USCAP

www.nature.com/labinvest

ARTICLE

An active learning approach for clustering single-cell RNA-seq data

Xiang Lin[1,3], Haoran Liu[1,3], Zhi Wei[1]✉, Senjuti Basu Roy[1] and Nan Gao[2]

Single-cell RNA sequencing (scRNA-seq) data has been widely used to profile cellular heterogeneities with a high-resolution picture. Clustering analysis is a crucial step of scRNA-seq data analysis because it provides a chance to identify and uncover undiscovered cell types. Most methods for clustering scRNA-seq data use an unsupervised learning strategy. Since the clustering step is separated from the cell annotation and labeling step, it is not uncommon for a totally exotic clustering with poor biological interpretability to be generated—a result generally undesired by biologists. To solve this problem, we proposed an active learning (AL) framework for clustering scRNA-seq data. The AL model employed a learning algorithm that can actively query biologists for labels, and this manual labeling is expected to be applied to only a subset of cells. To develop an optimal active learning approach, we explored several key parameters of the AL model in the experiments with four real scRNA-seq datasets. We demonstrate that the proposed AL model outperformed state-of-the-art unsupervised clustering methods with less than 1000 labeled cells. Therefore, we conclude that AL model is a promising tool for clustering scRNA-seq data that allows us to achieve a superior performance effectively and efficiently.

Laboratory Investigation (2022) 102:227–235; https://doi.org/10.1038/s41374-021-00639-w

INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) technology is a revolutionary tool that has been widely used to investigate cellular heterogeneity in various tissues [1, 2]. Despite its popularity, the analysis of scRNA-seq data remains a challenging task [3, 4]. Specifically, due to the low RNA capture rate and the low sequencing depth per cell, gene-expression measurements in the scRNA-seq data are low and sparse, with many "false" zero count observations defined as dropout events [5]. Due to the noise in scRNA-seq data, the tools designed for the analysis of bulk RNA-seq data may not be appropriate for analyzing scRNA-seq data. Additionally, cell types are largely unknown in most scRNA-seq studies. Researchers generally employ unsupervised clustering methods to group cells into sets. Based on the clustering results, they can characterize and determine cell types [6]. Identifying cell types is a challenging problem in the analysis of scRNA-seq data [7]. After cells are clustered into groups, a common practice is to use known marker genes to determine cell types [8, 9]. For example, clusters with marker genes CD8A and CD8B highly expressed are identified as the CD8+ T cells; cell clusters enriched with genes CST3, CD1C, and FCER1A can be defined as the dendritic cells [9]. However, many cells and cell types cannot be determined by using known marker genes [8]. For example, in the study from Wang et al., the authors used both the marker genes and the ADTs (antibody-derived tags) to estimate cell type, but there were still about twenty percent of cells that could not be labeled. Thus, they had to exclude these cells when evaluating the clustering performance [8].

Unsupervised clustering analysis has been widely used for the analysis of scRNA-seq data. It is a crucial step for identifying and uncovering cell types, the central goal for most scRNA-seq studies. Numerous clustering methods have been developed for the analysis of scRNA-seq data [10–12]. However, most, if not all, of them are unsupervised learning approaches [6]. Biologists annotate and label clusters using their domain knowledge after the clustering is done. The clustering step is separated from the cell annotation and labeling step, which may not be optimal. It is not uncommon for a totally exotic clustering with poor biological interpretability to be generated—a result generally undesired by biologists. To overcome this problem, a potential solution is to consider and integrate cell annotation and label information in the clustering step [13, 14].

On the other hand, it is not feasible for biologists to annotate and label all the cells in a dataset for two main reasons. First, it is time-consuming. A typical scRNA-seq dataset can have several thousand or even tens of thousands of cells. Biologists cannot afford to manually examine each of them for labeling. Second, while we recognize the value of prior biological domain information in facilitating cell type assignment, we still hope clustering is mainly decided by the data itself.

As a compromise, there is a (small) subset of cells that will be annotated by biologists using their domain knowledge, e.g., marker genes. Then cell type assignment will be done for the rest of the cells based on this small set of labeled cells. To optimize the cell type assignment (cell clustering), we propose to formulate it as an active learning (AL) problem. Here we have an abundance of

[1] Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. [2] Department of Biological Sciences, Rutgers University, Newark, NJ, USA. [3] These authors contributed equally: Xiang Lin, Haoran Liu ✉email: zhiwei@njit.edu

Received: 14 March 2021 Revised: 22 June 2021 Accepted: 23 June 2021
Published online: 9 July 2021

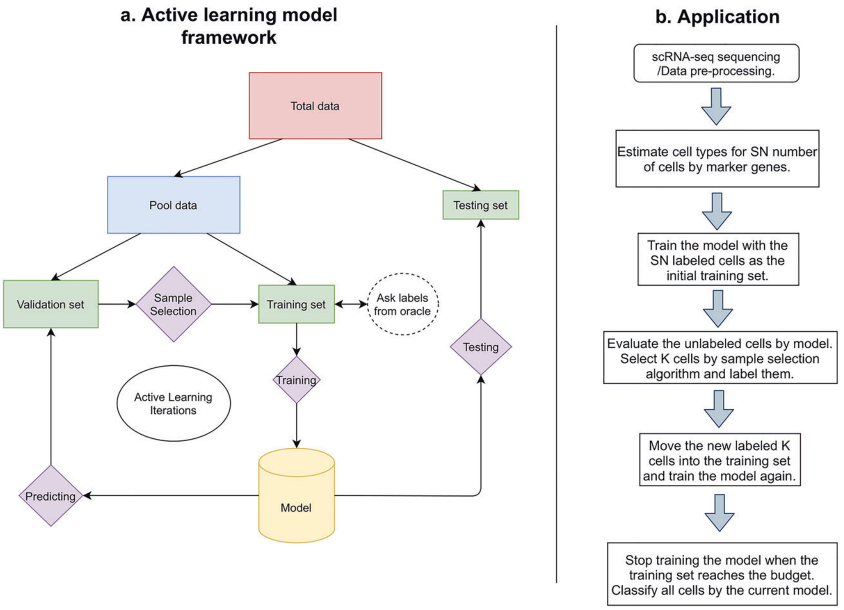
a. Active learning model framework
Total data
Pool data
Testing set
Validation set
Sample Selection
Training set
Ask labels from oracle
Testing
Training
Active Learning Iterations
Model
Predicting
b. Application
scRNA-seq sequencing /Data pre-processing.
Estimate cell types for SN number of cells by marker genes.
Train the model with the SN labeled cells as the initial training set.
Evaluate the unlabeled cells by model. Select K cells by sample selection algorithm and label them.
Move the new labeled K cells into the training set and train the model again.
Stop training the model when the training set reaches the budget. Classify all cells by the current model.

