table 2 summarizes the core features of these baseline approaches.

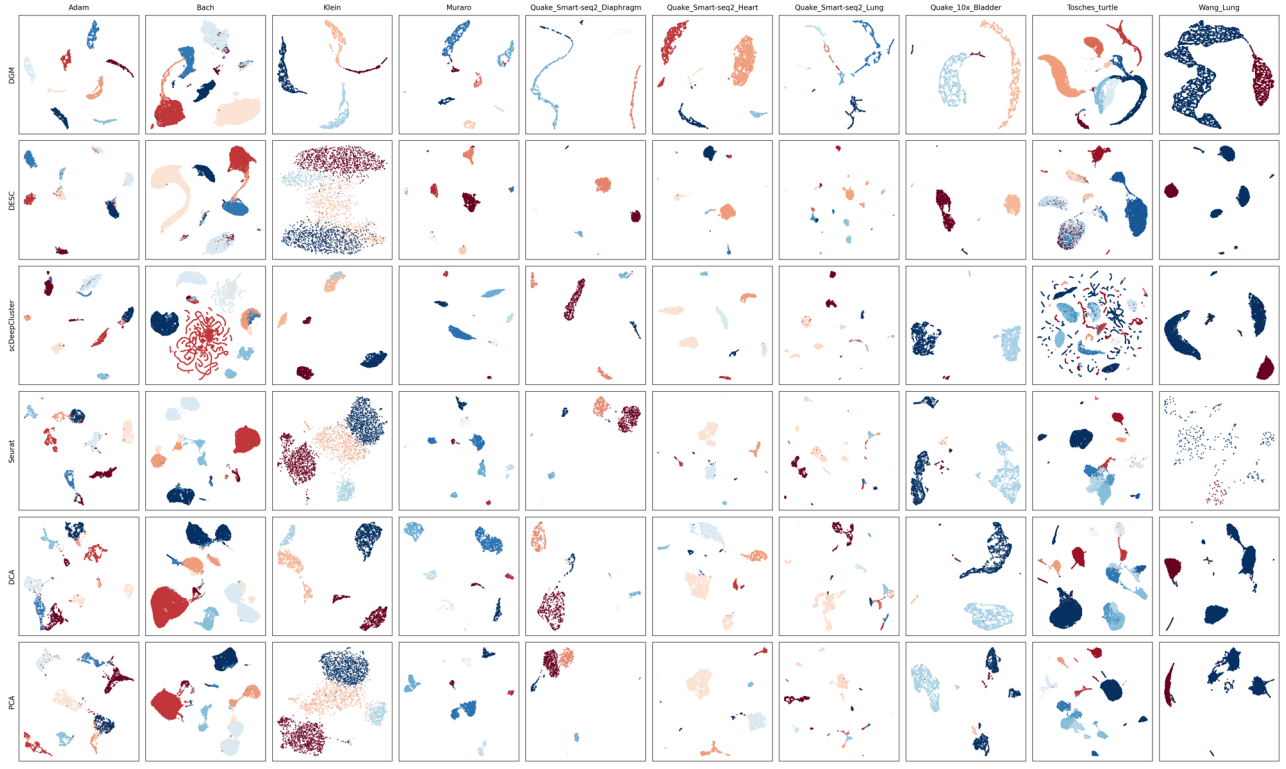

Fig. 3 Visualization of clustering results via DGM and baseline methods

Table 3 Silhouette score of several datasets obtained by baseline methods

	Adam	Bach	Klein	Muraro	QS_Diaphragm	QS_Heart	QS_Lung	Qx_Bladder	Tosches_turtle	Wang_Lung
DGM	**0.6924**	*0.5606*	**0.8098**	*0.6686*	**0.7815**	*0.5550*	**0.6695**	*0.6921*	**0.5221**	0.8900
DESC	0.4452	**0.6528**	0.118	0.3726	0.2796	0.2790	0.3355	0.4199	0.342	*0.4532*
scDeepCluster	*0.5240*	0.5500	*0.4578*	**0.7642**	*0.6942*	**0.5697**	0.3838	**0.8244**	*0.4199*	0.2163
Seurat	0.0524	0.0709	0.0448	0.1024	0.0752	0.0853	0.1507	0.0530	0.1147	-0.034
DCA	0.1641	0.3061	0.3294	0.4425	0.3682	0.1662	0.3411	0.4123	0.3271	0.0697
PCA	0.0828	0.0916	0.0439	0.1590	0.1004	0.0957	0.1638	0.0657	0.1599	0.1948

We carried out clustering experiments on these ten datasets, recorded the experimental results, and compared our results with several existing methods. As can be seen from Fig. 4, scBGEDA, scziDesk, scDeepCluster and other methods also perform well, and our results in the other three datasets are close to the top-ranked method, also showing competitive performance. For each dataset, we run each setting with multiple random seeds and record the median value. Finally, we compared the average performance gains of all methods across the ten datasets. Fig. (4a, b, c and d) shows the proposed method achieves the optimal NMI and ARI on 7/10 datasets, and the other 3 datasets are also ranked in the top two, which are better than the existing optimal methods, verifying the robustness and generalization ability.

