

Self-supervised deep clustering of single-cell RNA-seq data to hierarchically detect rare cell populations

Tianyuan Lei, Ruoyu Chen, Shaoqiang Zhang  and Yong Chen 

Corresponding authors: Shaoqiang Zhang, College of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China. Tel.: +86 18649006931; Fax: +86 22-23766303; E-mail: zhangshaoqiang@tjnu.edu.cn; Yong Chen, Department of Biological and Biomedical Sciences, Rowan University, NJ 08028, USA. Tel.: +1 856 256 4500 ext. 53589; Fax: +1 856-256-4478; E-mail: chenyong@rowan.edu

Abstract

Single-cell RNA sequencing (scRNA-seq) is a widely used technique for characterizing individual cells and studying gene expression at the single-cell level. Clustering plays a vital role in grouping similar cells together for various downstream analyses. However, the high sparsity and dimensionality of large scRNA-seq data pose challenges to clustering performance. Although several deep learning-based clustering algorithms have been proposed, most existing clustering methods have limitations in capturing the precise distribution types of the data or fully utilizing the relationships between cells, leaving a considerable scope for improving the clustering performance, particularly in detecting rare cell populations from large scRNA-seq data. We introduce DeepScena, a novel single-cell hierarchical clustering tool that fully incorporates nonlinear dimension reduction, negative binomial-based convolutional autoencoder for data fitting, and a self-supervision model for cell similarity enhancement. In comprehensive evaluation using multiple large-scale scRNA-seq datasets, DeepScena consistently outperformed seven popular clustering tools in terms of accuracy. Notably, DeepScena exhibits high proficiency in identifying rare cell populations within large datasets that contain large numbers of clusters. When applied to scRNA-seq data of multiple myeloma cells, DeepScena successfully identified not only previously labeled large cell types but also subpopulations in CD14 monocytes, T cells and natural killer cells, respectively.

Keywords: scRNA-seq, cell clustering, deep learning, self-supervised training

INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) is a powerful technique used in bioscience that allows researchers to measure gene expression in individual cells. It enables the identification of cell-to-cell differences in gene expression, revealing the complex biology of tissues and organs [1]. scRNA-seq allows for the identification of rare cell types or subpopulations that might be missed with bulk RNA sequencing, leading to the discovery of new biological processes and disease mechanisms [2]. It has been widely used in cell type identification, disease research and drug development and study dynamic changes in the development process [3–6]. To gain a deeper understanding of the transcriptome data and the variety of cells present, performing cell clustering is the first and crucial step to identify the different cell types. Furthermore, downstream analysis of scRNA-seq data relies heavily on accurate cell typing, and the quality of the cell clustering directly impacts the accuracy of downstream results [7].

Although scRNA-seq can reveal individual cell characteristics more accurately, the data often contains higher noise and missing values than bulk RNA-seq due to the low RNA capture rate [8]. As a result, the high dimensionality, sparsity and noise of scRNA-seq data make clustering a challenging task. Several clustering methods have been developed for scRNA-seq data, such as CIDR [9], SIMLR [10] and SC3 [11]. However, hierarchical clustering, spectral clustering and k-means have limited scalability, making them unsuitable for large datasets. Seurat4 [12] and SCANPY [13] are two popular pipelines for large-scale single-cell data analysis and use graph-based community detection algorithms like Louvain [14] or Leiden [15] for clustering k-nearest neighbors graphs after dimension reduction by principal component analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) [16], or Uniform Manifold Approximation and Projection (UMAP) [17]. Besides KNN graphs, there are also clustering algorithms on shared nearest neighbors (SNN) graphs, such as SNN-Clq [18] and PhenoGraph [19]. We recently introduced SCENA [20], a novel

Tianyuan Lei is a Ph.D candidate in the College of Computer and Information Engineering, Tianjin Normal University, China. Her research interests focus on Bioinformatics and machine learning, especially for single-cell omics data analysis.

Ruoyu Chen is a senior high school student. She is interested in data analysis by using deep-learning methods.

Shaoqiang Zhang is a Professor in the College of Computer and Information Engineering, Tianjin Normal University, China. He received his Ph.D. degree in operations research from Shandong University, Shandong, China. His research interests include bioinformatics, combinatorial optimization and high-performance computing. He is a member of IEEE and The China Computer Federation.

Yong Chen is an Assistant Professor in the Department of Biological and Biomedical Sciences at Rowan University. Dr. Chen is a computational biologist and has published over 40 peer-reviewed scientific research papers. His research interests include single cell omics data analysis, Deep learning, Mathematical modeling, cancer epigenetics and neuroscience.

Received: June 9, 2023. **Revised:** September 5, 2023. **Accepted:** September 6, 2023

© The Author(s) 2023. Published by Oxford University Press.

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

clustering method that utilizes multiple feature sets, enhancements of local affinity among cells and consensus spectral clustering. Through large-scale validations, we validated its high performance and demonstrated that consensus clustering and cell similarity enhancement is effective strategies for cell clustering. Although these graph-based community detection methods can handle large-scale data, the constructed graphs may not capture proper relationships between cells [21, 22]. Moreover, linear dimensionality reduction, except UMAP, is commonly applied to scRNA-seq data before clustering, which discards the nonlinear relationship between feature genes, resulting in reduced clustering accuracy [23].

To improve the clustering accuracy, several deep clustering methods have been developed to embed high-dimensional expression data points into a low-dimensional space using autoencoders for nonlinear dimensionality reduction [24], including DESC [25], scDeepCluster [26], scDMFK [27], scziDesk [28], scAIDE [29], scGMAI [30], scCAN [31] and scDCCA [32], but all have room for improvement. For example, some methods, such as DESC [25], scDeepCluster [26] and scDMFK [27], ignore the pairwise distance or affinity between cells, which limits their abilities to learn more cluster-friendly latent spaces with high confidence. In other methods, such as scAIDE [29], scGMAI [30] and scCAN [31], the lower-dimensional representation of cells is learned separately from the clustering procedure, which uses classical clustering methods such as k-means and graph clustering. Additionally, some methods, such as scDeepCluster [26], scGMAI [30], input all genes into autoencoders without feature selection, which can be time-consuming. Some methods, such as scDeepCluster [26], scziDesk [28], introduce data imputation in their autoencoders before clustering, modelling the expression values of each gene as a zero-inflated negative binomial (ZINB) distribution and training the autoencoders by estimating the parameters of the ZINB model. A recent systematic evaluation of imputation methods on a set of benchmark scRNA-seq datasets [33] found that autoencoder-based imputation methods such as DCA [34], AutoImpute [35] and SAUCIE [36] did not essentially help clustering algorithms to improve clustering accuracy, or even reduce the clustering accuracy. In fact, recent studies also suggest that the negative binomial (NB) is more appropriate than ZINB for droplet-based scRNA-seq data, which is the current mainstream commercial platform (e.g. 10 \times genomics) [37–40].

Another challenge in clustering is determining the optimal number of clusters as it significantly affects the accuracy and usefulness of the clustering results [21, 22]. However, there is no definitive criterion for selecting the optimal number of clusters, and it can vary depending on the clustering algorithm, dataset and evaluation metrics used. Moreover, the number of clusters can vary based on the specific goals and needs of the analysis, especially in detecting rare cell types in scRNA-seq data [7, 22]. In mammals, the hierarchical organization of different cell populations is naturally observed based on lineage relationships and differentiation pathways [41, 42]. For instance, immune cell populations have distinct subpopulations of T cells, B cells and natural killer cells with specific immune functions [43]. To reflect the complex and dynamic nature of cell differentiation and specialization, an effective computational strategy is to perform hierarchical clustering in scRNA-seq, where the expression levels of genes or sets of genes can be hierarchically used as features to perform clustering. So far, most of existing methods of scRNA-seq clustering require predefined cluster numbers, but limited methods can perform hierarchical clustering (CellBIC [44], MRTree [45]). As the features (e.g. high variable genes) used for

upper-level cell populations may have poor performance in detecting lower-level cell subpopulations (rare cell types), it is important to extract different features for different levels of different cell populations. This is especially pertinent when targeting rare cell types—specific cell populations within a tissue or organism present in very low frequencies but might play critical roles in specific biological processes. Several recent scRNA-seq studies have notably contributed to uncovering rare cell types, including circulating tumor cells [46], plasma cells in bone marrow [47], neuronal subtypes [48], rare immune cell subsets [49] and developing embryonic cells [50]. These discoveries have substantially contributed to our comprehension of both normal physiological functions and disease-related processes, subsequently influencing the formulation of novel treatment strategies [51]. However, due to their rarity and the specificity of gene expression attributes, the identification and characterization of these cell types remain challenging endeavors [29, 52–54]. In light of this, we propose that the strategic application of hierarchical clustering, involving the definition of distinct features for different clustering levels, could potentially facilitate the efficient detection of rare cell types.