Fig. 1 Architecture and protocol of the active learning model. In the architecture panel (**a**), Rectangles stand for the data partitions and diamonds stand for the actions. Total data is divided into two parts: pool data and testing data. Then the pool data is divided into a validation set and a training set. Only the training set needs labels. The training set will be used to train the model. Then the validation set will be predicted by the model. Based on the sample selection method, the most informative samples in the validation set will be moved to the training set and the labels of them are acquired from the oracle. This is one iteration of active learning. The iterations will continue until the number of samples in the training set reaches the pre-defined budget. The procedure of using the active learning model on the scRNA-seq data is shown in panel **b**.

unlabeled cells, while manual labeling is expected to be applied to only a subset of cells. In such a scenario, learning algorithms can actively query the biologist for labels. We call this type of iterative supervised learning AL. We hypothesize that the manual labeling by biologists, although applied to a small number of cells, would help to keep clustering on the right track. We expect that AL algorithms can benefit from the labeled samples and will outperform conventional unsupervised learning approaches. AL models allow us to achieve good performance with fewer labeled instances [15]. Given the general AL framework, some application-specific designs need to be investigated for optimizing its application to scRNA-seq data. Briefly, with the fixed budget (the number of cells to be labeled), we need to decide which cells are the most informative to be labeled by biologists, and how often they should be labeled, in order to make the model more effective and efficient [16].

In this study, we proposed an AL framework for scRNA-seq data clustering. We developed an optimal iterative learning procedure by exploring several key parameters in the experiments. Based on testing extensive real scRNA-seq datasets, the proposed AL model outperformed state-of-the-art unsupervised clustering methods.

MATERIALS AND METHODS

Data preprocessing

We downloaded the raw count matrices of the four scRNA-seq datasets from online databases (see details below). Normalization is first performed on the raw data to remove the batch effect. We performed data normalization by Seurat [12]: feature counts for each cell are divided by the total counts for that cell and multiplied by the scale factor (default 10,000). The values are then natural log transformed. Then the top 2000 high variable genes across the cells are selected. Generally, top 2000 genes are adequate to cover the whole cells' variances [12].

AL model framework

Before explaining the framework, we should introduce three key parameters in the AL model: (1) SN, the initial number of cells used for training, (2) K, the number of cells which will be added to the training set in each learning iteration, and (3) Budget, the total number of cells with labels. In the experiments, we tested the effects of these parameters on the AL model's clustering performance. In an AL process, samples will be divided into several parts (Fig. 1a): (1) Pool data, which contains the total cells allowed to be labeled by the oracle (such as a biologist), (2) Training set. Initially, the training set has only an SN number of cells. In each learning iteration, K cells are added to the training set until reaching the budget, (3) Validation set (Pool data—training set). In each learning iteration, a sample selection algorithm is performed on the validation set by which the most informative K cells will be moved from the validation set to the training set, and (4) Testing set (Total data—pool data). It is used to test the performance of the model in each learning iteration. The number of cells in the testing set is constant over the entire learning process. In this study's experiments, we set 70% cells as the pool data and 30% cells as the testing data. For each experiment setting, a baseline (BL) model is built as the AL model's benchmark. In the BL models, a budget number of cells is randomly sampled and used to train the model. In the initial training set, at least one cell is sampled from each class (cell type).

The pipeline of the AL model is shown in Fig. 1b. Before running the model, classifiers, budget, SN, and K should be predefined. We first estimate the cell types (labels) of the cells in the initial training set by prior knowledge (marker genes or other methods). The AL model is first trained on the initial training cells. Then, the validation set are predicted by this model, and the probability of each sample to be classified into each cluster is calculated. Based on these probabilities, the sample selection method (mentioned below) is used to move K cells from the validation set to the training set. Here, we need to estimate the cell types of the new training cells by prior knowledge (namely ask oracle for the cell types). The updated training set is then used to train the model again, and the training set will be further updated by the sample selection approach on the validation set. This loop will continue until the number of cells in the training set reaches the predefined budget.

Classifier

In this study, we tested four classifiers in the AL framework: (1) Support-vector machines (SVM), (2) Random Forest (RF), (3) Logistic Regression (LR), and (4) Multilayer Perceptron (MLP). All the classifiers are implemented by the scikit-learn package in python 3.8. For MLP, we set the layers as 256:128:64:32:16 after tunning the parameters. All other parameters are kept in default. Specifically, the activation function is Relu; the optimizer is Adam; the batch size is 200 and the learning rate is 0.001.

Clustering performance evaluation

We use multiple metrics to quantify the performance of clustering/classification. Firstly, we use accuracy, precision, recall and F1 score to compare the performance between the AL models and the BL models. Denoting the true positive, true negative, false positive, and false negative as TP, TN, FP, and FN, the accuracy (ACC) is defined as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is defined as:

$$Precision = \frac{TP}{TP + FP}$$

Then recall is defined as:

$$Recall = \frac{TP}{TP + FN}$$

And F1 score is:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

We also use Adjusted Rand Index (ARI) [17], Normalized Mutual Information (NMI) [18], and CA (clustering accuracy) to compare the clustering performance between the AL models and the unsupervised clustering methods. In the formula of ARI below, L_p and L_t are the predicted cluster labels and the true labels, respectively; k_p and k_t are the predicted cluster number and the true cluster number, respectively; n_k denotes the number of cells assigned to a specific cluster k ($k = 1, 2, \ldots, k_t$); similarly, n_t denotes the number of cells assigned to cluster t ($t = 1, 2, \ldots, k_p$); n_{kt} represents the number of cells shared between cluster k and t; and n is the total number of cells.

$$ARI(L_p, L_t) = \frac{\sum_{kt}\binom{n_{kt}}{2} - \left(\sum_k\binom{n_k}{2}\sum_t\binom{n_t}{2}\right)\Big/\binom{n}{2}}{\frac{1}{2}\left(\sum_k\binom{n_k}{2} + \sum_t\binom{n_t}{2}\right) - \left(\sum_k\binom{n_k}{2}\sum_t\binom{n_t}{2}\right)\Big/\binom{n}{2}}$$