As shown in Fig. 3, we applied different clustering methods to multiple single-cell transcriptome datasets and reduced the high-dimensional data to two dimensions by UMAP for visualization [54]. It can be observed that DEC and scDeepCluster also show strong performance, and these methods show strong advantages when dealing with large-scale datasets. Meanwhile, our proposed method (Ours) also shows clear clustering boundaries and reasonable cell population distribution on multiple datasets, showing similar performance with previous methods. For example, in the Adam and Bach datasets, our method can identify more cell populations with a clear degree of discrimination, which demonstrates the effectiveness of our method when dealing with single-cell transcriptome data.

As shown in Table 3, We use the Silhouette score (the ratio of intra-module compactness to inter-module separation) to evaluate the clustering quality. The closer the silhouette coefficient is to 1, the better the clustering quality is. Among the comparison of several methods, DGM achieved the highest contour coefficients in six of the ten datasets (Adam, Bach, Klein, QS_Diaphragm, QS_Lung, Wang_Lung), indicating that its clustering

results have the best cohesion and separability. scDeepCluster and DESC also demonstrated relatively high performanc.

Ablation study

The ablation results on the ARI and NMI metric (Fig. 4e and f) clearly show that removing any module leads to a performance degradation, confirming the indispensability of each component in the DGM framework, as the removal of any module leads to a marked decline in average NMI and ARI. The contribution of individual modules, however, exhibits dataset-dependent characteristics. For instance, the high-order MinCut loss is the dominant factor on the Adam dataset. A similar trend was observed for the ARI metric (Fig. 4e). Notably, the orthogonality loss contributed most significantly on the Bach dataset. This context-specificity underscores the necessity for future research into adaptive loss weighting mechanisms.

The synergistic effect of our joint optimization strategy is key to the performance gain. The autoencoder provides a denoised and low-dimensional feature representation. The GNN captures complex cell-cell relationships. Crucially, the high-order MinCut loss preserves the graph's topological structure, while the orthogonality loss enhances inter-cluster separability. The topology reconstruction constraint prevents over-segmentation and maintains global consistency. This integrated approach enables DGM to capture high-order structures effectively while avoiding topological collapse.

Parameter analysis

We studied the effects of K and σ. We tested different combinations of K and σ on 10 datasets. The range of