Based on these observations, we designed an enhanced deep-learning method called DeepScena for cell clustering by fully equipping the features of data imputation, dimensionality reduction, efficient clustering strategies of utilizing pairwise cell similarities and hierarchically clustering for detecting rare cell types. DeepScena especially employs a NB-based autoencoder by fitting the NB model to accomplish data imputation and improve accuracy. DeepScena then uses the deep clustering with self-supervision model [55] that can gradually update pairwise data similarities during training. To perform hierarchical clustering, DeepScena recalculate the feature genes for different levels in a top-down procedure. DeepScena was tested on eight large scRNA-seq datasets and results shown it can not only recall popular cell types but also delineate novel rare cell types.

MATERIALS AND METHODS

Overview of DeepScena

DeepScena is a deep clustering method for scRNA-seq data that is based on a convolutional autoencoder network and pairwise similarity enhancing networks. The framework of DeepScena is depicted in Figure 1A and consists of two modules. The first module is an NB-based convolutional autoencoder that is used for data denoising, dimension reduction and preliminary clustering. The second module is a fully connected neural network called MNet, which uses pairwise cell similarities in a reliable subspace to self-supervise the training procedure. The input layer of MNet is the encoder's output layer of the autoencoder.

Specifically, in the first module, DeepScena involves pre-training the NB model-based autoencoder before clustering, where the autoencoder maps each input cell x_i to a latent space U for dimensionality reduction, and soft assignments of each cell to clusters are given in the latent space. To capture the unique features of scRNA-seq data, an NB model-based loss function is added to the autoencoder, with NB parameterized by the mean (μ) and the dispersion (θ). Additionally, two independent full connection layers are added after the reconstruction layer to estimate the mean and dispersion. The initialization of cluster centers is obtained through standard k-means clustering in the embedded feature space of the pre-training autoencoder, using an NB distribution parameter loss function L_{NB} . Subsequently, DeepScena combines a weighted NB model-based loss function,

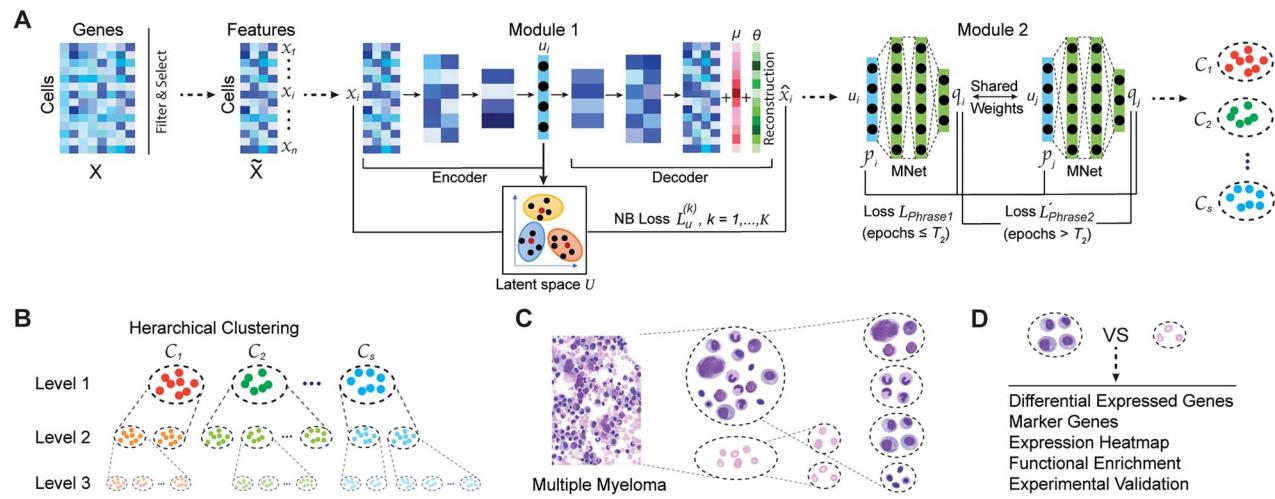


Figure 1. The workflow of DeepScena. **(A)** Encoder-decoder structures implemented in DeepScena. In the first module, we train a NB model-based denoising convolutional autoencoder to obtain a latent space U and a preliminary clustering in the latent space. The detailed specifications are shown in [Supplementary Figure S1](#). In the second module, we train MNet based on pairwise data similarities. **(B)** The top-down iterative application of DeepScena to detect cell subpopulations. **(C)** Multiple myeloma cells were hierarchically clustered by DeepScena. **(D)** Downstream analysis of cell types.

a weighted reconstruction loss function L_r , and a weighted centering loss function L_c to train the autoencoder by optimizing the latent space, cluster centers and clusters simultaneously. Moreover, the soft assignments of cells are treated as the weights of the reconstruction and centering loss functions to prevent error propagation caused by hard assignment.

In the second module, DeepScena uses a fully connected network called MNet for self-supervision. MNet is appended to the encoder part of the autoencoder trained in the first module after discarding the decoder part. The representation of each cell in the final latent space U is mapped to a trained K -dimensional space q , where K is the number of cell clusters, and the inner product is used for cell-cell similarity measurement. In this space, a pair of similar cells is very close to each other, and a pair of dissimilar cells is far away from each other. MNet strengthens similarities between two similar cells and weakens similarities between two dissimilar cells, with only similar and dissimilar pairs of cells contributing to training MNet. Consequently, the second module enhances the distinguishability of the final clusters and can better handle data with complex distributions.

To hierarchically detect cell populations, DeepScena includes a top-down loop that iteratively applies two modules to different levels of clusters. In each run of a cluster, the feature genes are reselected within the cells of the cluster, which reflects the natural differentiation linkages of mammalian cell systems ([Figure 1B](#)). As a real-case application, we used DeepScena to analyze scRNA-seq data of multiple myeloma (MM) and detect its heterogeneous cell types within the immune microenvironment under precursor stages [\[56\]](#) ([Figure 1C](#)). Finally, DeepScena was systematically designed to output analysis results for diverse downstream analysis purposes ([Figure 1D](#)).

Data preprocessing

DeepScena takes raw scRNA-seq reads count matrix as input. To remove extreme low-quality cells and/or genes, the Python package SCANPY (version 1.9.1) is applied to preprocess the raw reads count matrix X with m genes and n cells. Firstly, we filter out genes with non-zero counts in less than three cells. Secondly, we normalize each cell by total counts over all genes using a size factor of 10^4 (using the 'pp.normalize_total' function), followed by

log-transformation on the normalized matrix (using the 'pp.log1p' function). Since the matrix X is high-dimensional and sparse, both ubiquitous and low-expressed genes do not better identify and describe the cell types. Therefore, we select top t highly variable genes (using the 'pp.highly_variable_genes' function) to obtain a new matrix $X' = (x_{ij})_{n \times t}$. Then, for each cell i , we reshaped the values of the t selected genes in X' into a $r \times r$ matrix ($t = r^2$) as the input of cell i in DeepScena. Reshaping each cell into a matrix can better capture the nonlinear dependence among genes than using a single vector. Therefore, cell clustering can adopt similar methods of image clustering. Denote the preprocessed and reshaped dataset as $\tilde{X} = (x_1, x_2, \dots, x_n)$ where x_i indicates the $r \times r$ matrix of the i -th cell.

NB-based convolutional autoencoder

Given the preprocessed dataset, $\tilde{X} = (x_1, x_2, \dots, x_n)$, where each x_i represents the i -th cells, we construct a NB model-based autoencoder network to estimate NB parameters, assuming the number of clusters K is predetermined. The autoencoder network consists of an encoder and a decoder, each with two independent fully connected layers. The encoder uses three convolution layers and a fully connected layer to map \tilde{X} to a d -dimensional space (where $d \ll t$). The resulting latent representation of \tilde{X} is denoted by $U = (u_1, u_2, \dots, u_n)$. We apply soft clustering in the latent space U by training the autoencoder using weighted reconstruction, NB parameter and centering losses. The decoder then reconstructs the d -dimensional data into the original t -dimensional data through a fully connected layer and three deconvolution layers. For each cell x_i , the reconstructed output of the decoder is denoted by \hat{x}_i .

To capture the characteristics of scRNA-seq data, we incorporate a NB model-based loss function into the autoencoder. NB is parameterized with the mean μ and the dispersion θ :

$$P_{NB}([x'] | \mu, \theta) = \frac{\Gamma([x'] + \theta)}{[x']! \Gamma(\theta)} \left(\frac{\theta}{\theta + \mu} \right)^{\theta} \left(\frac{\mu}{\theta + \mu} \right)^{[x']}, \quad (1)$$

where $[x']$ represents the rounded preprocessed expression values in X' . Let D be the output of the decoder. We use an exponential

activation function for the mean and dispersion parameters since they are non-negative values.

Specifically, the NB-based loss function in the autoencoder is the negative log-likelihood of NB, which is given by $\mu = \exp(W_\mu D)$ and $\theta = \exp(W_\theta D)$, where W represents network weight parameter matrices. The NB-based loss function is given by:

$$L_{\text{NB}} = -\log(P_{\text{NB}}([x']|\mu, \theta)). \quad (2)$$

We first pretrain the autoencoder using the NB model-based loss function L_{NB} shown in Formula (2). This allows us to obtain low-dimensional representations of cells in the latent space. We then perform k -means clustering in the latent space to obtain K initial cluster centers $[u^{(1)}, u^{(2)}, \dots, u^{(K)}]$. Next, we train the autoencoder K times, once for each cluster, to make cell points more likely to belong to their true cell clusters. To achieve this, we define three loss functions. The first is for reconstruction of the convolutional autoencoder, the second is for centering of clusters and the third is for fitting NB distribution parameters.