NMI is defined as:

$$NMI = \frac{I(C, G)}{\max\{H(C), H(G)\}}$$

Where I (C, G) stands for the mutual information between the predicted clusters C and the true clusters G and is defined as:

$$I(C, G) = \sum_{p=1}^{tc}\sum_{q=1}^{tg}|C_p \cap G_q|\log\frac{n|C_p \cap G_q|}{|C_p| \times |G_q|}$$

Where tc and tg stand for the number of clusters in C and G. H (C) and H (G) represent the entropies:

$$H(C) = -\sum_{p=1}^{tc}|C_p|\log\frac{|C_p|}{n}$$

$$H(G) = -\sum_{q=1}^{tg}|G_q|\log\frac{|G_q|}{n}$$

Clustering accuracy (CA) is designed to measure the best matching between predicted and true clusters, which is:

$$CA = \max_m \sum_{i=1}^{n} 1\frac{\{\widehat{l_i} = m(l_i)\}}{n}$$

Where $\widehat{l_i}$ and l_i are the true and predicted labels from clustering algorithms, n is the number of cells and m is the number of all possible one-to-one mapping between $\widehat{l_i}$ and l_i. Hungarian algorithm [19] is used to find the best mapping.

Sample selection algorithm

In each iteration of training, K cells will be selected by a sample selection algorithm and moved from validation set to training set. Two sample selection methods are tested in this study: (1) Entropy-based sample selection:

$$H = -\sum_{i=1}^{C} P_i \log_2 P_i$$

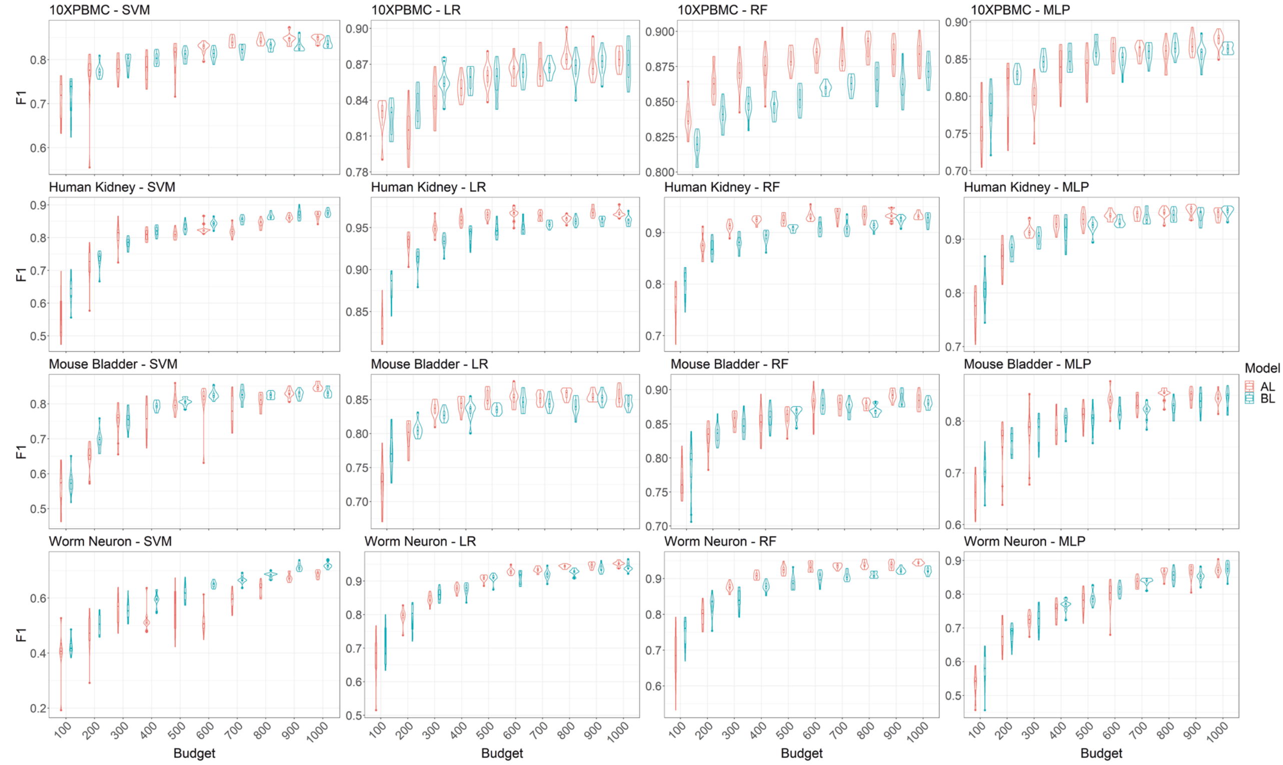

Fig. 2 Clustering performance test of AL models on the different datasets. This figure shows the F1 scores from the SVM (column 1), LR (column 2), RF (column 3) and MLP (column 4) - based AL and BL models for the dataset 1) 10X PBMC (row 1); 2) human kidney (row 2); 3) mouse bladder (row 3); and 4) worm neuron (row 4). Budgets are varied from 100 to 1000. SN and K are fixed as 50 and 20. In most datasets, RF-based AL models outperform the BL models ($P < 0.05$).

Where H is the entropy of a cell, p is the probability of the cell to be classified into the i cluster. In each training iteration, K cells with the highest entropy will be selected as the new training samples, (2) Margin based sample selection [20]: the margin is calculated as the difference of the highest probability and the second-highest probability that a cell is classified to the clusters. In each training iteration, K cells with the lowest margin will be selected as the new training samples. According to the results of the pre-experiments, different sample selection algorithms led to the similar clustering performance. Therefore, we only showed the results from the AL model with the entropy-based sample selection algorithm.

Parameter testing

The parameters tested in this study include: (1) the initial training sample size (SN), (2) the added (selected) cells in each learning iteration (K), and (3) the budget. Specifically, the SN is varied from 50 to 500 with the budget and the K fixed as 1000 and 20, respectively, the budget is varied from 100 to 1000 with the K and the SN fixed as 20 and 50, respectively, and the K is varied from 10 to 100 with the budget and the SN fixed as 1000 and 50, respectively. Pearson's correlation is performed between the budget, SN, K, and the clustering performance (such as F1 score and ARI). While running the AL model with each parameter setting, the BL model's performance is also calculated. The experiment with each parameter setting is replicated ten times, and a one-tailed independent *T*-test is performed between the performance of the AL and the BL models.