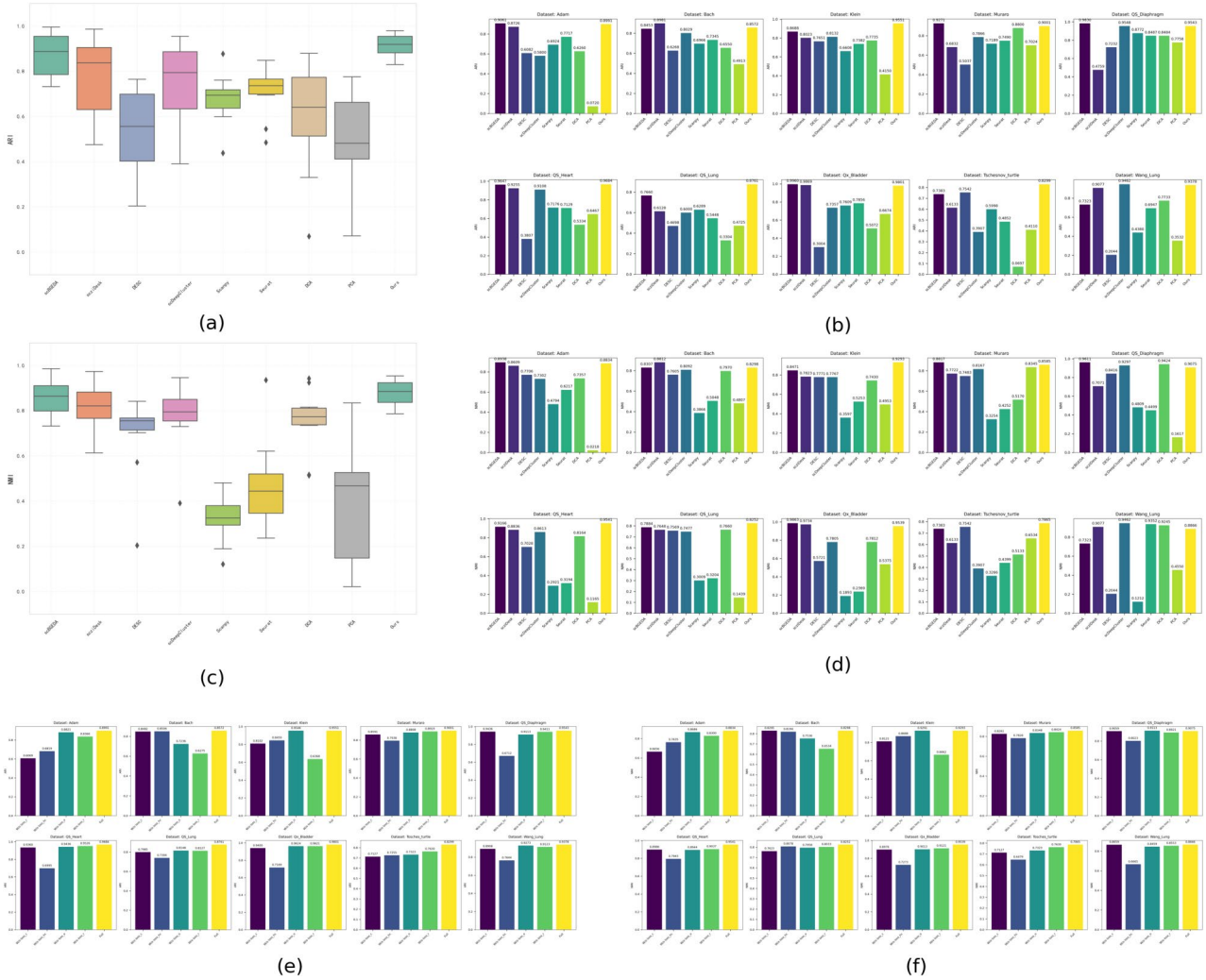

Fig. 4 ARI and NMI Performance Comparison, (**a**) Box plots of the ARI metric for various methods across the ten datasets. **b** Bar chart comparing ARI scores for the methods across the datasets. **c** Box plots of the NMI metric for various methods across the ten datasets. **d** Bar chart comparing NMI scores for the methods across the datasets. **e** Bar chart of the ARI metric for the ablation study of the proposed method across the datasets. **f** Bar chart of the NMI metric for the ablation study of the proposed method across the datasets

K was [3,5,10,15,20]. The range of σ was [1,3,5,10,100]. Fig. 5 shows the average ARI values for different combinations. The ARI values change non-monotonically. They show a non-linear transformation. The highest values appear near the yellow region. The lowest values are in the dark blue region. For topk, medium values (around 10–15) give higher ARI. When topk is too small (close to 5) or too large (close to 20), ARI decreases. The reason is that a too small K cannot construct enough edges. This makes the data too sparse. A too large K adds more edges. But it may also introduce noisy edges. For σ, as σ increases, ARI changes non-linearly. It peaks near a medium value. Too small or too large σ values lower ARI for the reason that a too large σ favors local information of nodes and a too small σ favors global information. A balance between global and local information is needed.

Cell type identification and marker gene analysis

Taking Adam dataset as an example, we further carried out differential expression analysis to verify whether the embedding representation obtained by the scDGM method can facilitate functional genomics interpretation. Adam dataset [47] uses psychrophilic protease technology to obtain high-quality single-cell transcriptome of kidney development. As shown in Fig. 6, the clustering results successfully reproduce the cell type-specific expression of known marker genes (such as Krt18, Fabp4, Slc12a1, and Emx1).

We used Scanpy [45] for downstream analysis of the clustering results (Fig. 6). Firstly, we plot the expression heat map of the top three differentially expressed genes in each cell cluster across all cells (Fig. 6a). Furthermore, we used the scanpy package to identify the most significantly differentially expressed genes in each cluster compared to other clusters The heat map, and violin plot consistently showed that Fabp4, Slc12a1, Emx1 and other marker genes were specifically expressed in each cluster. UMAP 2D visualization (Fig. 6c) reveals that Corresponding cell clusters can be identified based on marker genes, which verifies that the clustering results are highly consistent with biological knowledge and have interpretability and usability. Fig. 6d shows the similarity between marker genes from different methods and the gold standard cell types. Each column represents a cluster from a specific method. Each row represents a known gold standard cell type. The cell types are CM, Distal Tubule, ED, LH, PT, Podocytes, Stroma, and Ureteric Bud. We used the top 50 differentially expressed genes to identify specific cell