Specifically, the loss function $L_u^{(k)}$ for the k th run is given in Formula (3). This function includes three weighted sum losses: $L_r^{(k)}$, $L_c^{(k)}$ and $L_{\text{NB}}^{(k)}$, defined by Formulas (4), (5) and (6), representing the weighted sum losses of cell reconstruction, centering and NB fitting respectively. Here we use two hyperparameters α and β to balance the three components of the loss function $L_u^{(k)}$. The level of fuzziness, λ , is set to 1.5 in all experiments.

For each cell i , we calculate the membership degree p_{ik} to the k th cluster by measuring the Euclidean distance between u_i and the cluster center $u^{(k)}$. The autoencoder's parameters are then updated by minimizing the loss function $L_u^{(k)}$. The membership degree p_{ik} is defined in Formula (7). The soft assignment of x_i is denoted by $p_i = (p_{i1}, p_{i2}, \dots, p_{iK})^T$, which is used as weights in Formulas (4), (5) and (6). Every T_1 iterations (epochs), the cluster center $u^{(k)}$ is updated to the average of weighted cell points in the latent space as defined in Formula (8). If the total number of epochs is xT_1 , then each cluster center is updated x times. Thus, DeepScena adapts its dimensionality reduction and clustering procedures with every epoch by carrying out these steps alternately. As shown in Formulas (4)–(6), the λ power of the clustering probability p_{ik} is used as a nonlinear weight in the autoencoder's reconstruction, centering and NB loss functions. This approach effectively draws cells belonging to the same cluster into closer proximity within the low-dimensional latent space.

$$L_u^{(k)} = L_r^{(k)} + \alpha L_c^{(k)} + \beta L_{\text{NB}}^{(k)} \quad (3)$$

$$L_r^{(k)} = \sum_{i=1}^n p_{ik}^\lambda \|x_i - \hat{x}_i\|^2 \quad (4)$$

$$L_c^{(k)} = \sum_{i=1}^n p_{ik}^\lambda \|u_i - u^{(k)}\|_2^2 \quad (5)$$

$$L_{\text{NB}}^{(k)} = -\sum_{i=1}^n p_{ik}^\lambda \sum_{j=1}^t \log(P_{\text{NB}}([x_{ij}]|\mu, \theta)) \quad (6)$$

$$p_{ik} = \left(\|u_i - u^{(k)}\|_2^{\frac{2}{m-1}} \right)^{-1} / \sum_{j=1}^K \left(\|u_i - u^{(j)}\|_2^{\frac{2}{m-1}} \right)^{-1} \quad (7)$$

$$u^{(k)} = \sum_{i=1}^n p_{ik}^m u_i / \sum_{i=1}^n p_{ik}^m \quad (8)$$

Self-supervision via MNet

After running the first module of DeepScena, a fully connected network MNet is attached to enhance the cell-cell similarity. The input of MNet is the latent representation of each cell in the autoencoder's latent space, and the output layer consists of K neurons, each corresponding to a cell cluster. The Softmax function is applied in the output layer to obtain probability values indicating the likelihood of a cell belonging to each of the K clusters. For an input cell x_i , the output value at the k -th neuron in the output layer, q_{ik} , represents the probability of x_i belonging to the k -th cluster. The soft assignment of x_i in MNet is denoted by $q_i = (q_{i1}, q_{i2}, \dots, q_{iK})^T$.

The training of MNet is divided into two phases. In the first phase, MNet is randomly initialized. Within T_2 training iterations (epochs), the assignments p_1, p_2, \dots, p_n , in the space U are used to train q_1, q_2, \dots, q_n with the loss function (9). Here δ ($0 < \delta < 1$) is a hyperparameter used to identify cell pairs with similarity higher than δ or lower than $1 - \delta$, and $\mathbb{I}[\cdot]$ is the indicator function. Typically, δ is set to a decimal fraction less than 1, such as $\delta = 0.8$, so that similarity scores between $1 - \delta = 0.2$ and $\delta = 0.8$ are not used to train the MNet to avoid inconclusive clustering. In each of the T_2 iteration, the cluster centers $u^{(k)}, k = 1, \dots, K$, are updated again using formula (8), and a relatively reliable space q is obtained after T_2 iterations. The MNet is then fine-tuned for a number of epochs using the loss function L_{Phase2} defined in (10) in the second phase.

The self-supervision approach allows the adoption of self-defined pseudo labels as a form of supervision, where the cells exhibiting high similarity should be classified into the same type, while cells demonstrating significant dissimilarity should be categorized into different types. This curated supervised information is then used to enhance the cell similarity measurement. The utilization of self-supervised information contributes to an improvement in clustering accuracy by further refining the model's ability to discern cell types based on their intrinsic similarities.

$$L_{\text{Phase1}} = \sum_{x_i, x_j \in \tilde{X}} (\mathbb{I}[p_i^T p_j \geq \delta] (1 - q_i^T q_j) + \mathbb{I}[p_i^T p_j \leq (1 - \delta)] (q_i^T q_j)) \quad (9)$$

$$L_{\text{Phase2}} = \sum_{x_i, x_j \in \tilde{X}} (\mathbb{I}[q_i^T q_j \geq \delta] (1 - q_i^T q_j) + \mathbb{I}[q_i^T q_j \leq (1 - \delta)] (q_i^T q_j)) \quad (10)$$

Parameter settings for networks

The proposed autoencoder employs an asymmetric design with a bottleneck layer following each convolutional layer in the encoder (see specific structure and specification in [Supplementary Figure S1](#)). When a dataset contains $m \geq 2000$ highly variable genes identified by SCANPY, the input number of such genes is set as $r^2 = 28^2$; for datasets with $m < 2000$ highly variable genes, the input number is set as $r^2 = 16^2$. MNet's input and output stem from the d -dimensional latent space U and K -dimensional space q , respectively. The fully connected network architecture of MNet for all datasets is $d - 128 - 128 - 128 - K$. The autoencoder and MNet weights are updated with the Adam optimizer [57], utilizing the learning rate of 10^{-6} , respectively. DeepScena utilizes six hyperparameters $\alpha, \beta, \lambda, \delta, T_1$ and T_2 , which remain constant across all scRNA-seq datasets. Similar to DCSS [55], the value of hyperparameter α that indicates the importance of the centering loss $L_c^{(k)}$ in the loss function (3) is set to 0.1, the level of fuzziness λ is set to 1.5, and the similarity cutoff δ is also set to 0.8 in DeepScena. The other three hyperparameters β, T_1 and T_2 , will be discussed in the 'Results' section.

Datasets and evaluation metrics

To evaluate the performance of DeepScena, we selected scRNA-seq datasets with large sizes (e.g. from thousands to tens of thousands of cells) and mainstream sequencing protocols (Supplementary Table S1). Briefly, the PBMC 4K dataset, which contains the transcriptome of 4340 peripheral blood mononuclear cells (PBMCs) from a healthy donor, and a dataset (referred to as 'Hgmm'), which is 1:1 mixture of fresh frozen human HEK293T and mouse NIH3T3 cells, were downloaded from the 10 \times Genomics website. The scRNA-seq dataset of pancreatic islets from four human donors (referred to as Baron dataset) was downloaded from NCBI GEO database with access ID GSE84133. We also downloaded four additional datasets sequenced from mouse prefrontal cortex (referred to as Bhattacherjee dataset [58]), visual cortex (referred to as Tasic dataset [59]) and cerebral cortex (referred to as Zeisel dataset [60]), arcuate hypothalamus and median eminence (referred to as Campbell dataset) with access IDs of GSE124952, GSE115746, GSE60361 and GSE93374, respectively. Another real dataset for detecting rare cell types is multiple myeloma (MM) that was downloaded with access ID of GSE124310. After quality control, a total of 5541 cells were extracted from seven MM patients for downstream clustering analysis (referred to as Zavidij dataset). The top 25 differentially expressed genes (DEGs) were detected and plotted by using SCANPY [13].

Two standard metrics, normalized mutual information (NMI) and adjusted Rand index (ARI), were used to evaluate clustering performance as we did in SCENA [20]. Both have a maximum value of 1, with higher scores indicating better clustering performance. ARI and NMI are defined in (11) and (12), respectively, and are calculated based on a predicted clustering $X = (X_1, X_2, \dots, X_r)$ and a true partition $Y = (Y_1, Y_2, \dots, Y_s)$. Here, n represents the number of cells, n_{ij} represents the number of cells in true partition Y_j assigned to predicted cluster X_i . n_i and n_j represent the number of cells in X_i and Y_j , respectively.