Real datasets

The 10× PBMC dataset is provided by the 10× scRNA-seq platform [21], which profiles the transcriptomes of about 4000 peripheral blood mononuclear cells (PBMCs) from a healthy donor. We downloaded the filtered gene/cell matrix (2100 cells × 16,653 genes) from the 10× genomics website (https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc4k). Cell labels identified by the graph-based clustering are used as the ground-truth labels.

Worm neuron cells dataset is profiled by the sci-RNA-seq platform (single-cell combinatorial indexing RNA sequencing) [22]. The authors profiled about 50,000 cells from the nematode Caenorhabditis elegans at the L2 larval stage and identified the cell types (http://atlas.gs.washington.edu/worm-rna/docs/). We selected the subset of the neural cells and removed the cells with the label of "Unclassified neurons". Two thousand one hundred neural cells are used for this study (2100 cell × 13,488 genes).

The human kidney dataset contains 5685 cells by 25,215 genes in 11 clusters. Authors profiled 577 renal tumors and normal tissue from human fetal, pediatric, and adult kidneys [23]. We downloaded the data from the website (https://github.com/xuebaliang/scziDesk/tree/master/dataset/Young). The filtered data with 2100 cells are used in this study.

Mouse bladder cell dataset is provided by the Mouse Cell Atlas project [24] (https://figshare.com/s/865e694ad06d5857db4b). The count matrix contains 400,000 single cells sorted by the tissues. The authors annotated the cell types. In this study, we selected the cells from the bladder tissue (2100 cells × 20,670 genes).

For all datasets, feature selection is performed according to the pipeline of Seurat [12]. Top 2000 genes are selected for the downstream analyses. As shown in Fig. S1, cells in all datasets are unevenly distributed in the different clusters. In this case, all the initial training cells are evenly sampled from each cluster to remove the effect of cluster size.

Tests of four popular unsupervised clustering methods

Four popular unsupervised clustering methods of scRNA-seq data are tested on the four datasets and compared with the AL model, including: (1) K-means, (2) Seurat [12], (3) Tscan [10], and (4) SC3 [11]. The normalized data is used as the input for K-means and Tscan. The raw count is used for Seurat as it has an embedded normalization. SC3 needs both the raw count and the normalized count as the input. One-tailed independent T-test is performed between the performance of the AL model and the competing methods.

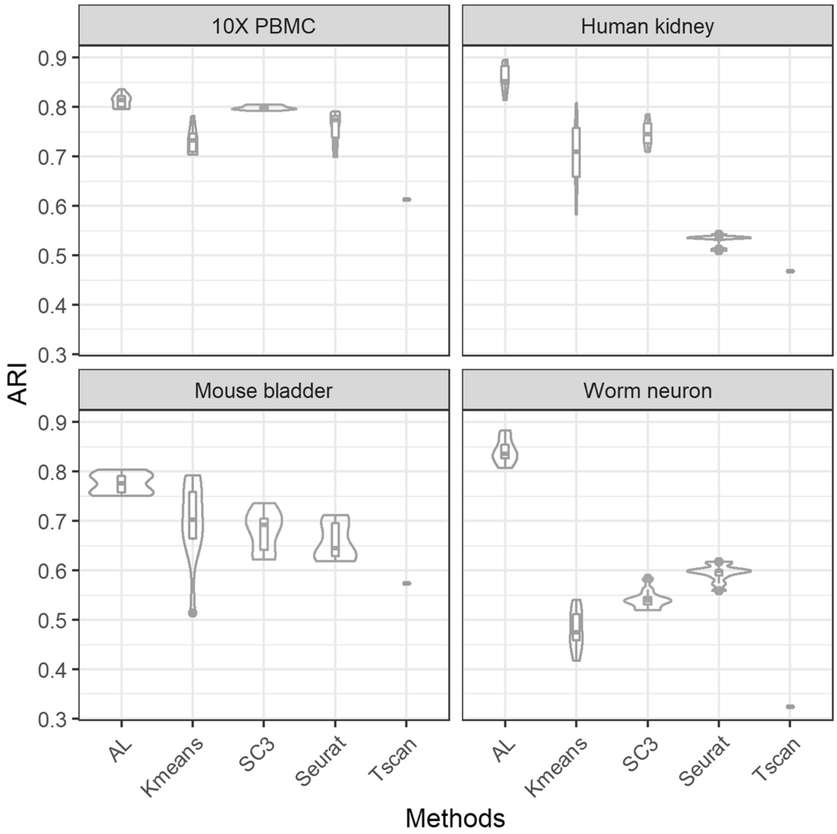

Fig. 3 Clustering performance of the AL model and four popular unsupervised clustering methods. This figure shows the ARI from the unsupervised clustering methods (Seurat, SC3, Tscan, and K-means) and AL model on the four datasets (top-left: 10X PBMC; top-right: human kidney; bottom-left: mouse bladder, and bottom-right: worm neuron). The parameters of AL model here are: {SN:50; K:20; Budget: 800 and Classifier: RF}. AL model can always outperform other methods.

RESULTS

The effects of the budget on the AL model

We first vary the budget and fix other parameters. We illustrate the F1 score, ACC, ARI, and NMI in the AL models and the corresponding BL models. Fig. 2 shows the F1 scores in the experiments of the four datasets using the different classifiers. In all the experiments, F1 scores are positively correlated with the budgets for AL models ($P < 0.05$, Fig. S5). Here, we find a breakpoint in the curve for each dataset, which means that using the massive training samples in the AL model will not increase but even decrease the model's performance. The cutoff (breakpoint) in all the experiments is about 600–900 cells. It is noted that for most MLP-based AL models, the clustering performance is still rising up after 900 cells, indicating its high dependency on the sample size. The rest of metrics (ACC, ARI, and NMI) are illustrated in Figs. S2, S3, S4, and their correlations with budgets are shown in Figs. S6, S7, S8.

Then, we focus on the improvements from AL to BL models with various budgets. For most datasets, the RF-based AL models get the higher F1 scores than the BL models ($P < 0.05$) (Fig. 2). The only exception is the mouse bladder dataset, in which the AL models only have subtle differences with the BL models. The LR-based AL models also show satisfactory improvements from the BL models (Fig. 2) in human kidney, mouse bladder and worm neuron datasets, but the magnitude of improvements is lower than that from RF-based AL models. The results for all the metrics of the models are listed in Table S1 and the results for the *T*-tests between the AL and the BL models with various budgets are listed in Table S2. For the SVM and MLP-based AL models, the improvements of F1 score (and other metrics) to the BL models are negligible (Figs. 2, S1, S2, S3).