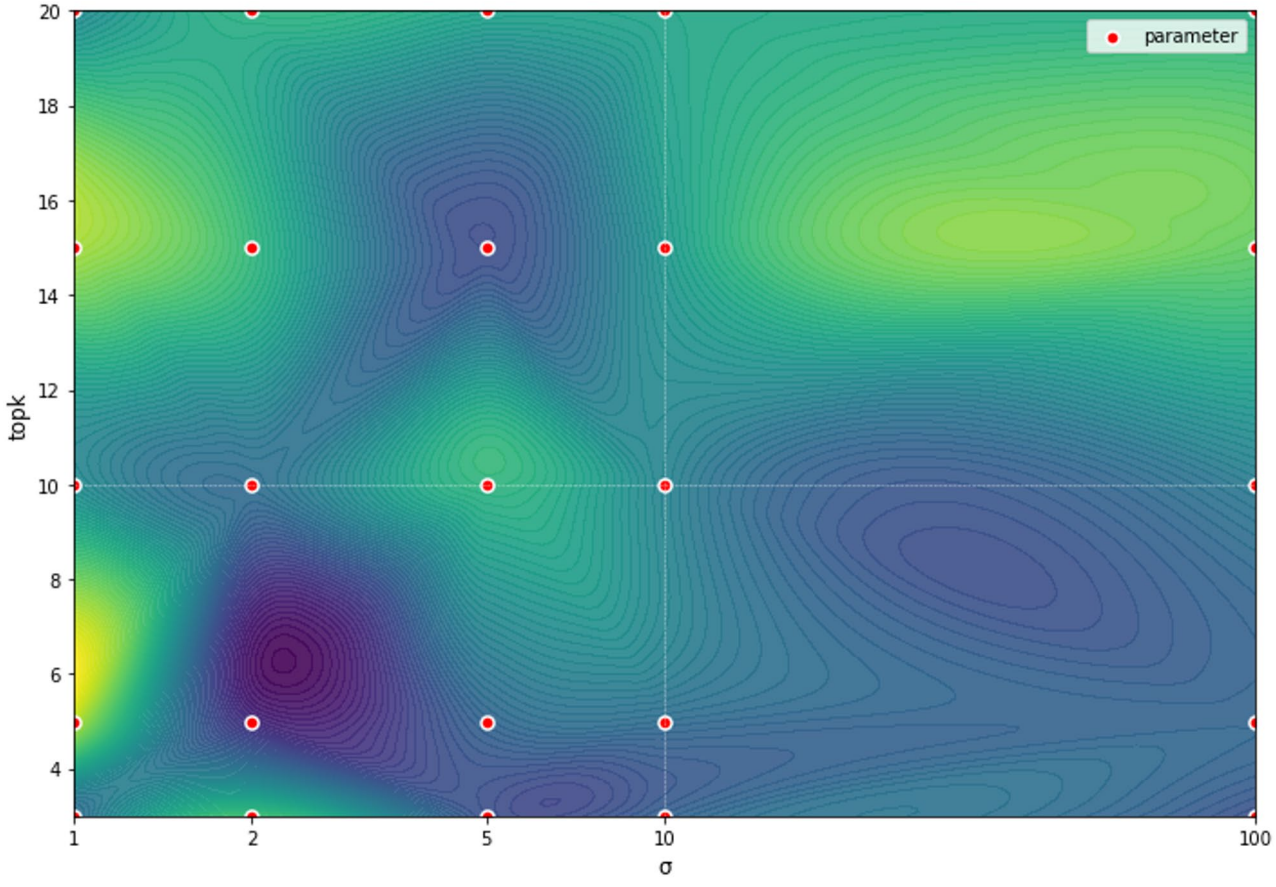

Fig. 5 ARI heatmap for different parameter combinations

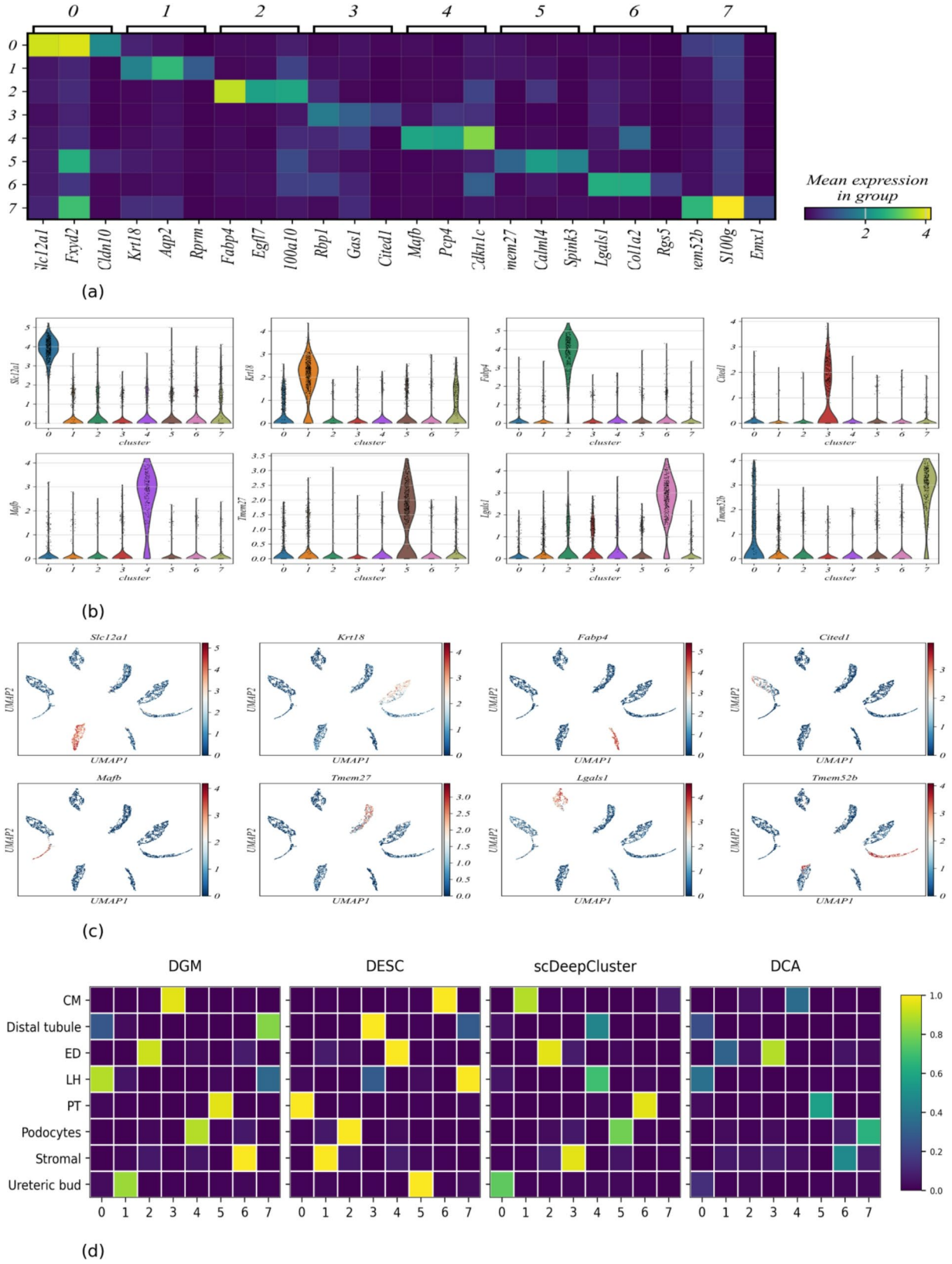

Fig. 6 Marker Gene Expression and Cell Type Identification Performance (**a**), Expression of top marker genes across the identified clusters, shown as a heatmap, (**b**) Violin plots display the expression distribution of the most significant marker gene for each cluster. **c** UMAP projections visualize the spatial expression patterns of the marker genes within the cell atlas. **d** Similarity matrix compares the cell clusters identified by four distinct deep learning methods (DCA, DESC, scDeepCluster, and DeepCluster) against established reference cell types, based on the similarity of their top marker gene profiles

types. Methods DESC and scDeepcluster also achieved high similarity. Our results are consistent with popular methods.

Trajectory inference and enrichment analysis

Figure 7a depicts the core lineage relationships in mouse kidney development, which is consistent with established biological literature. In the diagram, CM (Cap Mesenchyme) is positioned as the central starting point. During kidney development, CM serves as a multipotent progenitor pool that generates the epithelial cells of the nephron. An arrow points from CM to Oocytes, highlighting the key and direct differentiation pathway for glomerulus formation. Furthermore, CM is connected to other major terminal cell types, including PT (Proximal Tubule) and Distal tubule/LH. This schematic clearly presents the diverse cellular origins in the kidney and illustrates the main trunk of epithelial differentiation centered on CM, accurately reflecting their anatomical and functional continuity.

Functional enrichment analysis was performed on the two identified cell clusters: Podocytes and PT (Proximal Tubule) cells. Fig. 7b presents the results of gene set enrichment analysis, comparing the biological functions of two distinct renal cell populations: Podocytes and Proximal Tubule (PT) cells. The side-by-side bar charts reveal a striking and functionally coherent dichotomy in their enriched biological processes, strongly validating the