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2} \right] / \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2} \right] - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2} \right] / \binom{n}{2}} \quad (11)$$

$$\text{NMI} = \frac{\sum_i \sum_j n_{ij} \log \frac{n_{ij}}{n_i n_j}}{\max \left(-\sum_i n_i \log \frac{n_i}{n}, -\sum_j n_j \log \frac{n_j}{n} \right)} \quad (12)$$

Comparison with other methods

For further benchmarking the performance of DeepScena, we conducted a comparative analysis against seven recently developed unsupervised clustering algorithms: SCENA [20], scDeep-Cluster [26], sczIDesk [28], scAIDE [29], scGMAI [30], scCAN [31] and scDCCA [32]. Our selection of these tools was driven by specific factors. For instance, SCENA [20] has demonstrated its superiority over SC3 [11], Seurat4 [12], SCANPY [13], pcaReduce [61] and SNN-cliq [18]. Similarly, scAIDE claims to be superior to scVI [62], scScope [63], SC3 [11], CIDR [9] and SIMLR [10]. Meanwhile, sczIDesk [28] assets its advantage over to CIDR [9] and SIMLR [10]. We specifically selected two newly published methods scCAN [31] and scDCCA [32] that also use deep learning techniques. All tools were applied to the eight datasets (Supplementary Table S1) by using the parameters recommended by their authors, with the cluster numbers set as them in their original studies or automatically determined by built-in methods. This rigorous comparative analysis aims to evaluate the strengths and weaknesses of these algorithms in diverse clustering scenarios.

We have run all programs on a Linux workstation (CPU: Intel Xeon E5-2620/2.10GHz/8 cores) with a GPU (Nvidia GTX 1080Ti). To handle large-scale scRNA-seq data, DeepScena was developed by using the PyTorch framework, enabling CPU/GPU parallel computing. When GPU are available, it utilizes CUDA implementations for its main functions. The running instructions and codes of these eight tools are available on <https://github.com/shaoqiangzhang/DeepScena>.

RESULTS

The NB-based autoencoder is better than a regular autoencoder

Firstly, we tested the performance of the first module with different values of the hyperparameter β on various datasets. In DeepScena, the hyperparameter β indicates the importance of the NB-based loss $L_{\text{NB}}^{(k)}$ in the loss function (3). We ran the first module with different $\beta \in (0, 0.01, 0.1, 1)$ and $T_1 = 2$ on the Bhattacherjee, Tasic, PBMC4K and Zeisel datasets (Supplementary Figure S2A). The performance when $\beta = 0.1$ is more outstanding than the other β values, particularly than the regular autoencoder without the NB-based loss (i.e. $\beta = 0$). This indicates that the loss function containing a certain proportion of NB-based loss is more suitable for clustering scRNA-seq data than that without NB-based loss. Additionally, this shows that NB-based convolutional autoencoder has a certain imputation effect on these scRNA-seq datasets. We set $\beta = 0.1$ as the default value for all datasets.

In the first module of DeepScena, the cluster centers need to be updated every T_1 iterations. We tested the clustering performance of the first module with different intervals $T_1 \in (1, 2, 5, 10, 20)$ on each dataset with the number of epochs increasing. Results on four datasets show that a smaller value of T_1 achieves better clustering performance (Supplementary Figure S2B). To maintain higher performance while reducing training time, we have set $T_1 = 2$ for all datasets. Furthermore, we observed that for $T_1 = 2$, the performance of each experiment can achieve a relatively stable state after 20 epochs (Supplementary Figure S2B). Therefore, we set the default number of epochs in the first module of DeepScena to 20.

The MNet module improves clustering performance

To evaluate the performance of the second module, we first tested its results with different values of hyperparameter T_2 on the Bhattacherjee, Tasic, PBMC4K and Zeisel datasets. Specifically, we tested $T_2 \in (1, 5, 10, 15, 20)$ and observed that, as the number of epochs increased, smaller values of T_2 led to faster and more stable ARI scores. While larger values of T_2 required more epochs and longer running times without much improvement in performance (Supplementary Figure S2C). Therefore, selecting T_2 between 1 and 5 can result in high-performance clustering. In our experiments, we selected $T_2 = 1$ as the optimal value for all datasets and set the number of epochs for the second module to 20. As the total number of epochs for the two modules is 40, DeepScena proves to be time effective.

We also compared the clustering performance of the first module (NB-based autoencoder) and the second module (MNet) by running each module 10 times on each dataset and recording the ARI and NMI scores. We then used these scores to generate boxplots, which are shown in Figure 2A and B. Averagely, 5–10% increments of ARI and NMI scores were observed for the eight datasets. The results demonstrate that DeepScena with both modules outperforms the first module alone for all eight datasets. Furthermore, the clustering results of ARI and NMI

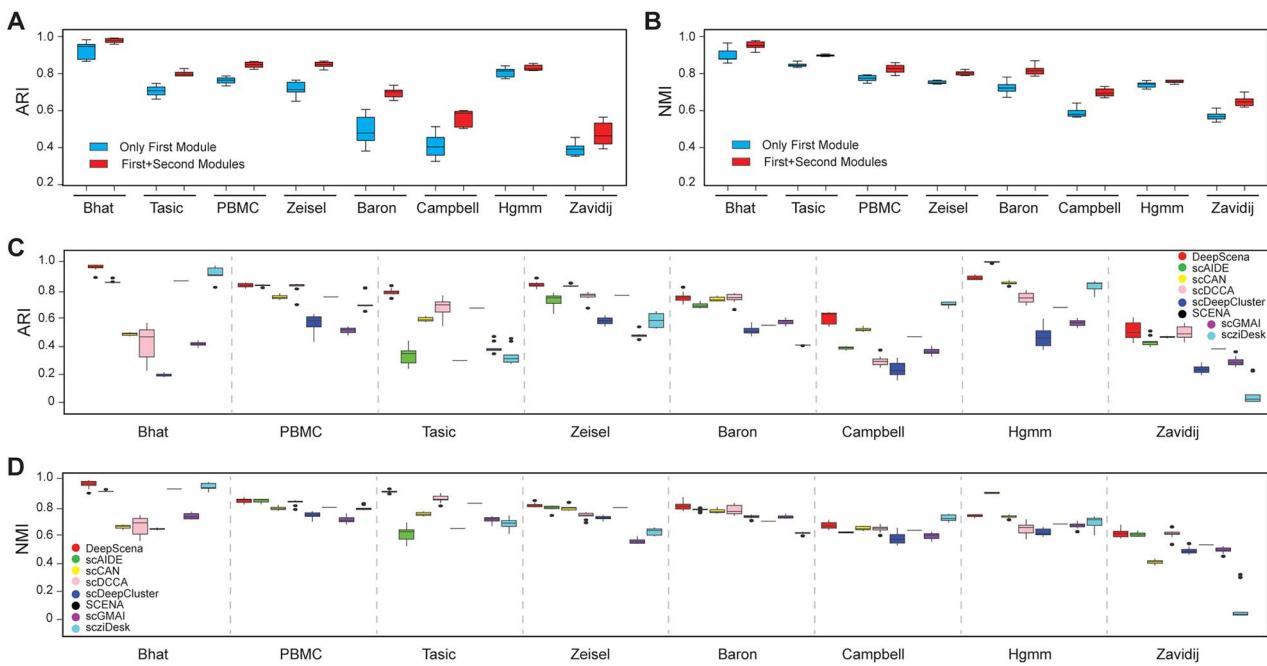


Figure 2. Performance and comparison of DeepScena. **(A)** ARI and **(B)** NMI comparison between only the first module (NB-based autoencoder) and the two modules (NB-based autoencoder + MNet) for each dataset. **(C)** ARI comparison of eight clustering tools. **(D)** NMI comparison of eight clustering tools.

obtained after MNet applied have smaller variances than those of the first module in seven datasets (Figure 2A and B), suggesting that MNet can improve the probability of similar cells belonging to the same cluster while reducing the likelihood of dissimilar cells being grouped together. Thus, implementing the second module is necessary for improving the clustering performance of DeepScena.

Performance comparison of DeepScena and other seven methods

Here, we aimed to compare DeepScena with other seven popular methods that are recently designed for unsupervised clustering. For each of the eight clustering methods, we generated box-plots of ARI scores by applying it to each dataset ten times. The results show that the ARI scores of DeepScena consistently remain high across all eight datasets (Figure 2C, Supplementary Table S2). In details, DeepScena excels in six datasets and secures the second rank in the remaining two datasets (Campbell and Hgmm). In comparison, the performance rankings of the other seven methods are on average lower and display greater variances than DeepScena across eight datasets. Specifically, DeepScena was compared with two newly published methods, scCAN [31] and scDCCA [32], both of which also incorporate deep learning-based techniques. Results show that while scCAN and scDCCA demonstrate commendable performance across these datasets, their ARI and NMI scores do not match those achieved by DeepScena. In general, scDCCA emerges as either the second or third best performer across most datasets; however, its performance stability is noticeably compromised, characterized by large standard deviations. The results are similarly observed in NMI metric (Figure 2D, Supplementary Table S3).