In addition, we explore the best model for each dataset. Although different datasets prefer different classifiers, AL models

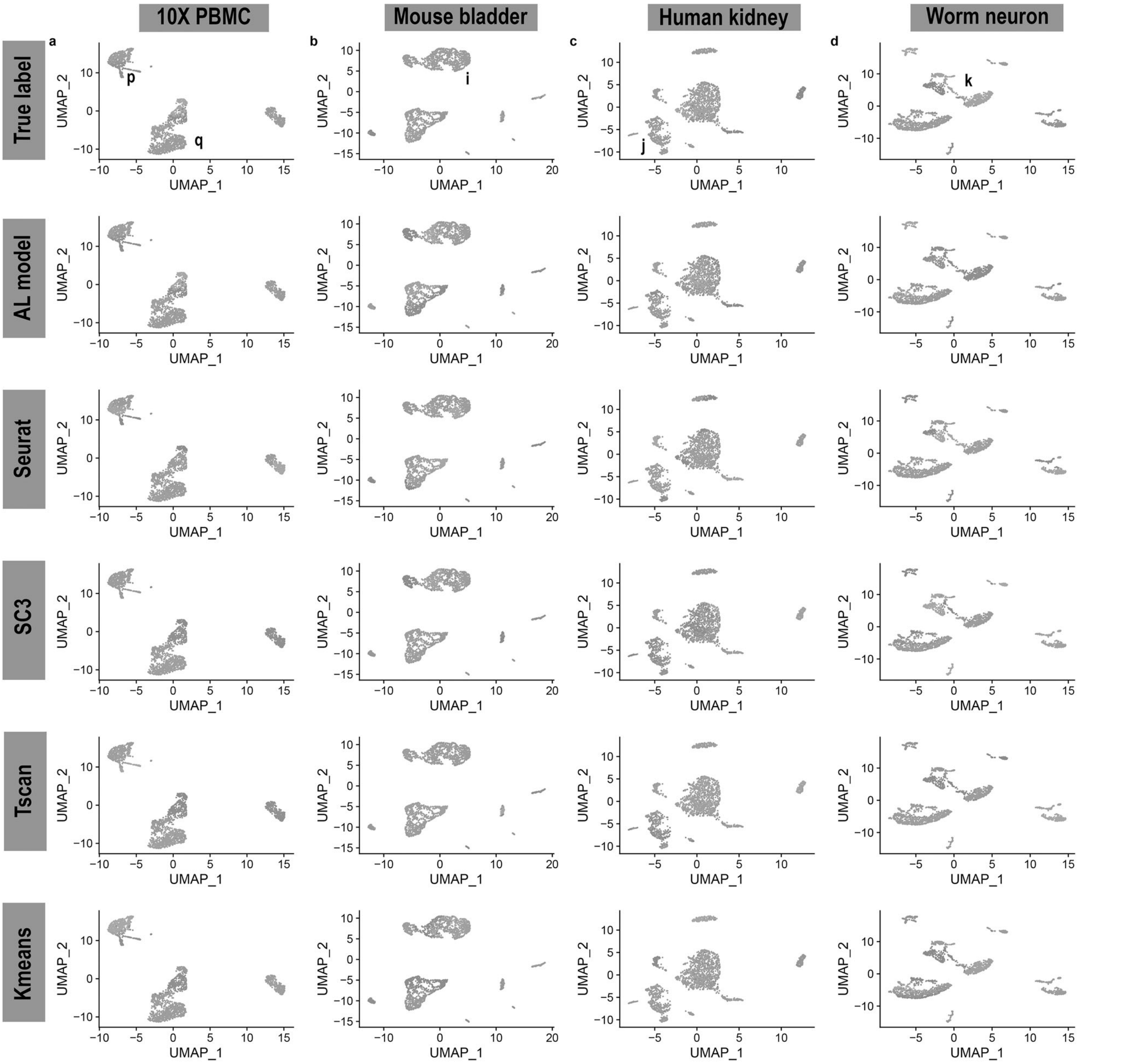

Fig. 4 Low-dimensional representations of the real datasets with the predicted labels from different methods. U-map is built for the dataset **a** 10X PBMC, **b** mouse bladder, **c** human kidney, and **d** worm neuron with the labels from 1) true label (1st row); 2) active learning model (2nd row); 3) Seurat (3rd row); 4) SC3 (4th row); 5) Tscan (5th row) and 6) K-means (6th row). The pattern of clustering from the AL model is more similar to that from the true labels than all other methods and for all datasets.

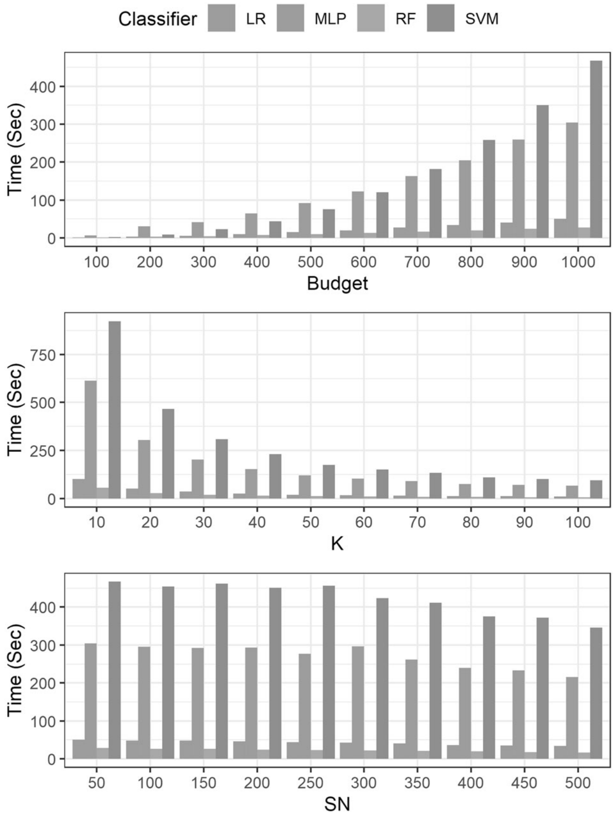

Fig. 5 Running time test of SVM and RF-based AL models. We varied Budgets (row 1), K (row 2), and SN (row 3) to explore their influence on the running time. Budgets and K are positively and negatively correlated with the running times, respectively. RF-based AL models have lower running time than other AL models. All the experiments here are performed on the 10X PBMC dataset.