cell type annotations. The Podocyte cluster is significantly enriched for processes intrinsic to its identity, such as Podocyte Differentiation, Glomerular Epithelial Cell Differentiation, and Kidney Development. In contrast, the PT cell cluster shows enrichment for pathways directly related to its core physiological functions, including Fructose Metabolic Process, Receptor-Mediated Endocytosis, and Phosphate Ion Homeostasis. This pattern of enrichment validates the cell type annotations, as the significant biological processes for each cluster align with their established roles in renal structure and function.

Simulated data analysis

We used six simulated datasets generated by Splatter [55, 56] to test scDGM. The dropout rates ranged from 0.05 to 0.25. The clustering performance, evaluated by NMI and ARI, was compared against baseline methods in(Fig. 8a and b).Cell visualization results for one dataset are shown in Fig. 8c. scDGM clearly separated the five cell types.

At series dropout rate from 0.05 to 0.20, DGM outperformed all baselines across different levels. Performance decreased slightly when the dropout rate increased to 0.25. To investigate this, we gradually increased the value of K on the dropout_25 dataset. Fig. 8d shows that NMI and ARI improved as K increased. The metrics stabilized after K exceeded 25. This suggests that the high-order cut is effective for dense graph structures and higher-order information, such as triangular motifs, can effectively reveal deeper relationships in dense subgraphs. However, in sparse graphs, there may not be too many higher-order structures for this method to detect. Thus, the high-order cut is useful, but its effectiveness depends on specific conditions.

Discussion and conclusion

Our study presents the Deep Graph MinCut (DGM) framework, which advances single-cell clustering by simultaneously addressing limitations of existing methods: the neglect of higher-order topology, ambiguous cluster boundaries, and over-cut. When compared to recent graph-based methods that focus on first-order convolutions [14], DGM's integration of higher-order MinCut loss, inter-cluster orthogonality constraints,