As the cluster numbers of the eight datasets range from 8 to 23 (Supplementary Table S1), we further checked if DeepScena can work well for datasets with large cluster numbers, where the data may include more heterogenous and rare cell types. We observed

DeepScena can achieve good performance of ARI and NMI on those datasets with large cluster numbers. For example, on three datasets Tasic dataset (23 clusters), Campbell dataset (20 clusters) and Baron dataset (14 clusters), DeepScena achieved ARI scores of 0.7952, 0.5632 and 0.7021, respectively, and NMI scores of 0.8935, 0.6916 and 0.8111, respectively. DeepScena outperformed scAIDE, which was designed for clustering rare cell types. The main reason for DeepScena's high performance is that the newly constructed latent space is better at separating different cell clusters. For example, Tasic dataset has 23 clusters and four of them each have less than 100 cells that are annotated in authors original annotation. However, the authors clustering annotations show multiple clusters aggregate each other (Supplementary Figure S3A), while they are separated by using the reconstructed latent space in DeepScena (Supplementary Figure S3B).

Robustness against dropout noise

In scRNA-seq, dropout refers to the phenomenon where a gene appears to have zero or very low expression in a single cell, even though it is known to be expressed in other cells or bulk tissue samples. Although much progress has been made in scRNA-seq data, different dropout levels are still frequently observed, largely due to technical limitations such as the low amount of RNA in single cells and the low sensitivity of detection methods. It is important to evaluate the performance of computational methods in accurately recovering cell-type populations at various dropout rates. To this end, we generated simulations by randomly reassigning certain proportions of genes as zeros. Among the eight datasets used in this study, five have high original dropout rates of over 90% (Supplementary Table S1). Therefore, we performed the above simulation procedures for three datasets that have lower dropout rates, less than 90% and increased the dropout rates to 90% (Zeisel and Tasic datasets) and 95% (Bhattacherjee), respectively. The simulation was performed 10 times for each dropout rate level, and performance indices were plotted for six

methods including the newly published deep learning-based methods scCAN and scDCCA. The results show that, in each dataset, the ARI scores of DeepScena outperform those of the other five methods under different dropout rates (Figure 3). Particularly, when the dropout levels increased from 85.7 to 90% for the Zeisel dataset, the average performance of DeepScena on Zeisel remained stable, but only the variances increased slightly (Figure 3A). We further noticed that DeepScena maintained fair performance on the Tasic (Figure 3B) and Bhattacherjee (Figure 3C) datasets when the dropout levels increased from ~80 to 85%. Overall, the results show that DeepScena has high potential and super performance in separating and delineating cell groups, regardless of high dropout levels.

Hierarchically detecting rare cell subpopulations

Unsupervised clustering analysis is an important practical strategy for detecting rare cell types and providing biological insights into cancer pathology. However, there are two technical challenges in unsupervised clustering: defining the cluster numbers of cell types and retaining high performance for low cell numbers of rare subpopulations. Clustering cells hierarchically is one strategy to overcome these limitations. Classical hierarchical clustering methods use cell-to-cell distance to reconstruct a tree structure, which is highly affected by the different choice of an appropriate metric and linkage criterion. Additionally, the cell-to-cell distance is usually calculated using a fixed gene group, such as high variable genes selected by dimension reduction approaches, which may be highly biased towards regular and large cell types but not suitable for rare cell types. Considering the complex and dynamic nature of cell differentiation and specialization, it is reasonable to perform hierarchical clustering by iteratively reselecting feature genes at different cell levels.

To implement this strategy, DeepScena was iteratively applied in a top-down process to detect rare cell types. We used Bhattacherjee dataset as a detailed case study. The dataset was obtained from the prefrontal cortex (PFC) cells of 12 mice, including a total of 24 822 PFC cells merged from saline (11 886 cells) and cocaine (12 936 cells) condition groups [58]. First, we applied DeepScena to all cells and detected 10 clusters (Figure 4A), which recovered major cell types in the original analysis using the Seurat pipeline [58, 64]. For each cell type, we predicted the top 25 DEG genes (Supplementary Figure S4A) and checked their cell-specific expression profiles. The expression of the top 2 DEGs showed that most of them are only expressed in one or two cell types (Supplementary Figure S4B). We also observed that DeepScena could delineate another neuron type among the excitatory neurons that were clustered together in both analyses. We further applied DeepScena to these neurons and found eight subpopulations (Figure 4B), including layer 2/3 pyramidal neurons, layer 5 pyramidal neurons, layer 6 pyramidal neurons and layer 4 spiny stellate neurons. After predicting the DEGs for these cell types (Supplementary Figure S5A and B), we found that some of these genes have been reported to be differentially expressed in certain layers or cell types. For instance, Bcl11b is highly expressed in layer 5 neurons, including pyramidal neurons [65], and Rab3c is more highly expressed in layer 2/3 than in layer 5 [66]. Calb1 (calbindin 1) is commonly used as a marker for a subpopulation of layer 2/3 pyramidal neurons that project to subcortical structures [67]. The clustering result of the second iteration further demonstrates that DeepScena can well reselect the feature gene sets to separate cell types. We then applied DeepScena to the largest group of layer 2/3 pyramidal neurons and found they could be clustered into seven further

subpopulations (Figure 4C). However, we also found that these seven clusters were aggregated together in UMAP visualization, suggesting that the highly variable genes have no significant difference to separate them well. After checking the top-ranked DEGs for each cell cluster (Supplementary Figure S6A), we noticed that although the top-ranked DEGs have a cell-specific expression pattern (Supplementary Figure S6B), they are not as specific as upper-level DEG genes as shown in Supplementary Figures S4B and S5B. Thus, the iteration of hierarchical clustering can be practically terminated under such a case. Furthermore, most of the marker genes at different levels are different (Supplementary Figures S4B, S5B, and S6B), supporting our technical hypothesis that different features are associated with different levels of different cell populations.

Detecting rare cell subpopulations in multiple myeloma

To demonstrate the performance of DeepScena in real data analysis, we applied it to scRNA-seq data from multiple myeloma (MM) patients. MM is a hematological malignancy of plasma cells characterized by extensive tumor heterogeneity, making it largely incurable [68]. To better understand the clonal complexity of tumor cells and immune microenvironment, several scRNA-seq research projects have been performed to delineate ongoing tumor dynamics and improve molecular stratification in patients with MM [56, 69]. Here, we hypothesize that DeepScena can improve data analysis by accurately dissecting diverse cell types within the MM tumor microenvironment. We applied DeepScena to the 5541 cells of seven newly diagnosed MM patients [56]. Our results show that the transformed latent space in DeepScena improved performance in separating potential cell clusters compared with author's original labeling. As shown in Figure 5A, the author's labels of 10 cell types are separated into several small clusters, particularly for CD14 monocytes, T cells and natural killer (NK) cells. For instance, CD14 monocytes are distributed into three individual groups, T cells are in four groups and NK cells are in three groups, suggesting that these three cell types can be further clustered into cell subpopulations.

To discover those potential subpopulations, we applied DeepScena to the raw data to obtain a total of 15 clusters, which not only recalled the authors original annotations (Figure 5A) but also well separated CD14 monocytes, T cells and NK cells, respectively (Figure 5B). To elucidate the potential functional characteristics of different cell types, we conducted analysis of top 25 DEGs for each cluster (Supplementary Figure S7) and particularly emphasized the investigation of the 15 most highly ranked DEGs corresponding to each of the 15 cell types. Remarkably, many of these DEGs are widely recognized markers for immune cells (Figure 5C). For instance, CD79A is specifically expressed in B cells and pre-B cells (cluster 1 and 7 in Figure 5B,C), where it plays a crucial role in B cell activation and the initiation of the humoral immune response [70]. This protein is commonly used as a marker for identifying B cells in various applications, such as flow cytometry, immunohistochemistry and immunofluorescence [71, 72]. Similarly, IL7R is a marker of T cells, B cells and NK cells, and is often used in conjunction with other markers to distinguish between different subsets of T cells [73]. MNDA, a myeloid cell nuclear differentiation antigen, is primarily expressed in the nucleus of myeloid cells, including monocytes, macrophages and granulocytes, where it regulates gene expression and differentiation [74]. LST1, a leukocyte-specific transcript 1, is also a marker of various immune cells, including monocytes, macrophages and neutrophils, and is involved in regulating various cellular