can always outperform all the BL models. Specifically, for human kidney dataset, the LR-based AL model with budget 900 has the best clustering performance overall models (*T*-test for F1 score with $P < 0.001$). For 10X PBMC dataset, the RF-based AL model with budget 800 outperforms all other models (*T*-test for F1 score with $P < 0.001$). For mouse bladder dataset, the RF-based AL model with budget 900 has the optimal clustering performance overall models (*T*-test for F1 score with $P < 0.001$). For the worm neuron dataset, the LR-based AL model with budget 1000 outperforms all other models (*T*-test for F1 score with $P < 0.05$). The RF-based AL model with budget 1000 also performs well on this dataset. Combining with the results before, we conclude that LR and RF are the superior classifiers for AL framework.

The effects from the K and SN to the AL model

We then fix the budget as 1000 and vary the K from 10 to 100 and then vary the SN from 50 to 500. We find that both K and SN have no significant influence on the clustering performance of the AL models (Figs. S9, S10). Similar to the results before, RF-based AL models always outperform the BL models in most datasets regardless of the variation of K and SN.

Comparing the AL model with the unsupervised competing methods

To demonstrate the advantages of using AL models, we compare it with four popular unsupervised clustering methods (Seurat, SC3, Tscan, and K-means) for the four scRNA-seq datasets. RF-based AL model with budgets = 800, K = 20, and SN = 50 is used to compare with the unsupervised methods. The experiment for each dataset is replicated ten times. The clustering metrics ARI, NMI, and CA are used. Fig. 3 indicates that the AL model has the highest ARI in all the methods. It exceeded about 10 percent ($P < 0.001$) and 30 percent ($P < 0.001$) of ARI than other methods in human kidney and worm neuron dataset, respectively. For 10× PBMC and mouse bladder dataset, the magnitude of improvement is lower but still significant ($P < 0.05$). The results of CA and NMI are shown in Figs. S11, S12. The data for this experiment is in Table S3.

On the U-map of the four datasets (Fig. 4), we find that the AL model's clustering pattern is more similar to that of the true label than other methods. Specifically, on the U-maps of the 10× PBMC dataset (column a in Fig. 4), the clustering patterns on the cell island p and q from the AL model are almost the same as that from the true labels, in which the cell island p and q are divided into 3 and 4 clusters, respectively. However, Seurat divide the cell island q into five clusters; SC3 define the cell island p as a whole cluster; Tscan and K-means divide the cell island q into three clusters. All the unsupervised methods have some biases on the clustering patterns, even for the large clusters. Similar scenarios can be found on the island i, j, and k from the U-maps of the mouse bladder, human kidney, and worm neuron dataset, respectively. Only the AL model has a similar clustering pattern with that from the true labels on these cells. In summary, these results indicate that the domain knowledge (cell types) is essential for the scRNA-seq data clustering. By only acquiring the cell types of a few hundred cells, the clustering performance will be highly improved using the AL model rather than using the unsupervised methods.

Running time of the AL model

Fig. 5 shows the running times of the AL models on 10× PBMC dataset. A large budget will prolong the AL model's running time. A small k will also increase the running time because, with a small k, more iterations of training are needed to reach the budget. SN just has a slight influence on the running time. Combining the results from Fig. 2, we claim that it is essential to find an appropriate budget for an AL model. A too-large budget will prolong the running time and impact the clustering performance. We find that the RF-based models always have the lowest running time than other models. When increasing the budgets or decreasing the K, RF-based AL models get the slowest growth of running time. This result demonstrates that RF is the best classifier for AL framework.

DISCUSSION

The real-world application of the AL model

As shown in Fig. 1b, the AL models can be easily employed in the real-world scRNA-seq data clustering. For using the AL model, researchers need to normalize the raw count data firstly. We suggest running the feature selection before doing the clustering analysis in which the top 2000 (or less) most informative genes can be selected. After preprocessing, the K, SN, and budget should be predefined according to the total number of cells. According to this study's results, we recommend setting the budget as a tenth of the total cell number, the SN as a tenth of the budget, and the K as half of the SN. Then, researchers need to estimate the cell types of the initial training cells by using marker genes. It is recommended to include as many cell types as possible in the initial training set. After the first iteration of training, the unlabeled cells will be evaluated by the model, and the most informative cells will be selected by the sample selection algorithm. Researchers need to estimate the cell types of these cells (by marker genes or other methods) and add them to the training set. Then the model will be trained again by the new training set. The training iterations will continue until the training set reaches the budget. After this AL process, only a tenth of the total cells is