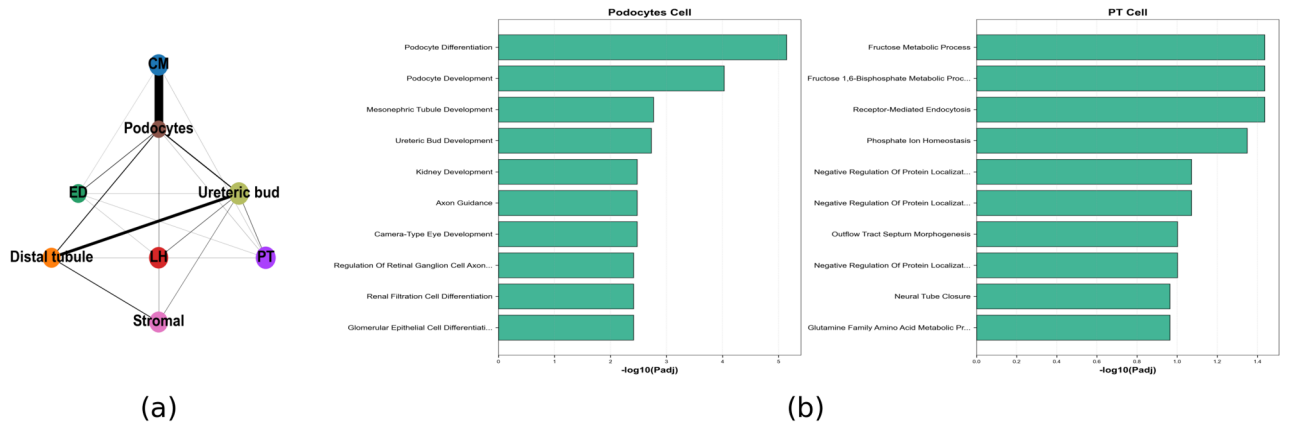

Fig. 7 Trajectory Inference and Cell Type-Specific Functional Enrichment. **a** Trajectory inference reveals the potential lineage relationships among major cell types. **b** Functional enrichment analysis demonstrates the specialized roles of two key epithelial cell types. The Podocyte cluster is enriched for developmental and structural pathways, while the Proximal Tubule (PT) cluster is enriched for metabolic and transport processes, aligning with their distinct biological functions

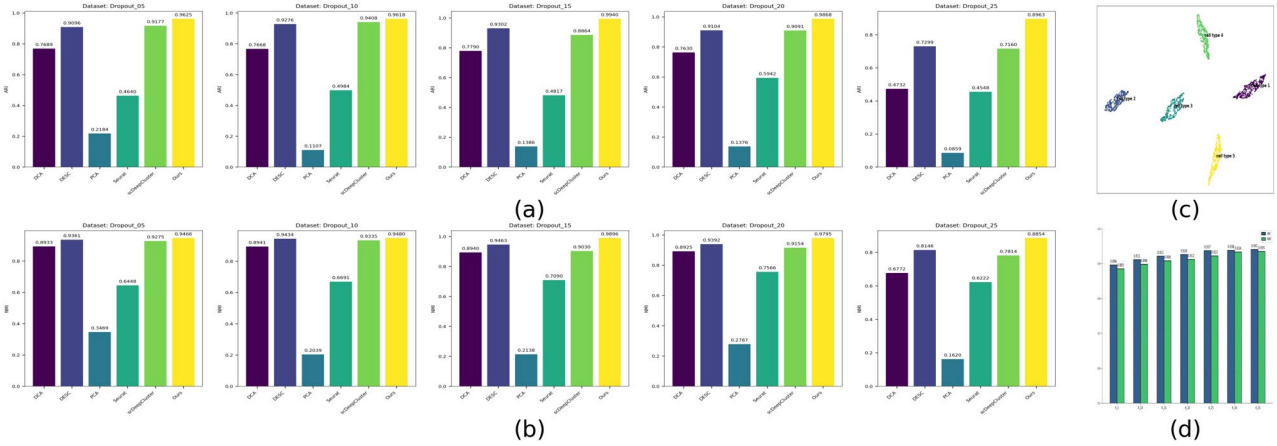

Fig. 8 Evaluation of Model Performance on Simulated Datasets. **a** Comparison of ARI scores across baseline methods and DGM under varying dropout rates (**b**) Comparison of NMI scores across baseline methods and DGM under varying dropout rates. **c** Impact of parameter K on DGM performance on the dataset with the highest dropout rate (dropout_25). **d** Clustering visualization for a simulated dataset (e.g., dropout_25), where data points are colored according to five identified cell type clusters, demonstrating clear separation

and topology reconstruction offers a more integrated approach. Evaluated on ten public scRNA-seq datasets, DGM achieved robust performance, with average improvements of 2.03% in NMI and 3.11% in ARI, outperforming state-of-the-art methods and underscoring its generalizability.

Crucially, the value of DGM extends beyond computational metrics to direct biological utility. Downstream analysis demonstrated strong concordance with known marker genes and developmental trajectories. For instance, as shown in Fig. 6, DGM accurately recapitulates the specific expression of key markers in distinct renal cell types. This biological interpretability suggests that DGM can serve as a tool for discovery, with potential to aid in applications such as characterizing tumor microenvironments or facilitating the discovery of novel biomarkers [57].

Despite its strengths, our work has several limitations that should be acknowledged. First, the computational complexity of graph construction and high-order motif calculation may present challenges for ultra-large-scale datasets. Second, the performance of DGM can be influenced by the hyperparameters of the graph construction process (e.g., the Gaussian kernel width σ and top k). Finally, while we demonstrated effectiveness on scRNA-seq data, its applicability to other single-modal or multi-omics data (e.g., spatial transcriptomics) requires further investigation. We acknowledge that these limitations define the current scope of our method's application.