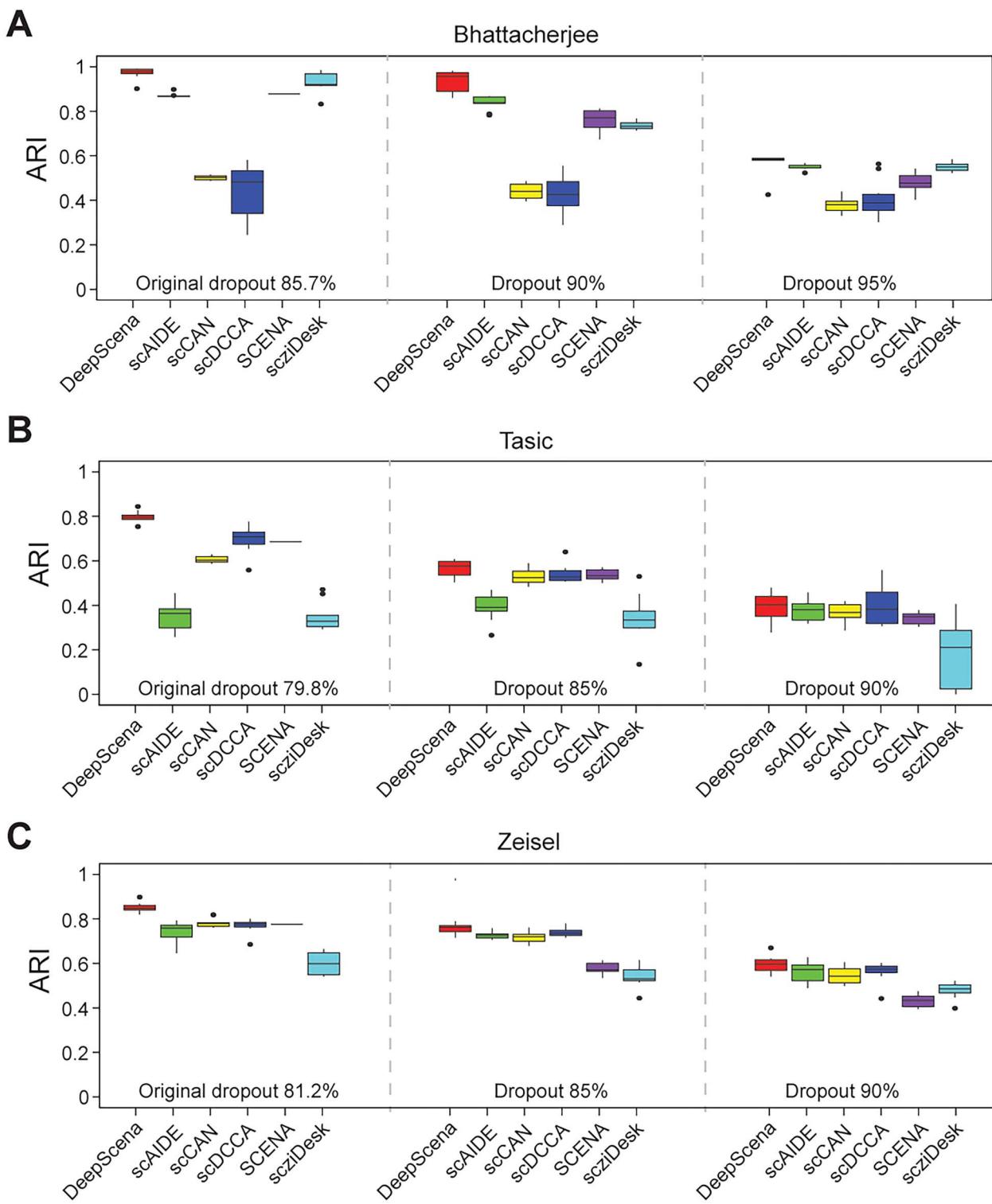


Figure 3. Performance of six tools on different dropout rates. The ARIs were calculated for three datasets, Bhattacherjee (A), Tasic (B) and Zeisel (C).

processes such as phagocytosis, cytokine production and antigen presentation [75]. CTSW, a protease enzyme, is primarily expressed in cytotoxic T cells, where it contributes to the cleavage and degradation of target cells [76, 77]. Although it has also been found to be expressed in other cell types, such as natural killer cells, mast cells and dendritic cells, its expression in these cells is generally lower than in cytotoxic T cells. Moreover, we found that although other genes were not marker genes, they were specifically expressed in different types of B cells or T cells and

implicated in various biological processes, providing insight into the differentiation and activation states of immune cells in MM pathology.

DISCUSSION

In this study, a novel deep-learning clustering method called DeepScena was designed for scRNA-seq data analysis that provides accurate results for detecting rare cell populations.

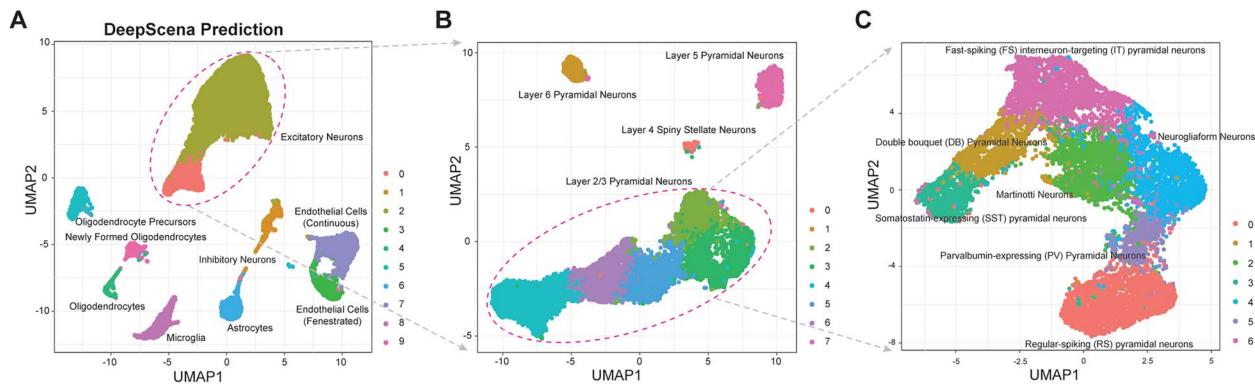


Figure 4. DeepScena can hierarchically detect rare cell populations. (A) DeepScena detected ten cell types at the top level. (B) Cluster the excitatory neurons into eight subpopulations. (C) Cluster the layer 2/3 pyramidal neurons into seven subpopulations.

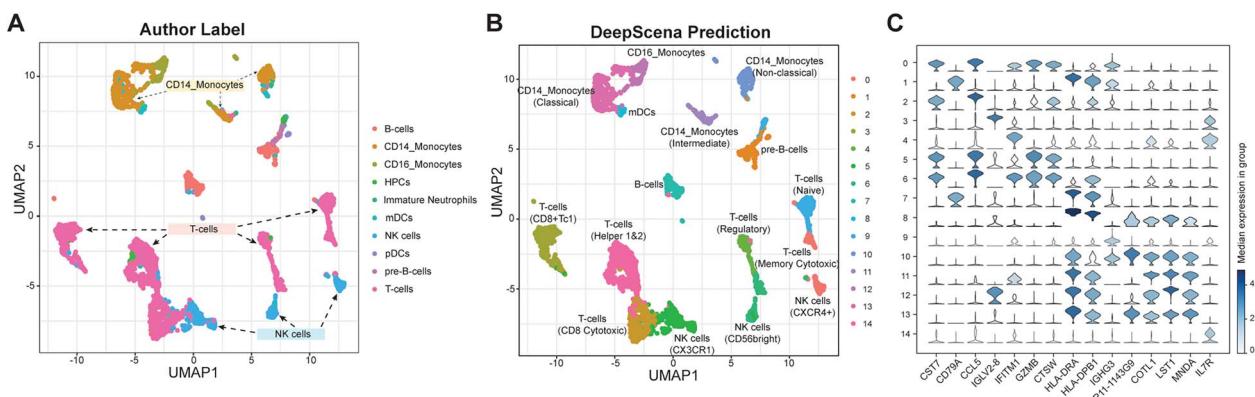


Figure 5. DeepScena separates different cell types in MM. (A) Author's labels on latent space. (B) DeepScena's prediction of fifteen cell types. (C) The violin plots for top-one ranked marker genes of each cell type.

The method was fully equipped with features such as data imputation, dimensionality reduction, enhanced pairwise cell similarities, and efficient clustering strategies. A key feature of DeepScena is the use of a negative binomial-based autoencoder to fit the NB model for data imputation, which enhances accuracy. Additionally, DeepScena uses paired data similarity as a self-supervised means to capture cell relationships and obtain a cluster-friendly space for efficient aggregation of similar cells. The study found that the NB-based autoencoder in DeepScena outperformed a regular autoencoder in terms of clustering scRNA-seq data. DeepScena was tested on eight expansive scRNA-seq datasets, yielding results that its ARI/NMI scores excel in six datasets and secure the second rank in the remaining two datasets (Campbell and Hgmm). One potential explanation for DeepScena not consistently outperforming other models could be its uniform utilization of hyperparameter settings across all datasets. This approach, however, in turn mitigates overfitting issues commonly observed in methods trained exclusively on specific datasets [78]. This notion is supported by the fact that the performance rankings of the other seven methods average lower and exhibit greater variances than DeepScena across the eight datasets.

The application of DeepScena to MM scRNA-seq datasets facilitated the identification of discrete subpopulations within CD14 monocytes, T cells and natural killer (NK) cells. This discovery holds critical implications for enhancing our understanding of immune cell diversity and its role in MM development. Through the exploration of gene expression profiles within these subpopulations, we gain insights into their distinctive functions within the

immune system. For instance, the specific expression of CD79A in B cells and IL7R across T cells, B cells and NK cells underscores their roles as essential markers for immune cell identification and differentiation. Similarly, the presence of MNDA and LST1 in myeloid cells provides insights into the regulation of gene expression and differentiation in monocytes, macrophages and granulocytes. Moreover, the identification of CTSW's specific expression in cytotoxic T cells and its lower expression in other immune cells like natural killer cells provides a nuanced understanding of its involvement in target cell degradation. These findings not only enrich our knowledge of immune cell subpopulations but also offer potential avenues for investigating their distinct functions and contributions to immune responses.