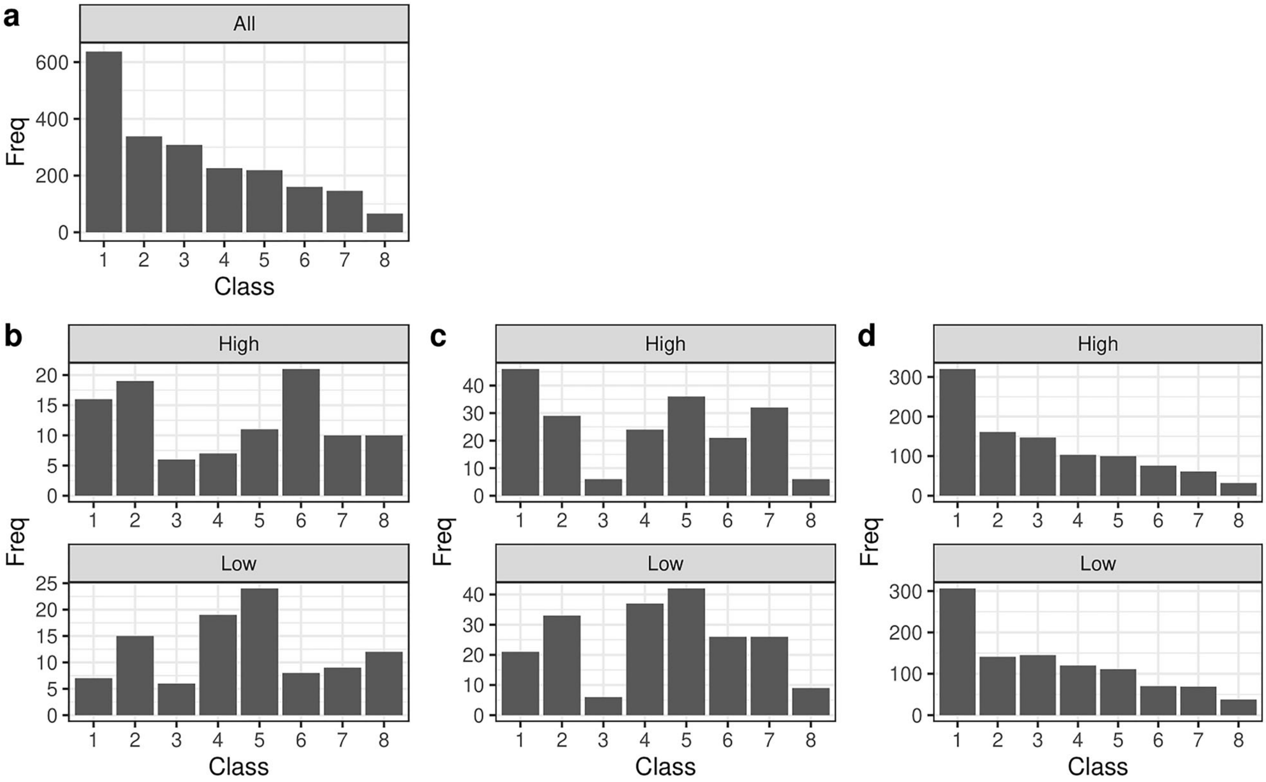

Fig. 6 The label distribution of the training cells in the best and worst AL models. The label distributions are from: 1) the best model with budget 100 (**b** top); 2) the best model with budget 200 (**c** top); 3) the best model with budget 1000 (**d** top); 4) the worst model with budget 100 (**b** bottom); 5) the worst model with budget 200 (**c** bottom); 6) the worst model with budget 1000 (**d** bottom). Panel **a** shows the distribution of the true labels. The dataset and model used here are 10X PBMC and RF-based AL model, respectively. K and SN are set as 20 and 50, respectively. When the model uses the data with a similar label distribution from the true labels, it tends to perform better.

labeled by the researchers. The well-trained model can be used for the final clustering (classification).

Explaining the high variances in the clustering performance of the AL model

In our experiments, we find that even with the same setting, the AL models' performance still has a high variance among the ten runs. The variances are higher when the budget is 100 and 200 and lower when the budget is higher. To explore the variances' causes, we choose the best and the worst run in the budget experiment of the 10× PBMC dataset where the budget is set as 100, 200, and 1000. The label distribution in the training set of the best run is shown in Fig. 6b top, c top, d top, and that of the worst run are shown in Fig. 6b bottom, c bottom, d bottom. The true label distribution is shown in Fig. 6a. We find that the training cells in the best runs (see Fig. 6b, c top) contain more cells from the big clusters (clusters 1 and 2 from the true label) and fewer cells from the small clusters. In other words, the best runs' training cell distribution is more consistent with that of the true label. On the contrary, the training cells in the worst runs (see Fig. 6b, c bottom) are majorly from the small clusters (clusters 4 and 5), so the cell distribution is inconsistent with that of the true label (Fig. 6a). For the experiments with budget 1000, the cell distribution of the best and the worst runs get closer (see Fig. 6d top, bottom). As a result, their performance also tend to be uniform. In further studies, as indicated by Wei et al. [25], adding a monitor for the cluster distribution of the training cells during the AL iterations may further improve the AL model's performance. An algorithm is needed here, which should consider the tradeoff between the cluster distribution of the overall training cells and the information carried by the individual cells.

Potential problems of the scRNA-seq data clustering

Two critical problems in the scRNA-seq data are the high ratio of zero and the low count per cell [6]. Fig. 7 shows the ratio of zero (b) and the average per cell count (a) in the four datasets. The worm neuron dataset has the highest zero ratios and the lowest average per cell count. Some classifiers cannot keep an excellent performance on the highly zero-inflated data. Our experiment show that the SVM-based models only got about 0.6 of F1 score and 0.4 of ARI for this dataset. However, RF and LR-based AL models can maintain a satisfactory performance on this dataset (Figs. 2, S1, S2, S3).

Limitations of the AL model

Like the AL model, semi-supervised clustering approaches typically focus on using labeled data (obtained from known datasets or derived from marker genes) to help initialize clusters and adjust clusters during the training. These approaches would always assume that the labeled data and the unlabeled data share the same distribution and therefore require high-quality labeled data. Also, the tuning of weights of labeled data and unlabeled data for the model would be tricky and varies with the size of labeled data. Recently, Tian et al. proposed a semi-supervised clustering method, scDCC [13], which converts the prior knowledge from marker genes into soft pairwise constraints to supervise the clustering. Although both the scDCC and the AL model can take advantages of the knowledge provided by the marker genes, they are different on both the clustering algorithm and the way of integrating prior knowledge. Compared to scDCC, AL model can integrate more prior knowledge from marker genes. However, as a supervised approach, it also has some disadvantages. As we showed in our experiments, we arbitrarily sampled at least one cell

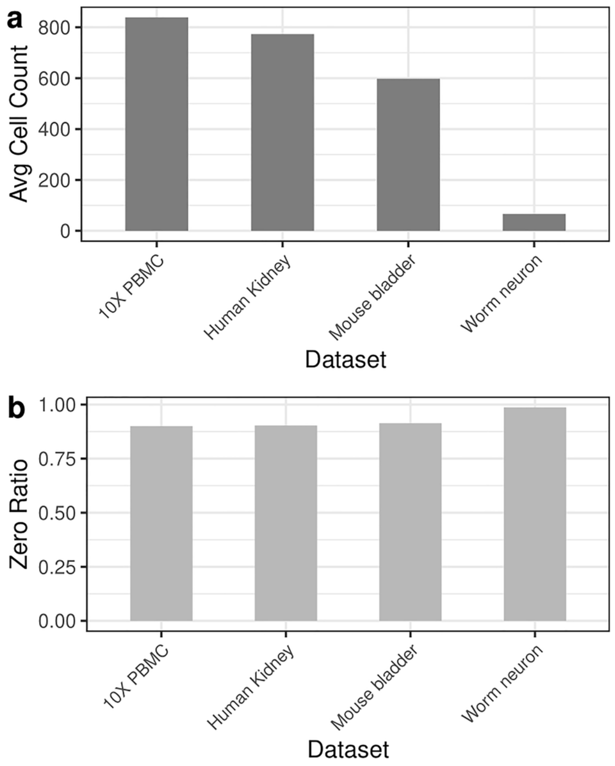

Fig. 7 Data structure of the real datasets used in this study. Average per cell count (**a**) and zero ratios (**b**) are shown for the four datasets: 1) 10X PBMC; 2) human kidney; 3) mouse bladder; and 4) worm neuron. The worm neuron dataset has a higher zero ratio and lower per cell count than other datasets. This is the possible reason that some models got very poor performance on this dataset.