To address these limitations, future work will focus on addressing the current limitations to broaden DGM's applicability. Concurrently, to improve usability and robustness, we will develop automated hyperparameter optimization techniques. These methodological advancements will be crucial for reliably applying DGM to large-scale clinical datasets, which is necessary to thoroughly validate its utility in practical scenarios, such as discovering diagnostics biomarkers or characterizing tumor microenvironment heterogeneity.

In summary, DGM offers an interpretable and potentially robust framework for single-cell transcriptomics through the integration of higher-order graph cuts with cluster orthogonality constraints. The results obtained with DGM may support ongoing biomedical research, particularly in the area of biomarker discovery, with the long-term aim of potentially contributing to improved diagnostic and therapeutic strategies.

Authors' contributions

Xi Liu designed this study and conducted deep learning algorithms. Yong Yu, Xiaolin Chen provided technical support for the experiment. Xi Liu drafted the initial manuscript, Yong Yu and Wenqian Yang revised the manuscript. All authors read and agreed to the final manuscript.

Funding

No funding was received for conducting this study.

Data availability

The datasets analyzed during this study are publicly available. The dataset can be available in the figshare repository, [https://doi.org/10.6084/m9.figshare.19657911.v2]. The source code will be available in the GitHub repository, [https://github.com/linuxclab/scDGM]

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 27 October 2025 / Accepted: 23 February 2026
Published online: 09 March 2026