As the features used to define large cell populations are different with those used for subpopulations within one cell group, the DeepScena method can be run iteratively on these clusters to further detect rare cells, thus partially resolved the question of determining optimal cluster numbers. However, determining the number of clusters is challenging in general as there is no clear objective criterion for selecting the optimal number of clusters. A common approach is to use metrics such as the silhouette score [79] or the gap statistic to evaluate the quality of the clustering results for different numbers of clusters. However, these metrics are not always reliable and may not accurately reflect the underlying biological structure of the data. Additionally, the choice of clustering algorithm and the choice of input parameters can have a significant impact on the clustering results, further complicating the task of determining the optimal number of clusters. To address these challenges, one approach is to use a consensus

clustering approach, where multiple clustering algorithms and input parameters are used to generate multiple clustering solutions. These solutions are then combined to identify a consensus solution that is more robust to the variability and noise in the data. Another approach is to use dimensionality reduction techniques such as PCA or t-SNE or UMAP to visualize the data and manually identify distinct subpopulations of cells. For example, we can visualize the distribution of cells by using the UMAP [17] method to reduce the input matrix to two dimensions and observe the number of cell blocks as the number of preliminary clusters. Furthermore, users can run DeepScena on different clusters numbers and choose the optimized numbers with best ARI and/or NMI scores. Trying different cluster numbers can lead to novel biological discoveries, as this empirical procedure is important for detecting rare cell types that we have no prior information about.

In summary, we have demonstrated that DeepScena performs exceptionally well in detecting rare cell types, thereby revealing important information about their characteristics and functions. The ability to detect and study these rare cell types using scRNA-seq has tremendous potential to provide new insights into the biology of various tissues and diseases, which could eventually lead to the development of novel treatments and therapies. Nonetheless, there remains room for further enhancing the precision and applicability of DeepScena. For instance, we can expand DeepScena's scope to integrate diverse single-cell omics datasets, such as scATAC-seq [80] or scMethyl-seq [81], to achieve a comprehensive understanding of cellular heterogeneity and epigenetic regulation. Additionally, we can explore transfer learning techniques [82], leveraging pre-trained models on extensive datasets and adapting them to datasets with limited samples, thus enhancing DeepScena's generalizability. Furthermore, we can extend DeepScena's capabilities to accommodate bulk RNA-seq data and spatial transcriptomics data [83], enabling a more comprehensive appreciation of cellular diversity within tissues [84]. By addressing these research topics and challenges, DeepScena could evolve into a more versatile and comprehensive tool for deep-learning-based clustering analysis of scRNA-seq data, with applications spanning across various biological contexts.

Key Points

- DeepScena is fully equipped with multiple deep-learning-based technical features for high performance of hierarchical clustering analysis.
- DeepScena enables efficient autoencoders by incorporating data-specific distributions (e.g. negative binomial).
- DeepScena is robust against high dropout noise levels and can hierarchically identify rare cell types from large scRNA-seq datasets.

SUPPLEMENTARY DATA

Supplementary data are available online at <https://academic.oup.com/bib>.

ACKNOWLEDGEMENTS

We would like to thank Prof. Manoj K. Pandey from Cooper Medical School of Rowan University for critically reviewing and providing feedback on the data analysis of Multiple Myeloma.

FUNDING

This work was supported by the NSF CAREER Award DBI-2239350 and the W. W. Smith Charitable Trust grant (C2204) for Y.C.; a key project of Natural Science Foundation of Tianjin City (19JCZDJC35100) and the National Science Foundation of China (61572358) to S.Z.

AUTHORS CONTRIBUTIONS

Shaoqiang Zhang and Yong Chen initiated the concept and supervised the study. Shaoqiang Zhang and Yong Chen designed the methodology. Tianyuan Lei and Shaoqiang Zhang implemented the software. Tianyuan Lei, Ruoyu Chen, Shaoqiang Zhang and Yong Chen performed the data analysis. Shaoqiang Zhang and Yong Chen drafted and reviewed the paper. All authors have read and approved the final manuscript.

DATA AVAILABILITY

All the scRNA-seq data is available from the NCBI GEO database or the 10x website, and the access IDs are listed in [Supplementary Table S1](#). The Python code for DeepScena, demo examples, and running instructions for comparative analysis of six tools are available at <https://github.com/shaoqiangzhang/DeepScena>. Additionally, all the code materials for DeepScena have been deposited to Zenodo, and the corresponding DOI is <https://doi.org/10.5281/zenodo.8010774>.

REFERENCES

1. Olsen TK, Baryawno N. Introduction to single-cell RNA sequencing. *Curr Protoc Mol Biol* 2018;**122**(1):e57.
2. Grün D, Lyubimova A, Kester L, et al. Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature* 2015;**525**(7568):251–5.
3. Li L, Dong J, Yan L, et al. Single-cell RNA-Seq analysis maps development of human germline cells and gonadal niche interactions. *Cell Stem Cell* 2017;**20**(6):858–873.e4.
4. Zhang Y, Wang D, Peng M, et al. Single-cell RNA sequencing in cancer research. *J Exp Clin Cancer Res* 2021;**40**(1):81.
5. Sardoo AM, Zhang S, Ferraro TN, et al. Decoding brain memory formation by single-cell RNA sequencing. *Brief Bioinform* 2022;**23**(6):1–15.
6. Mathys H, Davila-Velderrain J, Peng Z, et al. Single-cell transcriptomic analysis of Alzheimer's disease. *Nature* 2019;**570**(7761):332–7.
7. Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat Rev Genet* 2019;**20**(5):273–82.
8. Haque A, Engel J, Teichmann SA, Lönnberg T. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Med* 2017;**9**(1):75.
9. Lin P, Troup M, Ho JW. CIDR: ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biol* 2017;**18**(1):59.
10. Wang B, Zhu J, Pierson E, et al. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Methods* 2017;**14**(4):414–6.
11. Kiselev VY, Kirschner K, Schaub MT, et al. SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**(5):483–6.
12. Stuart T, Butler A, Hoffman P, et al. Comprehensive integration of single-cell data. *Cell* 2019;**177**(7):1888–1902.e21.

13. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* 2018; **19**(1):15.
14. Subelj L, Bajec M. Unfolding communities in large complex networks: combining defensive and offensive label propagation for core extraction. *Phys Rev E Stat Nonlin Soft Matter Phys* 2011; **83** (3 Pt 2):036103.
15. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* 2019; **9**(1):5233.
16. Van der Maaten L, Hinton G. Visualizing data using t-SNE. *Journal of machine learning research* 2008; **9**(11):2579–2605.
17. McInnes L, Healy J, Melville J. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software* 2018; **3**(29):861. <https://doi.org/10.21105/joss.00861>.
18. Xu C, Su Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics (Oxford, England)* 2015; **31**(12):1974–80.
19. Levine JH, Simonds EF, Bendall SC, et al. Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell* 2015; **162**(1):184–97.
20. Cui Y, Zhang S, Liang Y, et al. Consensus clustering of single-cell RNA-seq data by enhancing network affinity. *Brief Bioinform* 2021; **22**(6):bbab236.
21. Yu L, Cao Y, Yang JYH, Yang P. Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data. *Genome Biol* 2022; **23**(1):49.
22. Zhang S, Li X, Lin J, et al. Review of single-cell RNA-seq data clustering for cell-type identification and characterization. *RNA* 2023; **29**(5):517–30.
23. Flores M, Liu Z, Zhang T, et al. Deep learning tackles single-cell analysis—a survey of deep learning for scRNA-seq analysis. *Brief Bioinform* 2022; **23**(1):bbab531.
24. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science* 2006; **313**(5786):504–7.
25. Li X, Wang K, Lyu Y, et al. Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nat Commun* 2020; **11**(1):2338.
26. 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**(4):191–8.
27. Chen L, Wang W, Zhai Y, Deng M. Single-cell transcriptome data clustering via multinomial Modeling and adaptive fuzzy K-means algorithm. *Front Genet* 2020; **11**:295.
28. Chen L, Wang W, Zhai Y, Deng M. Deep soft K-means clustering with self-training for single-cell RNA sequence data. *NAR genomics and bioinformatics* 2020; **2**(2):lqaa039.
29. Xie K, Huang Y, Zeng F, et al. scAIDE: clustering of large-scale single-cell RNA-seq data reveals putative and rare cell types. *NAR Genom Bioinform* 2020; **2**(4):lqaa082.
30. Yu B, Chen C, Qi R, et al. scGMAI: a Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder. *Brief Bioinform* 2021; **22**(4):bbaa316.
31. Tran B, Tran D, Nguyen H, et al. scCAN: single-cell clustering using autoencoder and network fusion. *Sci Rep* 2022; **12**(1):10267.
32. Wang J, Xia J, Wang H, et al. scDCCA: deep contrastive clustering for single-cell RNA-seq data based on auto-encoder network. *Brief Bioinform* 2023; **24**(1):bbac625.
33. Hou W, Ji Z, Ji H, Hicks SC. A systematic evaluation of single-cell RNA-sequencing imputation methods. *Genome Biol* 2020; **21**(1):218.
34. Eraslan G, Simon LM, Mircea M, et al. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun* 2019; **10**(1):390.
35. Talwar D, Mongia A, Sengupta D, Majumdar A. AutoImpute: autoencoder based imputation of single-cell RNA-seq data. *Sci Rep* 2018; **8**(1):16329.
36. Amodio M, van Dijk D, Srinivasan K, et al. Exploring single-cell data with deep multitasking neural networks. *Nat Methods* 2019; **16**(11):1139–45.
37. Svensson V. Droplet scRNA-seq is not zero-inflated. *Nat Biotechnol* 2020; **38**(2):147–50.
38. Chen W, Li Y, Easton J, et al. UMI-count modeling and differential expression analysis for single-cell RNA sequencing. *Genome Biol* 2018; **19**(1):70.
39. Andrews TS, Hemberg M. M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics (Oxford, England)* 2019; **35**(16):2865–7.
40. Tang W, Bertaux F, Thomas P, et al. bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data. *Bioinformatics (Oxford, England)* 2020; **36**(4):1174–81.
41. Derenyi I, Szollosi GJ. Hierarchical tissue organization as a general mechanism to limit the accumulation of somatic mutations. *Nat Commun* 2017; **8**:14545.
42. Feliciangeli F, Dreiji H, López-García M, et al. Why are cell populations maintained via multiple compartments? *J R Soc Interface* 2022; **19**(196):20220629.
43. Fang P, Li X, Dai J, et al. Immune cell subset differentiation and tissue inflammation. *J Hematol Oncol* 2018; **11**(1):97.
44. Kim J, Stanescu DE, Won KJ. CellBIC: bimodality-based top-down clustering of single-cell RNA sequencing data reveals hierarchical structure of the cell type. *Nucleic Acids Res* 2018; **46**(21):e124.
45. Peng M, Wamsley B, Elkins AG, et al. Cell type hierarchy reconstruction via reconciliation of multi-resolution cluster tree. *Nucleic Acids Res* 2021; **49**(16):e91.
46. Xu J, Liao K, Yang X, et al. Using single-cell sequencing technology to detect circulating tumor cells in solid tumors. *Mol Cancer* 2021; **20**(1):104.
47. Ledergor G, Weiner A, Zada M, et al. Single cell dissection of plasma cell heterogeneity in symptomatic and asymptomatic myeloma. *Nat Med* 2018; **24**(12):1867–76.
48. Piwecka M, Rajewsky N, Rybak-Wolf A. Single-cell and spatial transcriptomics: deciphering brain complexity in health and disease. *Nat Rev Neurol* 2023; **19**(6):346–62.
49. Nguyen A, Khoo WH, Moran I, et al. Single cell RNA sequencing of rare immune cell populations. *Front Immunol* 2018; **9**:1553.
50. Kumar P, Tan Y, Cahan P. Understanding development and stem cells using single cell-based analyses of gene expression. *Development* 2017; **144**(1):17–32.
51. Potter SS. Single-cell RNA sequencing for the study of development, physiology and disease. *Nat Rev Nephrol* 2018; **14**(8):479–92.
52. Wegmann R, Neri M, Schuierer S, et al. CellSIUS provides sensitive and specific detection of rare cell populations from complex single-cell RNA-seq data. *Genome Biol* 2019; **20**(1):142.
53. Gerniers A, Bricard O, Dupont P. MicroCellClust: mining rare and highly specific subpopulations from single-cell expression data. *Bioinformatics* 2021; **37**(19):3220–7.
54. Jindal A, Gupta P, Jayadeva, Sengupta D. Discovery of rare cells from voluminous single cell expression data. *Nat Commun* 2018; **9**(1):4719.
55. Sadeghi M, Armanfard N. Deep Clustering with Self-supervision using Pairwise Data Similarities, 2021, TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.14852652.v3>.
56. Zavidij O, Haradhvala NJ, Mouhieddine TH, et al. Single-cell RNA sequencing reveals compromised immune microenvironment in precursor stages of multiple myeloma. *Nat Cancer* 2020; **1**(5):493–506.
57. Cortinas-Lorenzo B, Perez-Gonzalez F. Adam and the ants: on the influence of the optimization algorithm on the detectability of DNN watermarks. *Entropy (Basel)* 2020; **22**(12):1379.

58. Bhattacherjee A, Djekidel MN, Chen R, et al. Cell type-specific transcriptional programs in mouse prefrontal cortex during adolescence and addiction. *Nat Commun* 2019;10(1):4169.
59. Tasic B, Yao Z, Graybuck LT, et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature* 2018;563(7729):72–8.
60. Zeisel A, Muñoz-Manchado AB, Codeluppi S, et al. Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science (New York, NY)* 2015;347(6226):1138–42.
61. Žurauskienė J, Yau C. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics* 2016;17:140.
62. Lopez R, Regier J, Cole MB, et al. Deep generative modeling for single-cell transcriptomics. *Nat Methods* 2018;15(12):1053–8.
63. Deng Y, Bao F, Dai Q, et al. Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nat Methods* 2019;16(4):311–4.
64. Butler A, Hoffman P, Smibert P, et al. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;36(5):411–20.
65. Du H, Wang Z, Guo R, et al. Transcription factors Bcl11a and Bcl11b are required for the production and differentiation of cortical projection neurons. *Cereb Cortex* 2022;32(17):3611–32.
66. Bragina L, Fattorini G, Giovedí S, et al. Analysis of Synaptotagmin, SV2, and Rab3 expression in cortical glutamatergic and GABAergic axon terminals. *Front Cell Neurosci* 2011;5:32.
67. Babiczky A, Matyas F. Molecular characteristics and laminar distribution of prefrontal neurons projecting to the mesolimbic system. *Elife* 2022;11:11.
68. Dutta AK, Alberge JB, Sklavenitis-Pistofidis R, et al. Single-cell profiling of tumour evolution in multiple myeloma - opportunities for precision medicine. *Nat Rev Clin Oncol* 2022;19(4):223–36.
69. Boiarsky R, Haradhvala NJ, Alberge JB, et al. Single cell characterization of myeloma and its precursor conditions reveals transcriptional signatures of early tumorigenesis. *Nat Commun* 2022;13(1):7040.
70. Holla P, Dizon B, Ambegaonkar AA, et al. Shared transcriptional profiles of atypical B cells suggest common drivers of expansion and function in malaria, HIV, and autoimmunity. *Sci Adv* 2021;7(22):eabg8384.
71. Luger D, Yang YA, Raviv A, et al. Expression of the B-cell receptor component CD79a on immature myeloid cells contributes to their tumor promoting effects. *PLoS One* 2013;8(10):e76115.
72. Mason DY, Cordell JL, Brown MH, et al. CD79a: a novel marker for B-cell neoplasms in routinely processed tissue samples. *Blood* 1995;86(4):1453–9.
73. Chen D, Tang TX, Deng H, et al. Interleukin-7 biology and its effects on immune cells: mediator of generation, differentiation, survival, and homeostasis. *Front Immunol* 2021;12:747324.
74. Johnson RC, Kim J, Natkunam Y, et al. Myeloid cell nuclear differentiation antigen (MNDA) expression distinguishes Extramedullary presentations of myeloid Leukemia from Blastic Plasmacytoid dendritic cell neoplasm. *Am J Surg Pathol* 2016;40(4):502–9.
75. Fabisik M, Tureckova J, Pavliuchenko N, et al. Regulation of inflammatory response by transmembrane adaptor protein LST1. *Front Immunol* 2021;12:618332.
76. Wex T, Bühling F, Wex H, et al. Human cathepsin W, a cysteine protease predominantly expressed in NK cells, is mainly localized in the endoplasmic reticulum. *J Immunol* 2001;167(4):2172–8.
77. Stoeckle C, Gouttefangeas C, Hammer M, et al. Cathepsin W expressed exclusively in CD8+ T cells and NK cells, is secreted during target cell killing but is not essential for cytotoxicity in human CTLs. *Exp Hematol* 2009;37(2):266–75.
78. Brendel M, Su C, Bai Z, et al. Application of deep learning on single-cell RNA sequencing data analysis: a review. *Genomics Proteomics Bioinformatics* 2022;20(5):814–35.
79. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 1987;20:53–65.
80. Berest I, Tangherloni A. Integration of scATAC-Seq with scRNA-Seq data. *Methods Mol Biol* 2023;2584:293–310.
81. Liu F, Wang Y, Gu H, Wang X. Technologies and applications of single-cell DNA methylation sequencing. *Theranostics* 2023;13(8):2439–54.
82. Hu J, Li X, Hu G, et al. Iterative transfer learning with neural network for clustering and cell type classification in single-cell RNA-seq analysis. *Nat Mach Intell* 2020;2(10):607–18.
83. Williams CG, Lee HJ, Asatsuma T, et al. An introduction to spatial transcriptomics for biomedical research. *Genome Med* 2022;14(1):68.
84. Longo SK, Guo MG, Ji AL, Khavari PA. Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nat Rev Genet* 2021;22(10):627–44.