from each cluster as the initial training samples. This is consistent with the theory of active learning but is challenging to be achieved in reality since the total number of clusters (cell types) is unknown. Although it only needs a few cells for each cell type, AL models can only cluster cells into the known cell types estimated by the marker genes (or other methods). For the cell types that cannot be estimated, AL models cannot cluster cells into it. Two potential methods can, to some extent, solve this problem. Firstly, if the selected cells could not be labeled (by oracle), we could define them as an unknown class, such as a class X, then the AL model could cluster some cells into this class. After running the AL model, if there were too many cells clustered in the class X, we could use an unsupervised method to separate them further. We can also define more than one unknown class, such as class X, Y, Z, etc., based on the limited domain knowledge. In addition, although the cell types of some selected cells are unable to be identified, the probabilities of their clustering preferences are still useful to us. For example, suppose some unknown cells have a high probability of being classified in a cell type A, we can deduce that these cells are close to the cell type A in the cell differentiation trajectory. Combining with some domain knowledge, we may have an acceptable estimation of the cell type of these cells. However, these methods are unproven and underdeveloped; more studies are needed to improve them further.

In this study, we develop an AL model for the scRNA-seq data clustering. We find that the budget size is positively correlated with the performance of the AL model. RF is the best classifier for the AL model in terms of the clustering performance and the running time. The AL model can significantly exceed the clustering performance of the unsupervised methods with <1000 labeled cells, indicating it is a promising tool for the scRNA-seq data clustering.

DATA AVAILABILITY

The code and all datasets of this study are available on the GitHub: https://github.com/xianglin226/scAL.

REFERENCES

1. Bacher R, Kendziorski C. Design and computational analysis of single-cell RNA-sequencing experiments. Genome Biol. 2016;17:63.
2. Sun S, Zhu J, Ma Y, Zhou X. Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. Genome Biol. 2019;20:1–21.
3. Tian T, Wan J, Song Q, Wei Z. Clustering single-cell RNA-seq data with a model-based deep learning approach. Nat. Mach. Intell. 2019;1:191–8.
4. Wang T, Li B, Nelson CE, Nabavi S. Comparative analysis of differential gene expression analysis tools for single-cell RNA sequencing data. BMC Bioinform. 2019;20:40.
5. Talwar D, Mongia A, Sengupta D, Majumdar A. AutoImpute: Autoencoder based imputation of single-cell RNA-seq data. Sci Rep. 2018;8:1–11.
6. Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. Nat Rev Genet. 2019;20:273–82.
7. Jaitin DA, Kenigsberg E, Keren-Shaul H, Elefant N, Paul F, Zaretsky I, et al. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. Science. 2014;343:776–9.
8. Wang X, Sun Z, Zhang Y, Xu Z, Xin H, Huang H, et al. BREM-SC: a bayesian random effects mixture model for joint clustering single cell multi-omics data. Nucleic Acids Res. 2020;48:5814–24.
9. Ringeling FR, Canzar S. Linear-time cluster ensembles of large-scale single-cell RNA-seq and multimodal data. Genome Res. 2021;31:677–88.
10. Ji Z, Ji H. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Res. 2016:44;e117–e117.
11. Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, et al. SC3: consensus clustering of single-cell RNA-seq data. Nat Methods. 2017;14:483–6.
12. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat Biotechnol. 2018;36:411–20.
13. Tian T, Zhang J, Lin X, Wei Z, Hakonarson H. Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. Nat Commun. 2021;12:1–12.
14. Chen L, He Q, Zhai Y, Deng M. Single-cell RNA-seq data semi-supervised clustering and annotation via structural regularized domain adaptation. Bioinformatics. 2021;37:775–84.
15. Settles B. Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences; MINDS@UW; 2009.
16. Prince M. Does active learning work? A review of the research. J Eng Educ. 2004;93:223–31.
17. Hubert L, Arabie P. Comparing partitions. J Classif. 1985;2:193–218.
18. Strehl A, Ghosh J. Cluster ensembles–a knowledge reuse framework for combining multiple partitions. J. Mach Learn Res. 2002;3:583–617.
19. Kuhn HW. The Hungarian method for the assignment problem. Nav Res Logist Q. 1955;2:83–97.
20. Balcan M-F, Broder A, Zhang T. Margin based active learning. In: International Conference on Computational Learning Theory. 35–50. Springer; International Conference on Computational Learning Theory; 2007.
21. Zheng GXY, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun. 2017;8:1–12.
22. Cao J, Packer JS, Ramani V, Cusanovich DA, Huynh C, Daza R, et al. Comprehensive single-cell transcriptional profiling of a multicellular organism. Science. 2017;357:661–7.
23. Young MD, Mitchell TJ, Vieira Braga FA, Tran M, Stewart BJ, Ferdinand JR, et al. Single-cell transcriptomes from human kidneys reveal the cellular identity of renal tumors. Science. 2018;361:594–9.
24. Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, et al. Mapping the mouse cell atlas by microwell-seq. Cell. 2018;172:1091–107.e1017.
25. Wei C, Sohn K, Mellina C, Yuille A, Yang, F. Crest: a class-rebalancing self-training framework for imbalanced semi-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021;10857–66.

AUTHOR CONTRIBUTIONS

XL, ZW, and SB performed study design and development of methodology. ZW, SB, and NG review and revision of the paper; XL and HL performed data analysis and interpretation, and statistical analysis; ZW and SB provided technical and material support. All authors read and approved the final paper.

FUNDING

The research was partially supported by the National Center for Advancing Translational Sciences (NCATS), a component of the National Institute of Health (NIH) under award number UL1TR003017.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41374-021-00639-w.

Correspondence and requests for materials should be addressed to Z.W.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.