References

1. Zeisel A, Muñoz-Manchado AB, Codeluppi S, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. Science. 2015;347(6226):1138–42.
2. Shapiro E, Biezuner T, Linnarsson S. Single-cell sequencing-based technologies will revolutionize whole-organism science. Nat Rev Genet. 2013;14(9):618–30.
3. Patel AP, Tirosh I, Trombetta JJ, et al. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. Science. 2014;344(6190):1396–401.
4. Shalek AK, Satija R, Shuga J, et al. Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. Nature. 2014;510(7505):363–9.
5. Xin Y, Coré N, Oberlin E, et al. High-throughput single-cell RNA-sequencing method for sensitive detection of cell type-specific gene expression. Genome Res. 2016;26(12):1605–12.
6. AlJanahi AA, Danielsen M, Dunbar CE. An introduction to the analysis of single-cell RNA-sequencing data. Mol Therapy: Methods Clin Dev. 2018;10:189–96. https://doi.org/10.1016/j.omtm.2018.07.003.
7. Kriegel HP, Kröger P, Zimek A. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans Knowl Discovery Data. 2008;3(1):1–58.
8. Soneson C, Robinson MD. Bias, robustness and scalability in single-cell differential expression analysis. Nat Methods. 2018;15(4):255-261.
9. Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. 2007:1027–35.
10. Tian L, et al. Principal component analysis (PCA) of single-cell transcriptome data. Nat Methods. 2019;16(10):967–8.
11. Likas A, Vlassis N, Verbeek JJ. The global k-means clustering algorithm. Pattern Recogn. 2003;36(2):451–61.
12. Grun D, Lyubimova A, Kester L, et al. Single-cell messenger RNA sequencing reveals rare intestinal cell types. Nature. 2015;525(7568):251–5.
13. Tian L, Kang B, Yu H, Zhang X. scDeepCluster: Deep clustering of single-cell RNA-seq data. Bioinformatics. 2019;35(21):4309–10.
14. Li H, Chen M, Jiang R, Li Y. DESC: Deep embedded single-cell clustering with high-order graph convolution. Bioinformatics. 2020;36(19):5174–81.
15. Fard MM, Thonet T, Gaussier E. Deep k-Means: Jointly clustering with k-Means and learning representations. Pattern Recognit Lett. 2020;138:185–92. https://doi.org/10.1016/j.patrec.2020.07.028.
16. Peng L, Tian X, Tian G, et al. Single-cell RNA-seq clustering: Datasets, models, and algorithms. RNA Biol. 2020;17(6):765–83.
17. Wu Y, Zhang J, Zhang Y. Single-cell RNA-seq data analysis using graph neural networks. Bioinformatics. 2020;36(16):4928–35.
18. Duong CT, et al. Deep MinCut: Learning node embeddings by detecting communities. Pattern Recogn. 2023;133:109026.
19. Flake GW, Tarjan RE, Tsioutsiouliklis K. Graph clustering and Minimum Cut trees. Internet Math. 2003;1(4):385–408.
20. Wang Y et al. scBGEDA: Deep single-cell clustering analysis via a dual denoising autoencoder with bipartite graph ensemble clustering. Bioinformatics. 2023;39(2).
21. Jain AK, Murty MN, Flynn PJ. Data clustering: A review. ACM-CSUR. 1999;31(3):264–323.
22. Karypis G, Kumar V. Multilevel k-way partitioning scheme for irregular graphs. J Parallel Distrib Comput. 1999;48(1):96–129.
23. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell. 2000;22(8):888–905.
24. Hofmeyr DP. Clustering by Mincut Hyperplanes. IEEE Trans Pattern Anal Mach Intell. 2017;39(8):1547–60.
25. Von Luxburg U. A tutorial on spectral clustering. Stat Comput. 2007;17(4):395–416.
26. Ng AY, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. In Advances in Neural Information Processing Systems 14. 2002:849–856.
27. Andersen R, Chung F, Lang K. Local graph partitioning using PageRank vectors[Paper presentation]. 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), Berkeley, CA, USA. 2006.
28. Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. 2003;15(6):1373–96.
29. Wang H, Yan S, Xu D, Tang X. Trace ratio vs. ratio trace for dimensionality reduction [Paper presentation]. 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA. 2007.
30. Yan S, Tang X. Trace quotient problems revisited. In: Leonardis A, Bischof H, Pinz A, editors. Computer Vision – ECCV 2006. Volume 3952. Berlin Heidelberg: Springer; 2006. pp. 232–44.
31. Law MT, Urtasun R, Zemel RS. Deep spectral clustering learning. In: Proceedings of the 34th International Conference on Machine Learning; Sydney, NSW, Australia. 2017:985–94.
32. Bianchi FM, Grattarola D, Alippi C. Spectral clustering with graph neural networks for graph pooling. In: Proceedings of the 37th International Conference on Machine Learning;– Vienna, Austria. 2020:874–83.
33. Bresson X, Laurent T, Uminsky D, Von Brecht J. Multiclass total variation clustering. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Advances in Neural Information Processing Systems 26 (NIPS 2013). Curran Associates, Inc.; 2013:1421–29.
34. Karger, D. . "Minimum cuts in near-linear time." Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, - STOC '96. 1996:56–63.
35. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun. 2019;10(1):390.
36. Luo Z, Xu C, Zhang Z, et al. A topology-preserving dimensionality reduction method for single-cell RNA-seq data using graph autoencoder. Sci Rep. 2021;11(1):20028.
37. Petegrosso R, Li Z, Kuang R. Machine learning and statistical methods for clustering single-cell RNA-sequencing data. Brief Bioinform. 2019;21(4):1209–23.
38. Madalina, Ciortan , and Matthieu ,D. "GNN-based embedding for clustering scRNA-seq data." Bioinformatics. 2021;(4):4.
39. Wang, Juexin , et al. "scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses." Nature Communications. 2021;12(1).
40. Du J, Zhang S, Wu G, et al. Topology adaptive graph convolutional networks. arXiv. 2017. https://doi.org/10.48550/arXiv.1710.10370.
41. Abu-El-Haija S , Perozzi B , Kapoor A ,et al"MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing." Proceedings of Machine Learning Research. 2019.
42. Elhamifar E, Vidal R. Sparse subspace clustering: Algorithm, theory, and applications. IEEE Trans Pattern Anal Mach Intell. 2013;35(11):2765–81.
43. Eraslan G, Simpson JT, Jaitin DA, Teichmann SA. Single-cell RNA-seq denoising using a deep count autoencoder. Nat Commun. 2019;10(1):1–12.
44. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol. 2015;33(5):495–502.
45. Wolf FA, Angerer P, Theis FJ. Scanpy: Large-scale single-cell gene expression data analysis. Genome Med. 2018;10(1):1–11.
46. Chen M, Xiao Y, Yuan Y, Zhang X. scziDesk: A deep learning-based model for single-cell data integration. Bioinformatics. 2020;36(16):4515–21.
47. Adam M, Potter AS, Potter SS. Psychrophilic proteases dramatically reduce single-cell RNA-seq artifacts: a molecular atlas of kidney development. Development. 2017;144(19):3625–32.
48. Bach K, Pensa S, Grzelak M, Hadfield J, Adams DJ, Marioni JC, Khaled WT. Differentiation dynamics of mammary epithelial cells revealed by single-cell RNA sequencing. Nat Commun. 2017;8(1):1–11.
49. 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(5):1187–201.
50. Muraro MJ, Dharmadhikari G, Grün D, Groen N, Dielen T, Jansen E, Van Gurp L, Engelse MA, Carlotti F, De Koning EJ, et al. A single-cell transcriptome atlas of the human pancreas. Cell Syst. 2016;3(4):385–94.
51. Schaum N, Karkanas 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: The tabula muris consortium. Nature. 2018;562(7727):367–72.
52. Tosches MA, Yamawaki TM, Naumann RK, Jacobi AA, Tushev G, Laurent G. Evolution of pallium, hippocampus, and cortical cell types revealed by single-cell transcriptomics in reptiles. Science. 2018;360(6391):881–8.
53. 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. 2018;115(10):2407–12.
54. Mcinnes L, Healy J. Umap: uniform manifold approximation and projection for dimension reduction. J Open Source Softw. 2018;3(29):861202407060264.
55. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. Genome Biology. 2017;18(1):174.
56. Wan H, Chen L, Deng M. scNAME: Neighborhood contrastive clustering with ancillary mask estimation for scRNA-seq data. Bioinformatics. 2022;38(6):1701–9.

57. Chu Y, Zhang L, Jing L, et al. Single-cell transcriptomics reveals the origin and diversity of cutaneous squamous cell carcinoma. Nat Commun. 2020;11(1):1–12.

Publisher's note